FDRSOS

YOUR LIFE LINE TO

SAFEGUARD

OPEN

STORAGE

# Installation and User's Guide

INNOVATION® DATA PROCESSING

**CORPORATE HEADQUARTERS:** 275 Paterson Ave, Little Falls, NJ 07424 • (973) 890-7300 • Fax: (973) 890-7147
**E-mail: support@fdrinnovation.com • sales@fdrinnovation.com • http://www.innovationdp.fdr.com**

| **EUROPEAN OFFICES:** | FRANCE | GERMANY | NETHERLANDS | UNITED KINGDOM | NORDIC COUNTRIES |
|---|---|---|---|---|---|
| | 01-49-69-94-02 | 089-489-0210 | 036-534-1660 | 0208-905-1266 | +31-36-534-1660 |

# FDRSOS®

## VERSION V5.3

**PURPOSE OF THE MANUAL**

The purpose of this manual is to provide you with the information to install, use and understand FDRSOS.

**WHAT IS FDRSOS?**

FDRSOS provides disaster/recovery of Open System disk volumes by backing them up to and restoring from MVS-attached mainframe tapes and disks.

**Open System Disk volumes** are disks formatted for use by Unix systems and certain other systems such as OS/2®, Windows NT® and Novell®Netware®.

**EMC Symmetrix® ESP-capable storage subsystems** allow MVS systems to read and write Open System volumes, but only using the special I/O techniques built into FDRSOS.

**Performance.** FDRSOS supports ESCON channels for disk and tape access, and supports the newest high-speed, hi-capacity tape drives, to provide high-performance backup and restore..

FDRSOS is a trademark of INNOVATION DATA PROCESSING and is registered with the US patent office.

FDRSOS is a proprietary program product of
INNOVATION DATA PROCESSING
Innovation Plaza
275 Paterson Avenue
Little Falls, New Jersey 07424-1658.
It is available for license exclusively from INNOVATION DATA PROCESSING, INC.

**EUROPEAN OFFICES:**


**France**
Innovation Data Processing S.A.R.L.
191 Avenue Aristide Briand
94230 CACHAN
Tel: 01-49-69-94-02 Fax: 01-49-69-90-98
E-mail: frsupport@fdrinnovation.com
frsales@fdrinnovation.com

**Netherlands**
Innovation Data Processing
Brouwerstraat 8
1315 BP Almere
Tel: 036-534 1660 Fax: 036-533 7308
E-mail: nlsupport@fdrinnovation.com
nlsales@fdrinnovation.com

**Germany**
Innovation Data Processing
International Ltd.
Orleansstrasse 4a
D-81669 München
Tel: 089-489 0210 Fax: 089-489 1355
E-mail: desupport@fdrinnovation.com
desales@fdrinnovation.com

**United Kingdom**
Innovation Data Processing Ltd.
Clarendon House
125 Shenley Road
Borehamwood, Herts, WD6 1AG
Tel: 0208-905 1266 Fax: 0208-905 1428
E-mail: uksupport@fdrinnovation.com
uksales@fdrinnovation.com



© Copyright 1996, 1997, 1998, 1999

© INNOVATION DATA PROCESSING, INC.

**Reproduction of this User Manual is prohibited except by licensed users for their internal use.**


A copy of this manual is provided on CD-ROM in IBM BookManager format for online use, and in Adobe PDF format for local printing

**US CONTACTS**


Voice: 973-890-7300 Fax: 973-890-7147
Home page: www.innovationdp.fdr.com
E-mail: Sales: sales@fdrinnovation.com
Technical Support: support@fdrinnovation.com

THIS PAGE WAS INTENTIONALLY LEFT BLANK

## TABLE OF CONTENTS

Page No.

### FDRSOS INTRODUCTION AND OVERVIEW


SUMMARY OF MODIFICATIONS V5.3 LEVEL 30 . . . . . v
SUMMARY OF MODIFICATIONS V5.3 LEVEL 11 . . . . . v
SUMMARY OF MODIFICATIONS V5.3 LEVEL 05 . . . . . v
SUMMARY OF MODIFICATIONS V5.3 LEVEL 04 . . . . . v
SUMMARY OF MODIFICATIONS V5.3 LEVEL 03 . . . . . vi
SUMMARY OF MODIFICATIONS V5.3 LEVEL 02 . . . . . vi

200.01 FDRSOS OVERVIEW . . . . . 200-1
200.02 FDRSOS CONCEPTS . . . . . 200-2
200.03 FDRSOS IMPLEMENTATION . . . . . 200-6
200.04 FDRSOS VOLUME STATUS DISPLAYS . . . . . 200-13
200.05 FDRSOS TIMEFINDER™ BCV SUPPORT . . . . . 200-16


### FDRSOS JCL, STATEMENTS, EXAMPLES


210.01 FDRSOS JOB CONTROL REQUIREMENTS . . . . . 210-1
210.02 FDRSOS DUMP STATEMENT . . . . . 210-7
210.03 FDRSOS RESTORE STATEMENT . . . . . 210-11
210.04 FDRSOS PRINT STATEMENT . . . . . 210-15
210.05 FDRSOS ERASE STATEMENT . . . . . 210-17
210.06 FDRSOS LABEL STATEMENT . . . . . 210-19
210.07 FDRSOS VARYON/VARYOFF/VARYPATH STATEMENT . . . . . 210-21
210.08 FDRSOS ESTABLISH/RE-ESTABLISH . . . . . 210-24
210.09 FDRSOS SPLIT STATEMENT . . . . . 210-27
210.10 FDRSOS LOCALBACKUP STATEMENT . . . . . 210-29
210.15 FDRSOS MOUNT STATEMENT . . . . . 210-34
210.16 FDRSOS SELECT/EXCLUDE STATEMENT . . . . . 210-38
210.17 FDRSOS PRINT MODIFIERS . . . . . 210-39
210.20 FDRSOS INITIALIZATION EXAMPLES . . . . . 210-41
210.21 FDRSOS BACKUP EXAMPLES . . . . . 210-45
210.22 FDRSOS RESTORE EXAMPLES . . . . . 210-48
210.23 FDRSOS PRINT EXAMPLES . . . . . 210-50
210.24 FDRSOS ERASE EXAMPLES . . . . . 210-52
210.25 FDRSOS TIMEFINDER™ BCV EXAMPLES . . . . . 210-53

## TABLE OF CONTENTS

Page No.

### FDRSOS UTILITIES AND SPECIAL CONSIDERATIONS


220.01 FDRSOSTC BACKUP COPY UTILITY . . . . . 220-1
220.02 FDRSOSTC JCL REQUIREMENTS . . . . . 220-2
220.03 FDRSOSTC CONTROL STATEMENTS . . . . . 220-3
220.04 FDRSOSTC EXAMPLES . . . . . 220-4
220.05 FDRSOS OPEN SYSTEM UTILITIES . . . . . 220-6
220.10 FDRSOS SPECIAL CONSIDERATIONS . . . . . 220-7
220.11 FDRSOS COORDINATION . . . . . 220-8
220.12 FDRSOS AND OTHER BACKUPS . . . . . 220-11
220.13 FDRSOS AND FDR/UPSTREAM. . . . . 220-12
220.14 FILE RESTORES FROM FDRSOS BACKUPS . . . . . 220-15
220.15 FDRSOS DIAGNOSTIC TECHNIQUES . . . . . 220-17


### FDRSOS INSTALLATION AND MESSAGES


230.01 FDRSOS INSTALLATION. . . . . 230-1
230.02 INSTALLING THE INSTALLATION CONTROL LIBRARY. . . . . 230-2
230.03 INSTALLING THE PROGRAM LIBRARY . . . . . 230-4
230.04 INSTALLING THE JCL LIBRARY . . . . . 230-5
230.10 FDRZAPOP GLOBAL OPTION CHANGE FACILITY . . . . . 230-6
230.11 FDRZAPOP STATEMENTS AND OPTIONS . . . . . 230-7
230.20 FDRSOS MESSAGES . . . . . 230-9
230.21 FDRSOS CONSOLE MESSAGES. . . . . 230-10
230.22 FDRSOS PRINTER MESSAGES. . . . . 230-11
230.23 FDRSOS ABEND CODES . . . . . 230-29


### FDRSOS APPENDICES


A. FDRSOS WINDOWS NT . . . . . A-1
B. FDRSOS NETWARE . . . . . B-1
C. FDRSOS OS/2 . . . . . C-1
D. FDRSOS AIX . . . . . D-1
E. FDRSOS HP-UX . . . . . E-1
F. FDRSOS SOLARIS . . . . . F-1

## SUMMARY OF MODIFICATIONS

### Summary of Modifications for FDRSOS V5.3 LEVEL 30

**FBA TEST UTILITY**

FBACHK, a program provided by EMC but distributed in the FDR program library, is now documented in section 220.15.

**FDRSOS PROGRAM LIBRARY**

FDRSOS can now be installed in the same program library as other FDR programs, as long as they are all at the same maintenance level (some programs are shared by FDRSOS and other FDR components).

### Summary of Modifications for FDRSOS V5.3 LEVEL 11

**LOCAL BACK-UP ON SUN SPARC SYSTEMS**

Disk volumes managed by SUN SOLARIS SPARC systems can now be used as FDR/UPSTREAM/SOS local backups.

### Summary of Modifications for FDRSOS V5.3 LEVEL 05

**FDR/UPSTREAM LOCAL BACKUPS**

FDR/UPSTREAM has been enhanced to use disk volumes in an EMC Symmetrix as a special local backup. Data written to the local backup by FDR/UPSTREAM on the Open System can be read by FDR/UPSTREAM under MVS and copied to MVS-managed tape or disk files, without transmitting the data over the network. This results in much faster backups. FDRSOS has been enhanced to initialize the local backup disks for use with FDR/UPSTREAM. This feature is only available if you are licensed for the FDR/UPSTREAM/SOS product.

**TIMEFINDER™ BCV RESTORE**

FDRSOS now supports the BCV “Restore” function, where the contents of a BCV volume can be copied back to the primary volume. You might use the split BCV as a temporary backup of a primary volume while a test is being run; the restore function can quickly overlay any changes by copying all data back from the BCV. It is invoked by the RESTOREFROMBCV operand on the ESTABLISH or RE-ESTABLISH statements.

### Summary of Modifications for FDRSOS V5.3 LEVEL 04

**TIMEFINDER™ BCV VOLUMES**

With the advent of EMC's Timefinder(tm) product, EMC Symmetrix storage subsystems support Business Continuance Volumes (BCVs). BCVs are extra mirrors of primary volumes in the Symmetrix. These BCVs can be "split" from their primary volumes, creating an instant frozen copy of the volume. FDRSOS can then backup the frozen BCV, even though updates are occurring to the primary volume. When the backup is complete, the BCV can be reconnected to its primary volume, and any data that was updated while they were split are copied to the BCV.

This process allows you to quiesce your Open System volumes for a very short period of time, just long enough to split the BCVs of the volumes to be processed. The backup can then be done without time pressure. Support for BCVs includes new FDRSOS commands (ESTABLISH, RE-ESTABLISH, and SPLIT) plus new options for other commands.

## Summary of Modifications for FDRSOS V5.3 LEVEL 03

**VOLRESET**

Please note that the VOLRESET= parameter was incorrectly documented in the V5.3 level 02 manual. The meanings were exactly reversed, VOLRESET=YES should have been VOLRESET=NO and vice versa. However, in Appendix D (AIX/6000 restore), the use of VOLRESET was correct. The documentation in this manual is correct.

**VARYPATH**

A new FDRSOS command, VARYPATH, is available for use when one or more paths to an FDRSOS volume are offline. Normal MVS VARY commands cannot put those paths back online, but the VARYPATH command can do so, making all paths available if they are physically online.

**HP/UX AND SUN SOLARIS**

FDRSOS now contains enhanced support for HP/UX and SUN SOLARIS systems. PRINT=STATUS and VOLRESET=NO now support these systems.

## Summary of Modifications for FDRSOS V5.3 LEVEL 02

Version 5.3 level 02 was the initial production release of FDRSOS.

## 200.01 FDRSOS OVERVIEW

FDRSOS (Safeguard Open Storage) provides high-speed, reliable backup and restore of Open System (SCSI) data on an MVS system (including MVS/XA, MVS/ESA, and OS/390).

When used with an appropriately configured EMC Symmetrix Storage Subsystem, FDRSOS under MVS can be used to backup and restore disk volumes which are actually being used as SCSI disks by a UNIX system, a PC, or a server. Backups can be output to MVS tape, or to data sets on MVS DASD volumes.

FDRSOS uses the data management strength of IBM's S/390 hardware and MVS software data management to provide high-speed backups which are easily managed.

FDRSOS full-volume backups include all data sectors on the selected disks. A facility to backup and restore user-specified data sectors from Open System disks is provided.

FDRSOS also offers a function to print data from Open System disks and to scan for specific ASCII or hex data strings.

### EMC SYMMETRIX

FDRSOS is possible because of a teaming agreement between Innovation Data Processing and EMC Corporation (Hopkinton, MA). Under this agreement, EMC has produced storage subsystems which allow access to data from both SCSI and S/390 channels, and Innovation has produced the software (FDRSOS) which can be used on MVS to backup and restore this data.

EMC provides this capability as an option on their Symmetrix ESP (Enterprise Storage Platform) 3000-series and 5000-series ICDA (Integrated Cached Disk Array) Storage Subsystems. This option does not allow normal MVS access methods to read or write data on the Open System volumes; only special software such as FDRSOS has this ability.

### WHY?

Why do you want to backup Open System data on your MVS mainframe, when there are numerous alternatives? The reasons are numerous:

- Your MVS operations staff already has experience and procedures for taking and managing backups, both onsite and offsite.
- Your MVS tape management system provides automatic management of backup tapes, including retention, scratch pools, and automatic offsite vaulting.
- You may have automated tape libraries on your MVS mainframe, allowing for "lights-out" off-shift backups.
- Your Open System servers may be located in the MVS computer room. Even if they are not, ESCON channels allow the EMC Storage Subsystem to be accessed by MVS even though it is many kilometers away.
- You can consolidate the backups of data belonging to many Open System CPUs (as well as that of your S/390 data) in one location.
- Backups can be taken at high-speed, not depending on LAN or communication link speeds.
- New technology S/390 tapes, such as IBM Magstar, allow for high-speed backup to a small number of tape volumes.

Compared to most local or LAN-based backup systems, FDRSOS provides:

- Efficiency
- Centralized administration
- Improved resource utilization

## 200.02 FDRSOS CONCEPTS

### EMC SYMMETRIX

EMC Symmetrix ESP-capable Storage Subsystems can be formatted as:

- S/390 DASD (3380 and 3390 disk volumes) in CKD (count-key-data) format, attached to S/390 systems via parallel (bus/tag) or ESCON channels.
- SCSI disks in FBA (Fixed Block Architecture) format, accessed by UNIX or PC systems using SCSI channels. Although "Open System" is most commonly used to refer to UNIX systems, EMC disks in FBA format are called Open System disks even when they are used by PC-type systems such as Windows NT.
- A combination of the two in the same Symmetrix subsystem.

Although the FBA format of the SCSI disks is not supported by MVS access methods, EMC does allow Open System disks to be accessed by special software (such as FDRSOS) on the S/390 system as long as appropriate EMC channel adapters are installed and properly configured.

The physical disks within the EMC Symmetrix are divided into "logical disks" according to a user-designed configuration which assigns one or more logical disks of user-chosen size to each physical disk. Each logical disk can be configured as a S/390 disk or a SCSI Open System disk. The association of the logical disks to the actual physical disks in the Symmetrix is transparent to the end user.

Every S/390 logical disk naturally has a S/390 device address assigned so that it can be used by MVS. Every SCSI logical disk has a SCSI address associated with it. However, the SCSI disks can also be assigned a S/390 address as well, allowing them to be accessed by FDRSOS.

Although these are called "logical disks" by EMC, they appear to be physical disks to the S/390 or SCSI interfaces, and IBM documentation for the S/390 and documentation for the Open System hardware/software may refer to them as physical devices. In some Open System documentation, "logical volumes" may have a different meaning altogether.

### DEVICE ADDRESSING

The device addresses of S/390 and SCSI logical disks within the Symmetrix are assigned by the EMC representative in consultation with the customer. Naturally, the MVS I/O configuration must be updated to reflect these devices (via SYSGEN, IOCP, or HCD, depending on your level of MVS).

Those Symmetrix addresses which represent S/390-formatted disks (if any are present in the subsystem) can be varied online and used like any normal 3380 or 3390 DASD volume. Backup and management of these disks can be accomplished using Innovation's FDR DASD Management products, including FDR, FASTCPK, FDRREORG, and FDR/ABR, or similar software.

However, those addresses corresponding to the SCSI Open System disks cannot be varied online. They look to the S/390 hardware like FBA disks, which are not supported by MVS, so any attempt to VARY or MOUNT those devices will fail. These are the disks which can be backed up with FDRSOS.

CONTINUED . . .

200.02 CONTINUED . . .

**FDRSOS ACCESS TO OPEN SYSTEM DATA**

FDRSOS includes special I/O routines to identify EMC Open System devices and to access them even though they are offline to MVS. FDRSOS will identify those devices which might be EMC Symmetrix SCSI disks (offline DASD devices) and will perform several validations to ensure that they truly are this type of device (e.g., that they respond to FBA commands). Once identified, FDRSOS can perform high-efficiency FBA I/O to read and write these disks as required.

**FDRSOS FUNCTIONS**

FDRSOS DUMP allows you to:

- backup one or more Open System volumes serially (one at a time). Each backup must be directed to a separate output data set on tape or disk, but it is possible to stack multiple backups on a single tape volume (or multi-volume tape set) as multiple files.
- backup one or more Open System volumes in parallel, outputting to separate data sets. If tape output is used, separate tape drives must be used in order for the backups to execute in parallel.
- backup entire volumes (the usual method) or include only selected data sectors (by absolute sector number ranges).
- create a duplicate backup while reading the disk data only once.

**Note that FDRSOS has no knowledge of the allocation of data on the Open System disks and cannot selectively backup only allocated data areas.**

FDRSOS RESTORE will restore from the backups created by FDRSOS, restoring entire Open System volumes or optionally selected sectors (by absolute sector number ranges).

FDRSOS ERASE can quickly rewrite all (or selected portions of) an Open System volume with zeros.

FDRSOS can PRINT selected data sectors from Open System volumes, and can scan for user-specified ASCII or hexadecimal data strings and print the sectors containing those strings.

Finally, FDRSOS has a function (LABEL) which allows you to assign volume serials to each of the Open System volumes and two others (VARYON and VARYOFF) which allow you to make the Open System volumes available or unavailable for further FDRSOS processing.

**VOLUME IDENTIFICATION**

Open System volumes do not normally have MVS-style 6-character volume serials, but MVS disk operations are oriented around such a volser, so FDRSOS requires that every Open System volume be assigned a volume serial. You may assign any valid volume serial to the volumes, using the LABEL function of FDRSOS (See Section 210.05). This volume serial will be stored in an area of each Open System volume that EMC has reserved for FDRSOS use (outside of the normal data area of the volume).

Once assigned, this volume serial may be used to identify Open System volumes to FDRSOS in JCL and in control statements.

CONTINUED . . .

## 200.02 CONTINUED . . .

**COORDINATION WITH OPEN SYSTEMS**

Proper operation of FDRSOS backup and restore may require coordination with the Open System which is using the volume. For example, it may be necessary to quiesce updates to files on the Open System while the FDRSOS backup is taking place. Restores may require that the volume be dismounted on the Open System and remounted afterwards. FDRSOS includes utility programs which can be executed on the Open Systems to assist with this coordination for use in situations where the Open System does not provide native commands to do the necessary coordination.

The amount of time that updates must be quiesced for a backup can be minimized by use of EMC Timefinder™, an optional Symmetrix facility which can create instant "frozen" copies of Symmetrix volumes. It is described in detail in Section 200.05.

Innovation Data Processing's LAN-to-MVS backup product, FDR/UPSTREAM, can initiate programs and batch files on many Open Systems as described in Section 220.13

**COORDINATION WITH OTHER BACKUPS**

Depending on the type of data being backed up, you may need to coordinate FDRSOS backups with other file-level backups of the same data to provide all of the recovery that you need. Remember that FDRSOS backups are full-volume backups which are used primarily for full-volume recovery, while most other Open System backups process individual files. Section 220.11 goes into detail on the coordination that may be required.

Innovation Data Processing's LAN-to-MVS backup product, FDR/UPSTREAM, can provide the additional backups for most types of data. FDR/UPSTREAM is being enhanced to initiate and record FDRSOS backups so that they can be part of FDR/UPSTREAM backups. Section 220.13 has more details.

**SECURITY CONSIDERATIONS**

Security for FDRSOS operations is provided by RACF or equivalent third-party security products. Security checking is not the default; security must be enabled in FDRSOS using the FDRZAPOP utility described in Section 230.10. For every Open System volume to be dumped , restored or erased, FDRSOS will perform a security check against the DASDVOL security class, using the volume serial of the EMC Open System volume. FDRSOS will check for READ authorization for backups, and ALTER authorization for restores and erases.

**UTILITIES**

FDRSOSTC is a backup copy utility, described in Section 220.01. FDRSOS backups are usually written in a format which cannot be copied by normal copy utilities (such as IEBGENER); **however, such utilities may copy FDRSOS backups without any error message but will leave the copy in an unusable format.** It is important that FDRSOSTC be used whenever it is necessary to create a copy of a FDRSOS backup.

Several utility programs designed to run on various Open Systems in support of FDRSOS are provided. They are described in the Appendixes of this manual.

## 200.02 CONTINUED . . .

### FDR/UPSTREAM LOCAL BACKUPS

FDRSOS customers who also are licensed for FDR/UPSTREAM/SOS, Innovation's LAN-to-MVS backup product for file-level backups, can use EMC Symmetrix Open System volumes for backups of Open System data using an option called "FDRSOS local backup". On the Open System, FDR/UPSTREAM file backups are directed to specially formatted EMC Open System volumes. The backup data can include data from non-EMC disks which are attached to the Open System, as well as backups of other data in the Symmetrix.

Normally, FDR/UPSTREAM file backup data must be sent over the LAN or network to MVS. But when these FDRSOS local backups are used, the backup data is written solely to the FDRSOS local backup disk; only descriptions of the files involved are transmitted, so the backup will run much faster than a network backup. FDR/UPSTREAM has been enhanced to read those local backups from the MVS side, using the same I/O techniques used in FDRSOS, to copy the backup data to MVS tape or disk backup data sets.

FDRSOS is required to initialize the Open System volumes to be used as FDRSOS local backup volumes for FDR/UPSTREAM. See "LOCALBACKUP Statement" in Section 210.10 and "FDRSOS and FDR/UPSTREAM" in Section 220.13 for details.

## 200.03 FDRSOS IMPLEMENTATION

These are the steps necessary to implement FDRSOS backup and restore:

### FDRSOS INSTALLATION

Of course, FDRSOS must be installed on your MVS system as described in Section 230.01.

### EMC SYMMETRIX INSTALLATION AND CUSTOMIZATION

If not already present, a EMC Symmetrix 3000-series or 5000-series storage subsystem must be installed with the proper hardware and microcode necessary to support Open System (SCSI-attached) storage as well as concurrent S/390 access to the Open System volumes. The subsystem might also contain S/390-formatted (CKD) storage volumes; these volumes will not be processed by FDRSOS. Your EMC representative will ensure that the proper hardware and microcode is installed.

Your EMC representative will assist in configuring the SYMMETRIX storage subsystem. You will, of course, decide on the size and number of Open System (SCSI) and S/390 (CKD) volumes to be defined in the subsystem. Every S/390 volume will be assigned a S/390 address, and every Open System volume will be assigned both a SCSI address and a S/390 address.

### UPDATE MVS I/O CONFIGURATION

You must now update your MVS I/O configuration to include the new addresses. Depending on the type of CPU and the level of your MVS system, this might be done with:

- Sysgen
- IOCP (I/O Configuration Program) and MVSCP (MVS Configuration Program)
- HCD (Hardware Configuration Definition) dialog

The EMC SYMMETRIX subsystem will be defined as one or more 3990 control units. Depending on your EMC hardware configuration, the same control unit may be used to access both Open System and S/390 volumes, and the same channel paths (CHPIDs) may be used for both.

The Open System volumes must be defined as normal 3380 or 3390 disks. However, since they are NOT normal disks, any attempt to vary them online will result in error messages. **So, it is very important that the I/O configuration mark them as OFFLINE, so that MVS will not attempt to access them at IPL time.** Remember that FDRSOS will be able access them even though they are offline. They also be marked as SHARED.

## 200.03 CONTINUED . . .

**UPDATE MVS I/O CONFIGURATION (continued)**

For example, in HCD, the MVS options for an Open System device will look similar to:

```
                View Device Parameter / Feature Definition    Row 1 of 6
Command ===> ____________________________________________ Scroll ===> PAGE

Configuration ID    .   :   CURPIO99          Common Configuration data
Device number   .   .   :   01E0              Device type  .  .  .  :  3390
Generic  /  VM device type   .   .   .   .   :  3390

Parameter/          Value     Req.   Description
Feature
OFFLINE             Yes              Device considered online or offline at IPL
DYNAMIC             Yes              Device supports dynamic configuration
LOCANY              Yes              UCB can reside in 31 bit storage
ALTCTRL             No               Separate physical control unit path
SHARED              Yes              Device shared with other systems
SHAREDUP            No               Shared when system physically partitioned
********************************************** Bottom of data ***********************************************
```

In IOCP/MVSCP, specify OFFLINE=YES and FEATURE=SHARED on the IODEVICE macros for the Open System devices. The following is an example of a Symmetrix attached via 2 ESCON channels (CHPIDs 40 and 41), with 128 CKD (S/390) disk addresses (300-37F) and 128 FBA (Open System) addresses (380-3FF):

```
CNTLUNIT CUNUMBR=008,PATH=(40),LINK=(C0),
       UNIT=3990,UNITADD=((00,256))
CNTLUNIT CUNUMBR=009,PATH=(41),LINK=(C4),
       UNIT=3990,UNITADD=((00,256))
IODEVICE ADDRESS=(300,128),CUNUMBR=(008,009), ** CKD DEVICES
       UNIT=3390,UNITADD=00
IODEVICE ADDRESS=(380,128),CUNUMBR=(008,009), ** FBA DEVICES
       UNIT=3390,UNITADD=80,OFFLINE=YES,FEATURE=SHARED
```

Note that some documentation may indicate that UNITADD=00 is required for ESCON; this is true, but what it really means is that the value for the **first** IODEVICE definition for a given control unit must be 00. So when varying characteristics must be specified for some devices on a control unit (such as OFFLINE=YES), you can have multiple IODEVICE statements with varying values for UNITADD.

As is normal in a I/O configuration for a XA, ESA, or OS/390 system, the device numbers assigned to the Open System devices are arbitrary numbers which do not have to have any relation to the actual S/390 device addresses. The device numbers assigned will be the unit addresses referred to by the UNIT= operand of the FDRSOS MOUNT statement (Section 210.15).

CONTINUED . . .

## 200.03 CONTINUED . . .

**UPDATE MVS I/O CONFIGURATION (continued)**

You may wish to generate an esoteric unit name which includes the EMC Open System devices, which can be used in DISKx DD statements in FDRSOS JCL. For example, you might assign generic name OPENSYS to the Open System devices, allowing you to code UNIT=OPENSYS in JCL. Likewise, you will probably not want to include the Open System devices in other DASD esoteric names such as SYSDA since they will never be used for normal MVS disk allocations.

**Note to VM users:** If you are accessing Open System volumes from MVS running as a guest under VM, you should define the addresses of the Open System volumes to VM as "unsupported" DASD-class devices in your HCPRIO I/O configuration or your system configuration files. The definition should indicate that the devices have the DPS and RESERVE features. These devices should be DEDICATEd or ATTACHed to the MVS guest. Consult your VM Planning manual for more information. Here is an example of a VM RDEVICE statement for an Open System Volume:

```
Rdevice    0fd0   Type Unsupported,
                  DevClass DASD,
                  DPS yes,
                  Reserve_Release yes
```

CONTINUED. . .

## 200.03 CONTINUED . . .

**VERIFY CONNECTIVITY**

Once the updated I/O configuration has been activated (by an IPL or dynamic I/O reconfiguration) and the Symmetrix subsystem has been attached to the proper channel paths, you can perform a simple test to verify that the Symmetrix disk addresses are properly connected (addressable by FDRSOS).

For a normal CKD disk, this could be done by initializing them with ICKDSF or VARYing them online, but neither of these will work for Open System volumes.

One way to verify the connectivity is to use the MVS console command DEVSERV (abbreviated DS). For example, if your Open System volumes were at addresses 1E0-1EF, you could issue the command:

DS PATHS,1E0,16

which would produce a display similar to:

```
    IEE459I 16.49.32 DEVSERV PATHS 321
 UNIT DTYPE           M CNT VOLSER  CHPID=PATH STATUS
 01E0,3390           ,F,000,E#01E0,20=D 21=<
 01E1,3390           ,F,000,E#01E1,20=+ 21=<
 01E2,3390. . .       ,F,000,E#01E2,20=& 21=<
                           ...
******************************** SYMBOL DEFINITIONS
F = OFFLINE
< = PHYSICALLY UNAVAILABLE
+ = PATH AVAILABLE
& = RESERVED TO ANOTHER PATH/GROUP
* = LOGICALLY OFF, PHYSICALLY ON
- = LOGIALLY OFF, PHYSICALLY OFF
```

The status of each path (CHPID) to each device is shown. A good connection is indicated by:

- \+ path available
- & path available but the device is currently reserved to another system, such as a AIX/6000 system

Problems might be indicated by:

- < path online but device could not be contacted, control unit not responding
- * path available but marked offline for this device by a VARY PATH command
- \- path physically offline to the hardware. You may be able to vary it online with the command: CF CHP(xx),ONLINE.

Other status codes are possible and usually indicate a problem.

## 200.03 CONTINUED . . .

**VERIFY CONNECTIVITY (continued)**

So in this example, CHPID 20 is successfully connected to the Symmetrix Open System addresses while CHPID 21 is not. Device 01E2 is connected but reserved to another system.

If one or more CHPIDs have a status of * (logically offline but physically online), it may be the result of a MVS problem or a VARY PATH console command. The VARYPATH command of FDRSOS (See Section 210.07) may be able to resolve the problem.

When connection problems are indicated, they may be due to:

- errors in the Symmetrix internal configuration
- problems with the Symmetrix channel adaptor
- problems with the physical channel connection
- errors in the MVS I/O configuration

Your EMC representative will assist you in diagnosing these problems but you may also contact Innovation Technical Support for assistance. But before you call, please review Section 220.15 (FDRSOS DIAGNOSTIC TECHNIQUES).

## 200.03 CONTINUED . . .

### IDENTIFY THE OPEN SYSTEM VOLUMES

In order to manage the backups created by FDRSOS, you obviously must know what is being backed up. In other words, there must be some way to identify which backups contain which Open System volumes. Unfortunately, this is not straightforward.

On the Open System side (the UNIX system, or Windows NT system, or whatever), the disk volumes are known by their SCSI addresses. However, after initial installation, the SCSI addresses are rarely used and are often not even known to the system users. Disk volumes are assigned some other identification which varies by system type. They may be known by a drive letter (e.g., the "C" drive), by a system-assigned identification (e.g., "hdisk3") or some other identification. Unfortunately, this identification is known only to that system; it is usually not recorded on the disk volume itself so FDRSOS cannot determine what it is.

As described earlier, every Open System volume in the EMC Symmetrix subsystem will have both a SCSI address and a S/390 address. It is your responsibility to maintain documentation showing the correspondence between each S/390 Open System address and the data that has been placed on that volume from the Open System side. Your EMC representative can provide you with a configuration report showing the defined volumes and their addresses.

You may wish to maintain a spreadsheet or table similar to this:

| SCSI Addr | S/390 Addr | EMC Drive | EMC Sym# | Size (MB) | FDRSOS volser | Connected System | Open Sys Disk ID | Format | Usage |
|---|---|---|---|---|---|---|---|---|---|
| 00 | 01E0 | DA1C0 | 00 | 2000 | AIX1H1 | AIX Sys 1 | hdisk1 | JFS | Gen Files |
| 01 | 01E1 | DA1C0 | 01 | 2000 | AIX1H2 | AIX Sys 1 | hdisk2 | Oracle | Database |
| 10 | 01E2 | DA1D0 | 02 | 4000 | NTADRD | NT A | D: | HPFS | Payroll |

### STATUS PRINT

FDRSOS contains a function which can assist you with verifying that you have correctly identified the Open System volume mounted on each S/390 device address. The PRINT=STATUS displays will display a great deal of information about the format of each Open System volume processed and the Open System which created it. Section 200.04 contains more detail and examples on PRINT=STATUS.

## 200.03 CONTINUED . . .

### OPEN SYSTEM VOLUME SERIALS

FDRSOS requires that every Open System volume be assigned a volume serial number, in standard MVS format (up to 6 alphabetic, numeric or national characters, e.g., OPEN#2 or EMC123). This allows you to create jobstreams which are independent of the actual device addresses, since you specify the volume serials instead of device numbers to the FDRSOS backup and restore jobs.

We recommend that you assign volume serials that are in some way related to the name by which the volume is known on the Open System which uses it. You might use a form of the Open System volume name, or include the SCSI address (Target/Lun) or use the Open System device name. For example, for an AIX hdisk1 on a RS/6000 system, you might call it AIXHD1. For a Sun/Solaris hard disk "/dev/dsk/cat3d0s0" at SCSI Target/Lun 3/01, you might call it C2T3D1.

The Open System volume serials are assigned using the LABEL function of FDRSOS described in Section 210.06. This assignment needs to be done only once for a given volume; the volser is recorded in an area of the EMC volume that is reserved for FDRSOS use. If new Open System volumes are defined, you will need to run FDRSOS LABEL to assign volsers to the new volumes only. Otherwise you only need to run LABEL if:

- you need to change a volume serial
- the EMC hardware has been reconfigured changing the size or location of the Open System volumes so that the original volume serials have been lost
- a new EMC Symmetrix subsystem is replacing the original system (such as a replacement subsystem at a disaster site).

The assigned volume serial is also stored in the MVS UCB (Unit Control Block) of the Open System volume, although the device will continue to appear offline to MVS. This allows the volume to be referenced in JCL and also shows up in many device displays. For example, a MVS "D U" command will show results similar to:

```
        IEE457I 11.02.51 UNIT STATUS
UNIT    TYPE    STATUS          VOLSER          VOLSTATE
01E0    3390    OFFLINE         EMC123                  /RSDNT
01E1    3390    OFFLINE         OPEN04                  /RSDNT
```

However, after an IPL, the volume serials stored in the MVS UCBs are lost. So, after an IPL but before any other FDRSOS functions are run, you must execute the VARYON function of FDRSOS (see Section 210.07) to read the volume serials from each Open System volume and reinsert them in the MVS UCBs. Innovation recommends that you run the VARYON step as an automatically executed job or started task at IPL time or as the first step of any FDRSOS job.

### FDRSOS JOBSTREAMS

Finally, you must create the batch jobstreams to execute FDRSOS to backup and restore your Open System volumes, as described in the remainder of this manual. The examples in Sections 210.20-25 are a good starting point.

## 200.04 FDRSOS VOLUME STATUS DISPLAYS

Many FDRSOS functions accept an operand of PRINT=STATUS (see the statement descriptions in Section 210). If specified, FDRSOS will examine each Open System volume being processed, and attempt to identify the format of the volume using internal indicators and operating system footprints on the volume. If the volume type is successfully identified, FDRSOS will report the format and the type of Open System which created the volume (e.g., IBM AIX, HP-UX, SUN SOLARIS, NOVELL, OS/2, Windows NT, etc.). If applicable, it will display volume or volume group identifications, volume group information, UNIX mount points, and other information; the details will vary depending on the type of Open System.

If the additional operand DIR is specified, e.g., PRINT=(STATUS,DIR), for some disk formats it will also display file information from the root directory on the volume.

**NOTE:** The volume information displayed by FDRSOS is based on Innovation's current understanding of the format of volumes created by each of the Open Systems supported, as obtained by research and experimentation. Innovation will update the PRINT=STATUS display in subsequent releases of FDRSOS based on experience. If FDRSOS does not recognize the format of your Open System volumes ("UNKNOWN VOLUME TYPE") or displays incorrect information, please contact Innovation so that we can investigate and correct the problem.

Following are examples of PRINT=(STATUS,DIR) displays for some of the Open System formats currently supported by FDRSOS. Some additional formats are supported with similar displays. The first FDR216 message always displays the MVS unit address, FDRSOS pseudo volume serial, and Symmetrix device number (subsystem ID and internal logical disk number).

**SUN SOLARIS**

```
FDR216   STATUS OF SOS SCSI VOLUME UNIT=01E6 VOL=E#01E6 - SYM-NUMBER=X'00F0-0026'
FDR216   - UNIT=01E6 IS AN SUN SOLARIS VOLUME WITH A VOLUME NAME--->NO NAME
FDR216   - UNIT=01E6 -WITH A LABEL--->EMC-SYMMETRIX-5063 cyl 4428 alt 2 hd 15 sec 64
FDR216   - UNIT=01E6  -SLICE 00 (ROOT)          123 MB SIZE MOUNT POINT--->/
FDR216   - UNIT=01E6  -SLICE 01 (SWAP)          128 MB SIZE
FDR216   - UNIT=01E6  -SLICE 02 (BACKUP)      2,076 MB SIZE MOUNT POINT--->/
FDR216   - UNIT=01E6  -SLICE 06 (USR)           671 MB SIZE MOUNT POINT--->/usr
FDR216   - UNIT=01E6  -SLICE 07 (ALTTRK)        257 MB SIZE MOUNT POINT--->/export/home
FDR216   - UNIT=01EC  -SLICE 06 (USR)           257 MB SIZE--->/export/home
```

For SUN Solaris systems, the volume is defined in "slices". Note that the "backup slice" defines the entire volume, so in this example the EMC Symmetrix volume is defined as 2GB. The "usr" slice is the area usable for data files.

The label displayed is a label written by Solaris when the volume was formatted. The text is obtained from the Symmetrix subsystem at that time and shows the Symmetrix microcode level and device size. Note that the microcode level may not be the current level if it has been updated since the volume initialization.

**AT&T/NCR (SYSTEM V)**

```
FDR216   STATUS OF SOS SCSI VOLUME UNIT=01E6 VOL=E#01E6 - SYM-NUMBER=X'00F0-0027'
FDR216   - UNIT=01E6 - SYSTEM V UNIX PARTITION WITH A VOLUME NAME-*volumec
FDR216   - UNIT=01E6  -SLICE 00 (ROOT)          123 MB SIZE MOUNT POINT--->/
FDR216   - UNIT=01E6  -SLICE 01 (SWAP)          128 MB SIZE
FDR216   - UNIT=01E6  -SLICE 02 (BACKUP)      2,076 MB SIZE MOUNT POINT--->/
FDR216   - UNIT=01E6  -SLICE 06 (USR)           671 MB SIZE MOUNT POINT--->/usr
FDR216   - UNIT=01E6  -SLICE 07 (ALTTRK)        257 MB SIZE MOUNT POINT--->/export/home
```

AT&T and NCR systems run SYSTEM V UNIX, which has a disk structure similar to SUN Solaris.

## 200.04 CONTINUED. . .

**IBM AIX/6000**

```
FDR216   STATUS OF SOS SCSI VOLUME UNIT=01FA VOL=E#01FA - SYM-NUMBER=X'00F0-003A'
FDR216   - UNIT=01FA IS AN IBM AIX VOLUME WITH A PVID---->00024141-0340F21E
FDR216   - UNIT=01FA IS VOLUME   1 OF VOL GROUP ID VGID-->00024141-0340F764
FDR216   - UNIT=01FA GROUP HAS   4 LOGICAL VOLUMES ON   2 PHYSICAL DISKS
FDR216   - UNIT=01FA    259 PARTITIONS USED ON VOLUME OF   8 MBYTES EACH
FDR216   - UNIT=01E4 MIRROR COPY OF VOLUME->testmirr
FDR216   - UNIT=01E4 -WITH A MOUNT POINT--->/testmirr
FDR216   - UNIT=01FA LOGICAL VOLUME NAME--->testfs01
FDR216   - UNIT=01FA -WITH A MOUNT POINT--->/testfs01
FDR216   - UNIT=01FA LOGICAL VOLUME NAME--->testmirr
FDR216   - UNIT=01FA -WITH A MOUNT POINT--->/testmirr
FDR216   - UNIT=01FA -WITH A MOUNT POINT--->/testmirr
```

On AIX/6000 UNIX systems (for the IBM RS/6000), physical volumes (Symmetrix logical volumes) may be defined as one or more AIX logical volumes on a volume group. The volume group can consist of one or more physical volumes and the logical volumes may be spread across those physical volumes. Logical volumes may also be mirrored, where the data is recorded multiple times on the physical volumes for recoverability; however, the hardware mirroring facility of the Symmetrix is a superior way of accomplishing the same thing. The physical volumes are divided into partitions of fixed size; one or more partitions are assigned to logical volumes or mirrors as needed.

In this example, the PVID (physical volume ID) is assigned by AIX when the volume is initialized; it is a combination of the disk serial number and a timestamp. All volumes in the AIX volume group are assigned a similar VGID (volume group ID). The number of physical volumes (2) and logical volumes (4) in the volume group are displayed, as well as the partition size (8MB) and the number of partitions on this physical volume. The name and mount point of each logical volume on this physical volume are shown; logical volumes which are AIX mirrors are identified.

**HP/UX**

```
FDR216   STATUS OF SOS SCSI VOLUME UNIT=01E0 VOL=E#01E0 - SYM-NUMBER=X'00F0-0020'
FDR216   - UNIT=01E0 IS AN HP-UX VOLUME WITH A PVID------>78341E7E-330CBB81
FDR216   - UNIT=01E0 IS VOLUME   1 OF VOL GROUP ID VGID-->78341E7E-330CBB83
FDR216   - UNIT=01E0 GROUP HAS   1 LOGICAL VOLUMES ON   1 PHYSICAL DISKS
FDR216   - UNIT=01E0 -WITH A VERITAS FILESYSTEM
```

HP/UX UNIX systems have volume groups and logical volumes similar to IBM AIX/6000 systems (see description above). FDRSOS displays the PVID (physical volume ID) and VGID (volume group ID) assigned by HP/UX. Mount points and root directories may also be displayed.

However, unlike AIX, HP/UX has no convenient way of displaying the PVID and VGID of disks. Innovation has provided a utility which can be run on HP/UX to display the PVID and VGID of every online disk, allowing you to easily match the data displayed by PRINT=STATUS with the disk as seen by HP/UX. This utility is on the diskette supplied with FDRSOS and is called HPPVDISP.TAR; simply uncompress the program and execute HPPVDISP under HP/UX.

Veritas filesystems have no mount points to display. If the HP/UX volume has a standard HP file system, mount points are displayed.

## 200.04 CONTINUED. . .

**NOVELL NETWARE**

```
FDR216   STATUS OF SOS SCSI VOLUME UNIT=01EF VOL=E#01EF - SYM-NUMBER=X'00F0-002F'
FDR216   - UNIT=01EF HAS A PC/UNIX PARTITION TABLE WITH A SIGNATURE OF 147E84ED
FDR216   - UNIT=01EF PARTITION   1 OCCUPIES SECTOR       63 TO   2,120,579 - 1,035 MB SIZE
FDR216   - UNIT=01EF --OEM ID--->NOVELL   TYPE--->NETWARE
FDR216   - UNIT=01EF  --NETWARE VOLUME NAME--->DATABASE 1      -        100 MB SIZE
FDR216   - UNIT=01EF  --NETWARE VOLUME NAME--->DATABASE 2      -         10 MB SIZE
FDR216   - UNIT=01EF  --NETWARE VOLUME NAME--->SPARE 1         -          4 MB SIZE
FDR216   - UNIT=01EF  --NETWARE VOLUME NAME--->SPARE 2         -          5 MB SIZE
FDR216   - UNIT=01EF  --NETWARE VOLUME NAME--->DATABASE 3      -         25 MB SIZE
```

Novell Netware volumes may be partitioned, so FDRSOS displays the number and size of each partition (this example has only one partition). Within each partition it displays the partition format (Novell Netware) and the Netware volumes in that partition with their sizes.

**WINDOWS NT**

```
FDR216  STATUS OF SOS SCSI VOLUME UNIT=01FD VOL=E#01FD - SYM-NUMBER=X'00F0-003D'
FDR216  - UNIT=01FD HAS A PC/UNIX PARTITION TABLE WITH A SIGNATURE OF 147E84E7
FDR216  - UNIT=01FD PARTITION   1 OCCUPIES SECTOR      32 TO   2,119,679 -  1,035 MB SIZE
FDR216  - UNIT=01FD --OEM ID--->NTFS  TYPE--->NTFS  VOLID--->D4FD-D067 NAME--->DATABASE1
```

Windows NT volumes may be partitioned, so FDRSOS displays the number and size of each partition (this example has only one partition). Within each partition it displays the partition format (NTFS).

**OS/2**

```
FDR216   STATUS OF SOS SCSI VOLUME UNIT=01EA VOL=E#01EA - SYM-NUMBER=X'00F0-002A'
FDR216   - UNIT=01EA HAS A PC/UNIX PARTITION TABLE WITH A SIGNATURE OF 147E84E9
FDR216   - UNIT=01EA ->LINKAGE PARTITION ON SECTOR       2,048 TO   2,119,679 -      1,034 MB SIZE
FDR216   - UNIT=01EA PARTITION   1 OCCUPIES SECTOR       2,080 TO     206,847 -        100 MB SIZE
FDR216   - UNIT=01EA --OEM ID--->OS2 20.0 TYPE--->HPFS     VOLID--->A698-0C14 NAME--->PAYROLL 1
FDR216   - UNIT=01EA  --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216   - UNIT=01EA    01/14/97 15.59.15              8 20 QUERY.EXE
FDR216   - UNIT=01EA    01/14/97 15.59.15              8 20 README.TXT
FDR216   - UNIT=01EA    01/14/97 16.47.49 <DIR>          50 payroll
FDR216   - UNIT=01EA   -END OF DIRECTORY ENTRIES
FDR216   - UNIT=01EA ->LINKAGE PARTITION ON SECTOR     206,848 TO     227,327 -         10 MB SIZE
FDR216   - UNIT=01EA PARTITION   2 OCCUPIES SECTOR     206,880 TO     227,327 -         10 MB SIZE
FDR216   - UNIT=01EA --OEM ID--->OS2 20.0 TYPE--->HPFS     VOLID--->26E5-0815 NAME--->PAYROLL 2
FDR216   - UNIT=01EA  --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216   - UNIT=01EA   -END OF DIRECTORY ENTRIES
FDR216   - UNIT=01EA ->LINKAGE PARTITION ON SECTOR     227,328 TO   1,251,327 -        500 MB SIZE
FDR216   - UNIT=01EA PARTITION   3 OCCUPIES SECTOR     227,360 TO   1,251,327 -        500 MB SIZE
FDR216   - UNIT=01EA --OEM ID--->OS2 20.0 TYPE--->HPFS     VOLID--->26F1-4815 NAME--->HUMAN RES
FDR216   - UNIT=01EA  --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216   - UNIT=01EA   -END OF DIRECTORY ENTRIES
FDR216   - UNIT=01EA ->LINKAGE PARTITION ON SECTOR   1,251,328 TO   1,353,727 -         50 MB SIZE
FDR216   - UNIT=01EA PARTITION   4 OCCUPIES SECTOR   1,251,360 TO   1,353,727 -         50 MB SIZE
FDR216   - UNIT=01EA --OEM ID--->IBM 20.0 TYPE--->FAT      VOLID--->E272-C815 NAME--->NO NAME
FDR216   - UNIT=01EA  --ROOT DIRECTORY ENTRIES: ----SIZE FL FILENAME
FDR216   - UNIT=01EA    01/16/97 14.34.34 <VOLUME LABEL> 08 FAT-VOL03
FDR216   - UNIT=01EA    01/16/97 14.47.36             10 20 TESTERGG.GGG
```

OS/2 volumes may be partitioned, so FDRSOS displays the number and size of each partition and the volume ID on that partition (the volume ID consists of a "volume serial" assigned by OS2 and the optional "volume label" specified by the user when the partition was formatted. Within each partition it displays the partition format (OS2 HPFS or IBM FAT). If the DIR option was specified ("PRINT=(STATUS,DIR)") the contents of the root directory in each partition is displayed (files and subdirectory names). In this example, partitions 2 and 3 have been formatted but contain no files

## 200.05 FDRSOS TIMEFINDER™ BCV SUPPORT

### TIMEFINDER™ BUSINESS CONTINUANCE VOLUMES

In 1997 EMC enhanced their Symmetrix storage subsystems with Timefinder™, an optional Symmetrix facility. Timefinder™ and FDRSOS allow "instant" backups of Open System volumes by "freezing" an image of a volume at a particular point in time, then backing up that frozen image.

The Symmetrix has always allowed the definition of mirror volumes, which are exact up-to-date copies of the primary volumes to which they are assigned (all changes are written to both the primary volume and its mirrors). This provides protection against data loss due to hardware failure, but the mirror volumes do not have their own device address and can not be accessed directly by the host.

Timefinder™ introduced a new type of mirror, called a Business Continuance Volume or BCV. A given primary volume may have standard mirrors as well as BCVs, but unlike standard mirrors, a BCV

- is not permanently assigned to a primary volume
- can be assigned (established) by host software as a BCV mirror of any primary volume in the Symmetrix subsystem with the same size and format (FBA, 3390, or 3380).
- can be detached (split) by host software from its currently assigned primary volume.
- has its own device address (a S/390 address, SCSI address, or both) and can be used as a primary volume when not currently assigned to a primary volume.

To define BCVs, you must have the Timefinder™ support installed on your Symmetrix, and you must have unused disk capacity within the subsystem to assign as BCVs.

EMC can provide you with more information on Timefinder™ and BCVs, including more detail on the internal operation of BCVs. Although EMC provides BCV utility programs on MVS and Open Systems, all the necessary support is built into FDRSOS. Innovation recommends that you use FDRSOS's integrated BCV support, instead of the EMC utility, for Open System volumes processed by FDRSOS.

## 200.05 CONTINUED. . .

### BCV OPERATIONS

A BCV is first assigned as a mirror by an ESTABLISH command; this will synchronize it with the primary volume by copying all data on the primary to the BCV; this will take several minutes per Gigabyte (GB) to be copied. Once the copy is complete, all updates to the primary volume will be mirrored on the BCV, maintaining it as an exact copy. While the BCV is assigned, it cannot be directly addressed by the host.

At the point in time where a backup is desired, the BCV can be detached from the primary volume by a SPLIT command. The BCV will be an exact copy of the primary volume at the time that the SPLIT was issued. SPLIT is usually very quick. You will probably have to quiesce updates to the primary volume until the SPLIT is complete (see Section 220.11) but this usually takes only a few seconds.

Once split, the BCV is now usable via its assigned host address and can be backed up.

When the backup is complete, you can re-synchronize the BCV with the primary with a RE-ESTABLISH operation. During the time when the BCV was split, the Symmetrix keeps track of all data updated on the primary volume. RE-ESTABLISH copies only that updated data to the BCV, so it is much faster than a full ESTABLISH. FDRSOS supports a separate RE-ESTABLISH command, but it can also be automatically issued by FDRSOS DUMP as it finishes the backup of each volume.

Note that the copying of data during the ESTABLISH and RE-ESTABLISH is done internally by the Symmetrix subsystem and is a low priority task (real I/Os get priority). However, the amount of time required is not really important unless you need to SPLIT the BCV and do a backup before the volumes are completely synchronized; in that case the SPLIT is delayed until the synchronization is complete.

## 200.05 CONTINUED. . .

### SETTING UP A BCV ENVIRONMENT

There are 2 ways to setup BCV use:

1) you can permanently assign BCVs to every primary volume for which you plan to use BCV backup. This will require more Symmetrix capacity and more device addresses, but is much simpler to administer. The BCV is actively mirroring it's primary volume all the time, except during the backups, so the ESTABLISH is done only once. A SPLIT is done before the backup. After the backup, the RE-ESTABLISH will quickly re-synchronize the volumes since only changed data is copied.

2) you can create a pool of BCVs and assign them to primary volumes as necessary. This requires less Symmetrix capacity and fewer device addresses, but is more complex to administer. You will have to ESTABLISH the BCV every time it is to be used. This may take considerable time to copy all of the data from the primary volume, especially if multiple volumes are being ESTABLISHed. You may require operational procedures to insure that two FDRSOS jobs don't try to use the same BCV volume for two different primary volumes at the same time.

You might use a combination of the two, some primary volumes with permanent BCVs (perhaps for volumes with frequent or time-critical backups) and some with BCVs assigned from a pool (perhaps for less frequent or non-time-critical backups).

In either case, the BCV volumes must match the size (2GB, 4GB, or whatever) and format (FBA) of the primary volumes to which they will be attached. BCV volumes may also be assigned for CKD disks in the Symmetrix system but CKD BCVs cannot be assigned to FBA Open System volumes and vice versa.

It is also possible to use unassigned BCVs for other purposes, such as for restoring backups. This is described later in this section.

## 200.05 CONTINUED. . .

**FDRSOS AND TIMEFINDER™**

To allow use of BCVs with FDRSOS, the BCVs must be defined as Open System volumes, just like normal Open System volumes. They must be assigned both a S/390 address and a SCSI address, and they must be in the MVS I/O configuration as OFFLINE DASD devices (see Section 200.03).

FDRSOS has been enhanced with 3 new command statements:

**ESTABLISH**

**SPLIT**

**RE-ESTABLISH**

to perform the functions described above. Like other FDRSOS operations, each must be executed in a separate FDRSOS step, but each can process multiple Open System volumes in that step. You can tell FDRSOS to terminate the step as soon as the operation has been started, or you can tell it to wait for the synchronization or split to complete; use the latter when you must positively know that the BCV is in the desired status (such as when quiescing or enabling updates).

The **VARYON** statement has been enhanced to recognize BCVs. BCVs will not normally have their own FDRSOS volumes serials so VARYON will assign them a dummy volume serial of "E#uuuu" where "uuuu" is the MVS device address in hex, e.g., E#01F4. This dummy serial is not stored on the BCV itself, but is only stored in the MVS UCB. **After each IPL, you must run a VARYON function which includes the BCV volume addresses;** this can be the same VARYON that you use for primary Open System volumes as long as the MOUNT statement(s) include the addresses of both primary and BCV volumes. A BCV cannot be used by FDRSOS until the VARYON is performed. VARYON can also be executed at any time to display the status of the BCVs; it will identify them and show if they are unused, assigned to a primary volume, or split from a primary volume.

However, **VARYON** and **LABEL** accept a option (BCV=IGNORE) which allows BCV volumes to be used as primary volumes when they are not currently assigned to a normal primary volume. More details are given later in this section.

The **DUMP** statement has been enhanced to support BCVs. The BCV=USE operand tells it to identify and backup the BCV most recently split from the primary volume indicated by the MOUNT statement. The SPLIT function must be executed before the DUMP step, but there is also an option on DUMP to automatically RE-ESTABLISH the BCV to its primary volume when the backup is complete. If the primary volume does not have a current BCV (ESTABLISH never done or BCV was reused for another primary), or the BCV is still attached to the primary (SPLIT not done), an error message will result and the backup will be bypassed. If a preceding SPLIT has not yet completed (perhaps due to data which was not fully synchronized between the primary and BCV), the DUMP will automatically wait until the SPLIT has completed.

With the exception of ESTABLISH, you only need to specify the primary volume to FDRSOS; it will automatically determine the BCV volume most recently assigned to that primary volume.

## 200.05 CONTINUED. . .

**FDRSOS BCV OPERATIONS**

If you have a permanently assigned BCV volume for every primary volume, use these steps:

1) Execute an ESTABLISH one time for each primary-BCV pair. There is no need to wait for the synchronization to complete.

2) When ready to backup, quiesce system and/or application update activity on the primary volumes (if required, see Section 220.11) and execute a SPLIT with BCV=WAIT for the primary volumes. Once the SPLIT step ends, you can re-enable updates on the primary volumes.

3) Execute the DUMP with BCV=(USE,RET) for all the primary volumes. RET will cause an automatic RE-ESTABLISH for each volume as soon as the dump of the volume completes.

If you are using a BCV pool (described earlier), use these steps:

1) Execute an ESTABLISH for each primary volume to be backed up, attaching it to one BCV volume from the pool. Be sure to do the ESTABLISH far enough of the required backup time to allow the synchronization to complete. If you need to know when synchronization is complete (e.g., to know when you are ready to do the backups), use BCV=WAIT.

2) When ready to backup, quiesce update activity on the primary volumes (if required, see Section 220.11) and execute a SPLIT with BCV=WAIT for the primary volumes. If the ESTABLISH is not yet complete, the SPLIT will wait for synchronization. Once the SPLIT step ends, you can re-enable updates on the primary volumes.

3) Execute the DUMP with BCV=USE for all the primary volumes.

4) There is no need to RE-ESTABLISH the BCV unless you plan to do another BCV backup of the same primary volume. Once the backup is complete the BCV is available for use with a different primary volume. However, you may need operational procedures to insure that different FDRSOS jobs do not attempt to use the same BCV at the same time.

## 200.05 CONTINUED. . .

### BCV AUTOMATION

If you are using automation software to quiesce and re-enable updates to the primary volume on the associated Open System, there are two ways to tell when an FDRSOS BCV operation (such as SPLIT) is complete:

- the FDRSOS step ends (remember that the operation may not be truly complete unless BCV=WAIT is specified).
- FDRSOS puts message FDRW20 on the MVS console when BCV=WAIT is specified.

Effective use of automation will require some facility for executing the quiesce and re-enable procedures on the Open System when these events occur on the mainframe. You could manually execute the proper commands on the Open System at the appropriate times, but this is awkward and error-prone.

One software product which can automate this is FDR/UPSTREAM, Innovation's Lan-to-MVS backup system. FDR/UPSTREAM can initiate execution from MVS of programs, scripts, or procedures on many of the Open Systems supported by FDRSOS; it can also initiate MVS jobs and tasks from those Open Systems.

Here are two scenarios for the use of FDR/UPSTREAM to coordinate updates on the Open System with the BCV backups. A complete example based on scenario 1 can be found in Section 210.25. USTBATCH is the FDR/UPSTREAM utility which executes programs or scripts on an Open System. Note that creation of the jobstreams can be automated using FDR/UPSTREAM's ISPF interface.

1) this scenario is initiated from the MVS mainframe and consists of a jobstream with the following steps:

a) a USTBATCH step which causes FDR/UPSTREAM to execute a script or program on the Open System to quiesce updates to the volumes about to be backed up.

```
CONV=WAIT        wait for updates to be quiesced
TARGNAME=SERVER1      Open System ID
ACTION 5              run a job
SPECNUMBER 1
FILES C:\DB\CLOSEALL.BAT  execute this script
```

b) a FDRSOS step to SPLIT the BCVs.

```
SPLIT TYPE=SOS,BCV=WAIT  wait for SPLIT to complete
```

c) a USTBATCH step to re-enable updates on the Open System.

```
CONV=KEEP            wait until request accepted
TARGNAME=SERVER1     Open System ID
ACTION 5               run a job
SPECNUMBER 1
FILES C:\DB\OPENALL.BAT  execute this script
```

d) a FDRSOS step to backup the BCVs and RE-ESTABLISH them as they complete.

```
DUMP TYPE=SOS,BCV=(USE,RET)
```

## 200.05 CONTINUED. . .

**BCV AUTOMATION (continued)**

2) this scenario is initiated from the Open System and consists of the following 2 parts:

a) on the Open System, a procedure or script which will quiesce update activity to the volumes to be backed up, and then invokes FDR/UPSTREAM to submit a job to the MVS host. This job could consist of a complete jobstream (stored as a file on the Open System); alternately, it could simply invoke a cataloged procedure on MVS containing all of the necessary JCL and control statements.

b) the MVS jobstream would consist of the last 3 steps in scenario 1 (steps b, c, and d).

**USING A BCV AS A SPARE VOLUME**

BCVs are intended for use as backup vehicles as described in the rest of this section. However, a BCV that is not currently assigned as a mirror of a primary volume can actually be used as a separate primary volume. Obviously, you cannot store permanent data on a BCV, since that data would be lost as soon as you assigned the BCV as a mirror again. But it is possible to use BCVs as spare volumes when they are currently SPLIT from (or never ESTABLISHed to) a primary volume.

You might use BCVs for short-term storage of data, as long as that data can be discarded before your next backup cycle that requires the BCVs. One such short-term activity might use the BCVs as the targets for FDRSOS restores as described in Section 220.14 "File Restores from FDRSOS Backups". That section describes the procedures necessary to restore a copy of Open System data which is still online to a spare volume, so that older copies of individual Open System files can be recovered (copied back to the normal volumes). But this requires that sufficient spare volumes be available in the Symmetrix to enable these restores. Using BCVs as those spares, only when required, can reduce the total amount of disk storage required in the Symmetrix subsystem, replacing disks which are usually idle with multi-purpose disks which are usually doing productive work.

To use an unassigned BCV as a primary volume for FDRSOS, you will have to do a LABEL operation on the volume with the BCV=IGNORE option; the BCV will be labeled with the volume serial specified on the MOUNT statement (not the E#uuuu dummy volser normally used for BCVs). If you reIPL while a BCV volume is in use as a primary volume, you will have to do a VARYON operation with BCV=IGNORE to pick up the assigned volume serial.

Once the LABEL or VARYON is done, you can then use the BCV like any other primary volume. RESTORE (or any other FDRSOS operation) can refer to the BCV by the volume serial you assigned.

You may need operational procedures in place to insure that no other job tries to use the BCV while you are temporarily using it for the restore.

## 200.05 CONTINUED. . .

**RESTORE FROM BCV**

FDRSOS supports one more Timefinder™ function which may be useful. Timefinder™ has the ability to copy data from a BCV back to the primary volume. EMC calls this a "RESTORE", but FDRSOS supports it for Open System volumes with an option (RESTOREFROMBCV) on the ESTABLISH and RE-ESTABLISH statements.

If you split an established BCV from its primary volume, it comes a "frozen" copy of that volume, as described earlier. But if you later do a RE-ESTABLISH with RESTOREFROMBCV, then all tracks which were changed on the primary volume during the split will be copied from the BCV back to the primary volume, exactly the opposite of what RE-ESTABLISH normally does. The primary is reset to its original contents at the time of the split. This might be used to make a copy of one or more primary volumes, run tests against those volumes, and then reverse all the changes made during the test, returning those volumes to their original status.

Note that if you update any tracks on the BCV while it is split, the RE-ESTABLISH will copy to the primary the updated BCV tracks as well as all tracks which were updated on the primary.

An ESTABLISH with RESTOREFROMBCV will copy all tracks currently on the BCV to the primary volume, the opposite of a normal ESTABLISH. The BCV does not have to be currently associated with the primary volume. You might use this to do FDRSOS RESTORE of a primary disk backup to a BCV, then use the ESTABLISH to copy the restored data to the primary volume.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

## 210.01 FDRSOS JOB CONTROL REQUIREMENTS

To execute FDRSOS, the following JCL statements are required. Please review the examples (sections 210.20-25) for a better understanding of the JCL requirements of FDRSOS.

**STEPLIB or JOBLIB DD STATEMENT**

If required, specifies the job library in which FDRSOS resides. The library must be authorized.

**EXEC STATEMENT**

Specifies the program name (PGM=FDRSOS), region requirement (REGION=nnnK or nM), and optional PARM= field which may contain the first FDRSOS control statement
For example,

```
//SOS EXEC PGM=FDRSOS,REGION=0M,
//PARM='DUMP TYPE=FULL,RESERVE=YES'
```

The parm may also contain the second control statement, separated from the first by a slash (/), e.g.,

```
PARM='VARYON TYPE=SOS/MOUNT UNIT=03*'
```

Note that there must be no space immediately before the slash. Additional control statements, if any, must be contained in the SYSIN data set.

FDRSOS acquires most of its storage below the 16M line, in the area specified by the REGION= operand.

For each concurrent DUMP and PRINT operation, FDRSOS requires about 200K plus the storage required for the number of 36K buffers specified by the BUFNO= operand. With the recommended default of BUFNO=MAX, 30 buffers are acquired (30*36=1080K). If COMPRESS= is used, an additional 100K is need for each concurrent backup.

So, FDRSOS requires about 1300K for each concurrent backup (1400K with COMPRESS). If multiple concurrent backups are run in the same step because MAXTASKS=n is specified, multiply the region requirement by "n". You must specify a REGION= value at least as large as this requirement; if you do not, some operations may fail. There is no penalty for specifying a region larger than required; FDRSOS will still use only what it needs. The largest value you can specify for REGION= is different at every installation but is usually 6M or more. Innovation recommends that you specify REGION=0M so that FDRSOS has the maximum below-the-line region available.

For RESTORE operations, FDRSOS requires about 1400K for each concurrent restore. If MAXTASKS=n is specified, the region must be 1400K*n.

ERASE, LABEL, VARYON, and VARYOFF operations require about 500K.

**SYSPRINT DD STATEMENT**

Specifies the output message data set; it is required. It is usually a SYSOUT data set but if is it assigned to a data set on tape or disk, this DD must specify DISP=MOD.

210.01 CONTINUED . . .

**SYSPRINx DD STATEMENT**

Specifies the output data set for dump messages related to the matching TAPEx DD when the MAXTASKS= option is used; there must be a SYSPRINx for every TAPEx. It is usually a SYSOUT data set but it is it assigned to a data set on tape or disk, this DD must specify DISP=MOD. If MAXTASKS= is not used, the SYSPRINx DD statements can be omitted since all messages will be written to SYSPRINT.

**FDRSUMM DD STATEMENT**

(Optional) if present, FDRSOS will write one-line messages for each Open System volume dumped, restore, or printed, giving result codes, elapsed time, and byte counts. Usually a SYSOUT data set.

**SYSUDUMP DD STATEMENT**

Specifies the abend data set. Usually a SYSOUT data set. A SYSUDUMP DD statement should always be included to assist in error diagnosis. If you have the ABEND-AID product from COMPUWARE include the following so that a fully-formatted dump is produced: `//ABNLIGNR DD DUMMY`

210.01 CONTINUED . . .

**TAPEx DD STATEMENT**

Used to specify the output data set for DUMP, the input data set for RESTORE and the print data set for PRINT. "x" may be any alphanumeric character (0-9, A-Z). Multiple TAPEx DD statements may be present in the FDRSOS step JCL; a unique value for 'x' must be used for each of them (e.g., TAPE1, TAPE2, etc.)

For DUMP Operations:

Specifies a tape or sequential disk data set to which the backup will be written. You **must** provide a TAPEx DD statement for each Open System volume to be backed up in this step. If DISKx DD statements (described below) are used, the TAPEx DD will receive the backup of the volume specified by DISKx. If MOUNT statements are used, the TAPEDD=x operand on the MOUNT statement controls which volume will be directed to each backup data set (if only a single volume is backed up with MOUNT, TAPEDD= can be omitted and the backup will be written to the first or only TAPEx DD in the step).

If MAXTASKS= is specified with a value greater than 1, FDRSOS will attempt to attach that many concurrent backup subtasks, but may postpone some of them if it detects that they require a tape drive in use by another backup (see the notes on UNIT=AFF and VOL=REF below).

DUMMY is supported, for **testing** purposes only.

You must provide all the JCL parameters required to allocate and catalog the backup data set on disk or tape, which may include some or all of: DSN=, UNIT=, VOL=, SPACE=, and DISP=(NEW,CATLG). For tape, a volume count should be specified since the default is only 5 tape volumes, e.g., VOL=(,,,255).

DCB parameters are not required and should be omitted. However, IDRC (tape unit hardware compression, available on most tape cartridge drives) can be requested by adding DCB=TRTCH=COMP to your DD statement; IDRC compression may be the default depending on local MVS options. For tapes attached by ESCON channels, Innovation recommends use of IDRC compression instead of FDRSOS software compression (the COMPRESS= option).

Examples:

```
//TAPE1  DD DSN=PROD.OPEN.BACKUP1,UNIT=TAPE,
//    VOL=(,,,255),DISP=(NEW,CATLG)
//TAPE2  DD DSN=PROD.OPEN.BACKUP2,UNIT=DISK,
//    SPACE=(CYL,(500,100),RLSE),DISP=(,CATLG)
```

210.01 CONTINUED . . .

**TAPEx DD STATEMENT (continued)**

For tape backups, UNIT=AFF or VOL=REF may be specified, referencing another TAPEx DD statement, to reduce the number of tape drives used in the step. UNIT=AFF=TAPEx will cause MVS to allocate the same tape drive for both DD statements, but will call for separate output tapes when each DD is opened. VOL=REF=*.TAPEx with LABEL=n can be used to stack multiple backup files on the same tape, providing more complete utilization of the tape volumes (which may be important as new technology increases tape volume capacity). FDRSOS will automatically recognize that multiple TAPEx DDs point to the same tape drive and will serialize operations on that drive so that only one backup is directed to that drive at a time.

Examples:

```
//* The following creates 2 backups on 2 different
//* tape volumes using the same tape drive. This may not
//* fully utilize the tape volumes but will allow for
//* concurrent restores from these backups.
//TAPE1   DD DSN=PROD.OPEN.BACKUP1,UNIT=TAPE,
//        VOL=(,,,255),DISP=(NEW,CATLG)
//TAPE2   DD DSN=PROD.OPEN.BACKUP2,UNIT=AFF=TAPE1,
//        VOL=(,,,255),DISP=(NEW,CATLG)

//* The following creates a multi-file (and possibly
//* multi-volume) tape containing 3 backups.
//TAPE3   DD DSN=PROD.OPEN.BACKUP3,UNIT=TAPE,
//        VOL=(,,,255),DISP=(NEW,CATLG)
//TAPE4   DD DSN=PROD.OPEN.BACKUP4,LABEL=2,
//        VOL=REF=*.TAPE3,DISP=(NEW,CATLG)
//TAPE5   DD DSN=PROD.OPEN.BACKUP5,LABEL=3,
//        VOL=REF=*.TAPE4,DISP=(NEW,CATLG)
```

For RESTORE Operations:

Specifies a backup data set on tape or disk from which the data is to be restored. The backup must, of course, be a backup created by FDRSOS. You may include multiple TAPEx DD statements in order to restore multiple Open System volumes. A DISKx DD statement or the TAPEDD=x operand on a MOUNT statement is required to define which backup is to be restored to which volume.

Example: `//TAPE1 DD DSN=PROD.OPEN.BACKUP1,DISP=OLD`

However, if the backup data set is cataloged, RESTOREs can also be done without TAPEx DD statements. The BACKUPDSN= operand on the MOUNT statement can be used to specify the name of a cataloged data set from which an Open System volumes is to be restored; FDRSOS will dynamically allocate that data set.

If MAXTASKS= is specified with a value greater than 1, FDRSOS will attempt to attach that many concurrent restore subtasks, but may postpone some of them if it detects that they require a tape drive in use by another restore. You can use UNIT=AFF (see notes under DUMP above) to assign two or more restores to use the same type drive, but MVS may do this automatically if it detects that two input data sets required the same tape volume. MAXTASKS= is ignored if BACKUPDSN= is used to allocate the input data sets instead of TAPEx DD statements; restores with BACKUPDSN= are always done serially within a given job but you can execute multiple restore job to get the same effect.

210.01 CONTINUED . . .

**TAPEx DD STATEMENT (continued)**

For PRINT Operations:

Specifies the destination of the printed output from the PRINT statement. This is usually SYSOUT, but it may be directed to a disk or tape data set.

Example:

```
//TAPE1  DD  SYSOUT=*
```

For ERASE Operations:

Can be omitted if MAXTASKS=n is **not** specified and MOUNT statements are used to identify the volumes to be erased. However, if MAXTASKS=n is included, a TAPEx DD pointing to DUMMY and a SYSPRINx DD for messages must be included for each disk to be erased. If DISKx DDs are used to identify the volume, matching TAPEx DDs pointing to DUMMY must also be included.

Example:

```
//TAPE1  DD  DUMMY
```

For ALL other Operations:

TAPEx DD statements are not required and should be omitted.

**TAPExx DD STATEMENT**

For a DUMP operation only, specifies a second backup data set. The TAPExx data set will contain a backup identical to that written to TAPEx. For example, if a backup is being written to TAPE6, the inclusion of a TAPE66 DD statement will cause a second duplicate backup file to be produced. A given TAPExx DD statement will be ignored if there is no TAPEx DD with the same "x" character.

However if TAPEx is a backup data set on disk (backup to disk), then TAPExx cannot also point to a backup data set on disk. In other words, you cannot create 2 simultaneous disk backups of an Open System volume. You can backup TAPEx to tape and TAPExx to disk, or vice versa.

## 210.01 CONTINUED. . .

**DISKx DD STATEMENT**

Optionally used to specify the Open System volume to be dumped, restored, erased or printed (a FDRSOS MOUNT statement can also be used to identify the volumes to be processed). If used, the format will be:

```
//DISK1  DD  UNIT=unitname,VOL=SER=volser,DISP=OLD
```

"unitname" is either a generic name, such as 3390, or an esoteric name assigned during your I/O configuration, such as OPENSYS (see Section 200.03), and "volser" is the volume serial assigned to the Open System volume using the FDRSOS LABEL function (see Section 210.06).

**Warning**: if an IPL of your MVS system has occured since the FDRSOS LABEL function was run for a given Open System volume, you **must** run the FDRSOS VARYON function for each of the Open System devices before the volume serial can be used in JCL (see Section 210.07). Innovation recommends that you either run the VARYON function as an automatically executed job or started task at IPL time, or as the first step of any FDRSOS backup/restore job.

The "x" in DISKx must match a corresponding TAPEx DD statement. For DUMP, the Open System volume specified by DISKx will be backed up to TAPEx (and optionally TAPExx). For RESTORE, the DISKx volume will be restored from the TAPEx backup. For ERASE, the DISKx volume will be erased. For PRINT, the DISKx volume is printed on the TAPEx output data set. DISKx DD statements must be omitted for LABEL, VARYON and VARYOFF functions.

If DISKx DD statements are used, the Open System volumes will be processed in the order that the DISKx DDs appear in the FDRSOS JCL. If MOUNT statements are used, the volumes will be processed in the order that their UCBs are found in the MVS system, which may be difficult to predict. If the order of processing is important (such as when you are creating multiple files on a tape volume), DISKx DD statements are the only way to guarantee the order.

**Note:** JES3 systems may not accept DDs pointing to offline disks. Use MOUNT statements.

**Note:** Although the EMC microcode and cache will optimize performance in most cases, degradation due to excessive movement of the disk heads may occur when you are concurrently processing logical volumes which reside on the same physical disk within the EMC subsystem. For best performance, you should be aware of the physical location of the logical volumes involved and plan your backups and restores accordingly. You may need to segregate FDRSOS operations into multiple jobs run serially to enforce this.

**Note:** the MVS system messages for a FDRSOS step will contain an allocation for a DDname of "D#FDRSOS" as a temporary data set on your system residence volume. This is normal; it is for allocation purposes only and no real data set is allocated.

**SYSIN DD STATEMENT**

Specifies the control statement data set. Usually an input stream or DD * data set. It may be DUMMY if all necessary control statements are included in the PARM= on the EXEC statement.

## 210.02 FDRSOS DUMP STATEMENT

```
DUMP      TYPE=FULL | PARTIAL                  ,MAXERR=nnnn
D
                                               ,MAXTASKS=1 | n
          ,BCV=USE
          ,BCV=(USE,RET)                       ,PRINT=STATUS
                                               ,PRINT=(STATUS,DIR)
          ,BLKSIZE= nnnnn                      ,PRINT=UCB

          ,BUFNO=MAX | nn                      ,RESERVE=YES | NO

          ,COMPRESS=ALL | COPY1 | COPY2
```

**DUMP STATEMENT**

This statement requests a backup operation. It must be the first statement in the input; only one DUMP statement is allowed per execution. DUMP must be followed by one or more MOUNT statements to specify the Open System volumes to be backed up unless DISKx DD statements are used to identify the volumes (See Section 210.01).

If you have security checking enabled (via FDRZAPOP as shown in Section 230.10), DUMP requires that the user have at least READ authority to the Open System volser under the DASDVOL security class.

The BCV= operand us used when you have Business Continuuance Volumes assigned to the Open System volumes being backed up. By following the procedures outlined in Section 200.05, you can instantly create "frozen" copies of the volumes to be backed up, ignoring any further updates may take place to the primary volumes. These BCVs can be backed up at you convenience.

**OPERANDS**

**TYPE=** Specifies the type of backup and must be specified on the DUMP Statement.
**FULL** - requests that entire Open System volumes be backed up. This will backup every data sector on the selected volumes, including those which may not be currently allocated to a file.
**PARTIAL** - requests that selected sections of Open System volumes be backed up. One or more SELECT statements must be present. Partial backups should not be done unless you have a detailed knowledge of the structure and layout of data on the disk volumes.

## 210.02 CONTINUED. . .

**BCV=**

Used with Symmetrix BCV (Business Continuance Volumes) which are described in Section 200.05. BCVs allow a instant "frozen" copy of an Open System volume to be backed up instead of the volume itself, allowing updates to take place against the primary volume while the backup of the frozen BCV is proceeding. In order to use a BCV, the primary volumes to be backed up by this FDRSOS step must all have been ESTABLISHed (synchronized) with a BCV (see Section 210.08) and then SPLIT from the BCV (see section 210.09) at the point in time where the instant frozen copy is to be captured for backup.

**USE** - specifies that the BCV most recently SPLIT from each primary volume selected for backup is to be read instead of the primary volume itself. Only the primary volume serial is specified on the MOUNT statement or DISKx DD statement; FDRSOS will determine the most recently associated BCV device. The address of the BCV will be identified in a message, but all other messages will contain the address and volume serial of the primary volume.

**(USE,RET)** - same as USE, except that an automatic RE-ESTABLISH will be done as soon as the backup of each BCV volume is successfully completed, starting the process to resynchronize the BCV volume and the primary volume by copying all primary data which changed since the last SPLIT.

The default is that the primary volume named on the MOUNT statement will be backed up.

**Note:** if the backup of a BCV fails, you can simply resubmit the backup job after correcting the error, since the BCV will remain unchanged. Even if you specify BCV=(USE,RET), the RE-ESTABLISH will not be done unless the backup is successful.

**BLKSIZE=**

specifies the size in bytes of the largest block to be written on the backup tape. The value may range from 6000 to 61440. BLKSIZE= is ignored when outputting to a sequential DASD data set; the blocksize will be the maximum size of a half-track record on the output DASD device type. Regardless of the actual blocksize used, the tape label or disk DSCB of the backup data set will show RECFM=U and BLKSIZE=32760 because of MVS restrictions, so this cannot be used to determine the actual blocksize of the backup. If a BLKSIZE= value of 26000 or less is specified, the value of BUFNO= may be automatically decreased.

The default is 61440.

**Note:** Innovation recommends that you do not override the blocksize, for best performance and maximum tape and DASD utilization. However, tapes with blocks over 32760 in length may not be copied by standard copy utilities and usually cannot be transmitted with data transfer utilities; if this is necessary, BLKSIZE=32760 can be used to create a compatible backup tape with minimal impact on FDRSOS performance.
See Section 220.01 for documentation on FDRSOSTC, a utility to copy FDRSOS tapes with any blocksize.

## 210.02 CONTINUED. . .

**BUFNO=** specifies how many buffers will be used for dumping each Open System volume. Each buffer is 36K in length. The buffers acquired will be divided into 2 sets in order to overlap input and output I/O operations; each disk I/O will read into one half of the buffers. Reducing the number of buffers will reduce the amount of below-the-line storage required for each concurrent backup, but will also reduce the efficiency of the backup and increase the elapsed time.
**MAX** - specifies that 30 buffers will be acquired.
**nn** - the specified number of buffers is acquired. The value may be from 1 to 30 but it will be rounded up to the next higher even number. Values over 30 are treated as 30.

The default is MAX (30). **Innovation recommends that you do not override the default.**

**COMPRESS=** **ALL** – specifies that FDRSOS is to compress the output data on the backup file for both copies (TAPEx and TAPExx if present).
**COPY1** – specifies that only the backup data on TAPEx DD statements will be compressed.
**COPY2** – specifies that only the backup data on TAPExx DD statements, if present, will be compressed.
Compression of either or both copies will add 100K to the region required for each concurrent backup.

Default is NONE.

COMPRESS is recommended for backups to disk files, and for tape backups to tapes attached on parallel (bus/tag) channels. For tapes attached on ESCON (serial fiber optic) channels, use of IDRC (tape hardware compression) usually results in better performance.

**Note:** COMPRESS is not supported in the current release. It will be supported in a future release. If you are planning on using this feature, please contact Innovation.

**MAXERR=** Specifies the number of disk errors that are permitted by FDRSOS prior to abending the DUMP operation. MAXERR may specify a value from 1 to 9999 errors; MAXERR=1 forces termination immediately on any error. Each error will be indicated by a message and other diagnostic information; such errors usually indicate that some data was omitted from the backup which may make the backup unusable.

Default is 20 errors.

## 210.02 CONTINUED. . .

**MAXTASKS=** Specifies the maximum number of Open System volumes that will be dumped concurrently in this step. The value may be from 1 to 9 but the actual number of concurrent dumps will be no greater than the number of TAPEx DD statements in this step's JCL (if multiple TAPEx DD statements point to the same tape drive, one only concurrent dump will use that drive at any time).

The default is 1. However, if MAXTASKS= is omitted all backup messages will be printed on SYSPRINT while if MAXTASKS=n is specified (even if "n" is 1), messages are directed to the SYSPRINx DD corresponding to the TAPEx DD associated with the backup
(see Section 210.01).

**PRINT=** **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being backed up. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,PRINT=(STATUS,DIR),PRINT=UCB.

**RESERVE=** specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being backed up.

**YES** - the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the backup is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the backup will fail immediately. If BCV=USE is specified, the BCV will be reserved, not the primary volume.

**NO** - no RESERVE is issued.

The default is NO.

**Note**: Innovation does not recommend use of RESERVE=YES for backups unless you must prevent other MVS and/or SCSI systems from accessing the Open System volumes during the backup.

## 210.03 FDRSOS RESTORE STATEMENT

| | | |
|---|---|---|
| **RESTORE**<br>**R** | **TYPE=FULL \| PARTIAL** | **,MAXTASKS=**n |
| | **,CONFMESS=YES \| NO** | **,PRINT=STATUS** |
| | **,CPYVOLID=YES \| NO** | **,PRINT=(STATUS,DIR)**<br>**,PRINT=UCB** |
| | **,MAXERR=**nnnnn | **,RESERVE=YES \| NO** |
| | | **,VOLRESET=YES \| NO** |

**RESTORE STATEMENT**

This statement requests a restore operation. It must be the first statement in the input; only one RESTORE statement is allowed per execution. RESTORE must be followed by one or more MOUNT statements to specify the Open System volumes to be restored unless DISKx DD statements are used to identify the volumes (See Section 210.01).

If you have security checking enabled (via FDRZAPOP as shown in Section 230.10), RESTORE requires that the user have ALTER authority to the Open System volume serial under the DASDVOL security class.

**OPERANDS**

**TYPE=** Specifies the type of restore and must be specified on the RESTORE Statement.
**FULL** – requests that the entire contents of each backup data set be restored. This may be an entire Open System volume if the entire volume was backed up (DUMP TYPE=FULL) or may be selected portions of a volume if the backup was a partial backup (DUMP TYPE=PARTIAL).
**PARTIAL** – requests that selected sections of the contents of each backup data set be restored. A SELECT statement must be present. Note that data from Open System volumes will always be restored to the same block locations it was backed up from. Partial restores should not be done unless you have a detailed knowledge of the structure and layout of data on the disk volumes.

**CONFMESS=** **YES** – specifies that, before beginning the restore, FDRSOS will request confirmation via a WTOR message (FDRW01)to which the operator must reply.
**NO** – suppresses the WTOR and begins the restore immediately. CONFMESS=NO can be very useful at a disaster recovery site to avoid full volume restores being delayed waiting for an operator response.

Default is YES.

## 210.03 CONTINUED. . .

**CPYVOLID=** **YES** – FDRSOS will compare the volume serial of the Open System volume which was backed up in the input backup data set to the volume serial of the output disk. If they are different, at the end of the restore FDRSOS will change the volume serial of the output disk to the volume serial from the backup. The volume serial stored in the area of the volume reserved for FDRSOS use will be updated. The volume serial stored in the UCB of the output disk will be updated unless the same volume serial is already set in the UCB of another disk (either online or offline); in that case, the volume serial in the output UCB will be cleared (effectively a VARYOFF of the output volume).
**NO** – the volume serial of the output disk will not be changed, even If it is different from that of the Open System volume on the backup data set.

Default is NO.

**MAXERR=** Specifies the number of disk errors that are permitted by FDRSOS prior to abending the restore operation. MAXERR may specify a value from 1 to 9999 errors; MAXERR=1 forces termination immediately on any error. Each error will be indicated by a message and other diagnostic information; such errors usually indicate that some data was lost from the backup which may make the restored volume unusable.

Default is 20 errors.

**MAXTASKS=** Specifies the maximum number of Open System volumes that will be restored concurrently in this step. The value may be from 1 to 9 but the actual number of concurrent restores will be no greater than the number of TAPEx DD statements in this step's JCL (if multiple TAPEx DD statements point to the same tape drive, one only concurrent restore will use that drive at any time). MAXTASKS= is ignored if the backup data set is dynamically allocated using the BACKUPDSN= operand on a SELECT statement; it works only if TAPEx DDs are used to identify the backup data sets.

The default is 1. However, if MAXTASKS= is omitted all restore messages will be printed on SYSPRINT while if MAXTASKS=n is specified (even if "n" is 1), messages are directed to the SYSPRINx DD corresponding to the TAPEx DD associated with the restore
(see Section 210.01).

## 210.03 CONTINUED. . .

**PRINT=**

**STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being restored. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. This information will be displayed both before and after the restore. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g., PRINT=(STATUS,DIR),PRINT=UCB

**RESERVE=**

specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being restored. This is recommended during a RESTORE operation to ensure that the volume will not be accessed until the restore is complete.
**YES –** the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the restore is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the restore will fail immediately.
**NO** – no RESERVE is issued.

The default is YES.

**Warning: if RESERVE=YES is in effect, you must be sure that no other system (MVS or SCSI) has the Open System volume reserved; otherwise the FDRSOS restore will fail. See Section 220.11 for additional warnings about restore.**

## 210.03 CONTINUED. . .

**VOLRESET=** Most Open System volumes contain a volume identifier (usually called a PVID or signature), which is assigned when the volume is formatted. The format and location of the volume identifier varies, depending on the operating system (UNIX, Windows NT, etc.) and can also vary depending on the version of UNIX and the type of file system on the volume. Some volumes may have no volume identifier at all, such as "raw volumes" or other special format volumes. This volume identifier is often used by the Open System in identifying the volume when it is mounted. There may also be other identifiers and volume descriptors, such as logical volume identifiers. Normally, FDRSOS will backup and restore these identifiers, since they are part of the data sectors read and written by FDRSOS. However, when you are restoring an Open System volume to a different location (such as when you are restoring to spare volumes in order to recover individual files, as described in Section 220.14), you may need to preserve the identifiers currently assigned to the target volumes instead of restoring them from the backup. Please review Section 220.14 for more details.

**NO** - FDRSOS will attempt to preserve the current identifiers of the target (output) volumes. The target volumes should be preformatted as the same type of volume and file system as the volume being restored from backup.
If FDRSOS is successfully able to identify the type of volume on the target and identify the location of the identifier fields, it will read them from the output disk and substitute them for the identifiers being restored from the backup. This will allow the Open System to see these volumes with unchanged PVID values, although the contents have been restored. FDRSOS may also modify other identifiers (such as logical volume IDs) during the restore. If it cannot identify the IDs of the output disk, it will simply use the IDs from the input volume, with an incremental added to make them unique.
For UNIX and Novell Netware systems, FDRSOS will also attempt to identify the mount point name (UNIX) or logical volume name (Novell) and modify them so that they can be successfully mounted even though the original volumes are still mounted. It will add " _SOS" to mount points and " _S" to logical volume names.
VOLRESET=NO should be used when you are restoring to new volumes when the original volumes are still mounted (such as when preparing for a file-level recovery).

**YES** - identifier values will be restored from the backups unchanged. VOLRESET=YES should be used when restoring back to the original volumes and when restoring at a disaster site. If you are restoring backups to new volumes (not back on top of the original volumes) and the original volumes are still mounted, this will probably cause the restored volumes to be unusable.

Default is YES.

**Warning: do not specify VOLRESET=NO unless you understand the implications as described above and in Section 220.14. VOLRESET=NO may not support all type of volumes, see Section 220.14 for details.**

CONTINUED. . .

## 210.04 FDRSOS PRINT STATEMENT

| | | |
|---|---|---|
| **PRINT** | **TYPE=FULL \| PARTIAL** | **,PRINT=STATUS** |
| **P** | | **,PRINT=(STATUS,DIR)** |
| | **,MAXERR=**nnnnn | **,PRINT=UCB** |
| | | **,RESERVE=YES \| NO** |

**PRINT STATEMENT**

This statement requests a print operation. It must be the first statement in the input; only one PRINT statement is allowed per execution. PRINT must be followed by one or more MOUNT statements to specify the Open System volumes to be printed unless DISKx DD statements are used to identify the volumes (See Section 210.01).

The printed output will be directed to the TAPEx DD statement (the first TAPEx DD in the JCL unless DISKx DDs are used or the TAPEDD= operand is coded on the associated MOUNT statement). Each 512 byte data sector will be printed in "dump" format, with 32 bytes of hexadecimal data on each line, and the same data on the right in printable format (translated from ASCII to EBCDIC).

The PRINT function also includes the ability to scan for specified hexadecimal or ASCII strings of data and print only the data sectors containing the specified strings. Certain PRINT options can also be overridden. Details are found in Section 210.17.

## 210.04 CONTINUED. . .

**OPERANDS**

**TYPE=** Specifies the type of print and must be specified on the PRINT Statement.
**FULL** – requests that the entire contents of each Open System volume be printed. Since this may be a large amount of printout, this is not recommended unless a SCAN statement (Section 210.17) is present to limit the printout.
**PARTIAL** – requests that selected sections of the contents of each Open System volume be printed. A SELECT statement must be present.

**MAXERR=** Specifies the number of disk errors that are permitted by FDRSOS prior to abending the print operation. MAXERR may specify a value from 1 to 9999 errors. Each error will be indicated by a message and other diagnostic information.

Default is 20 errors.

**PRINT=** **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being printed. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g., PRINT=(STATUS,DIR),PRINT=UCB

**RESERVE=** specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being printed.
**YES** – the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the restore is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the print will fail immediatley.
**NO** – no RESERVE is issued.

The default is NO.

**Note:** Innovation does not recommend use of RESERVE=YES unless you **must** prevent other systems from accessing the Open System volumes during the print.

## 210.05 FDRSOS ERASE STATEMENT

| | | |
|---|---|---|
| **ERASE** | **TYPE=FULL** | **,MAXTASKS=**n |
| | **,CONFERASE=YES \| NO** | **,PRINT=STATUS**<br>**,PRINT=(STATUS,DIR)**<br>**,PRINT=UCB** |
| | **,MAXERR=**nnnnn | **,RESERVE=YES \| NO** |

### ERASE STATEMENT

This statement requests an erase operation. It must be the first statement in the input; only one ERASE statement is allowed per execution. ERASE must be followed by one or more MOUNT statements to specify the Open System volumes to be erased unless DISKx DD statements are used to identify the volumes (See Section 210.01).

ERASE will rewrite all sectors on the target Open System volume as binary zeros. **Obviously this is a very dangerous operation and must be used with care.** ERASE might be useful after a disaster recovery test or when replacing a EMC Symmetrix containing Open System data in order to be sure that all company data is removed from the disks.

If you have security checking enabled (via FDRZAPOP as shown in Section 230.10), ERASE requires that the user have ALTER authority to the Open System volume serial under the DASDVOL security class.

ERASE does not require TAPEx DD statements unless:

- MAXTASKS=n is specified, in which case a TAPEx DD DUMMY and a SYSPRINx DD are required for each volume to be erased (the volume must be identified by either a DISKx DD statement or the TAPEDD=x operand on a MOUNT statement. ERASE messages will be written to SYSPRINx
- A DISKx DD statement is used to identify the volume to be erased, in which case a TAPEx DD DUMMY must be included.

### OPERANDS

**TYPE=** **FULL** - requests that the entire contents of the Open System volume be erased. This is the only value currently supported for ERASE.

**CONFERASE=** **YES** – specifies that, before beginning the erase, FDRSOS will request confirmation via a WTOR message (FDRW01) to which the operator must reply.
**NO** – suppresses the WTOR and begins the erase immediately. CONFERASE=NO can be very useful at a disaster recovery site when erasing many volumes to avoid waiting for an operator response. **However, CONFERASE=NO is not recommended unless speed is of the essence and you are very sure that you have correctly specified the volumes to be erased.**

Default is YES.

**MAXERR=** Specifies the number of disk errors that are permitted by FDRSOS prior to abending the erase operation. MAXERR may specify a value from 1 to 9999 errors; MAXERR=1 forces termination immediately on any error. Each error will be indicated by a message and other diagnostic information.

Default is 20 errors.

CONTINUED. . .

## 210.05 CONTINUED. . .

**MAXTASKS=** Specifies the maximum number of Open System volumes that will be erased concurrently in this step. The value may be from 1 to 9 but the actual number of concurrent restores will be no greater than the number of TAPEx DD statements in this step's JCL.

The default is 1. However, if MAXTASKS= is omitted all erase messages will be printed on SYSPRINT while if MAXTASKS=n is specified (even if "n" is 1), messages are directed to the SYSPRINx DD corresponding to the TAPEx DD associated with the restore (see Section 210.01).

**PRINT=** **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being erased, before the erase begins. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. . This information is printed both before and after the ERASE; naturally, the after display will show that FDRSOS cannot determine the volume format. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g., PRINT=(STATUS,DIR),PRINT=UCB

**RESERVE=** specifies if FDRSOS is to issue a hardware RESERVE against each Open System volume while it is being erased. This is recommended during an ERASE operation to ensure that the volume will not be accessed while the erase is proceeding.
**YES** - the RESERVE will be issued. This RESERVE will prevent any other MVS system or SCSI-interface from accessing the volume until the erase is complete. If the volume is currently reserved to any other MVS system or SCSI-interface, the erase will fail immediately.
**NO** - no RESERVE is issued.

The default is YES.

**Warning: if RESERVE=YES is in effect, you must be sure that no other system (MVS or SCSI) has the Open System volume reserved; otherwise the FDRSOS erase will fail.**

## 210.06 FDRSOS LABEL STATEMENT

```
LABEL        TYPE=SOS

             ,BCV=IGNORE

             ,PRINT=STATUS
             ,PRINT=(STATUS,DIR)
             ,PRINT=UCB
```

**LABEL STATEMENT**

This statement requests a volume labeling operation. It must be the first statement in the input; only one LABEL statement is allowed per execution. LABEL must be followed by one or more MOUNT statements with the UNIT= and SETVOL= operands to specify the Open System volumes to be labeled.

The LABEL function must be executed against each Open System volume before it can be used by other FDRSOS functions. It assigns a volume serial to a volume and records that serial in an area of the volume reserved by EMC for FDRSOS use. You only need to execute LABEL against each Open System volume when:

- the Open System volume has not previously been used by FDRSOS
- you need to change an Open System volume serial
- the EMC hardware has been reconfigured changing the size or location of the Open System volumes so that the original volume serials have been lost
- a new EMC Symmetrix subsystem is replacing the original system (such as a replacement subsystem at a disaster site).

LABEL also stores the volser in the UCB of the Open System device so that it can be used in a DISKx DD statement in other FDRSOS steps. However, that volser is lost when your MVS system is reIPLed; after an IPL, the FDRSOS VARYON function (Section 210.07) must be executed before any FDRSOS backup, restore or print is performed.

**Note:** LABEL is not normally used with BCVs (see Section 200.05). Since BCVs are exact duplicates of their associated primary volume, they will have the same internal volser as that primary. However, it is possible to LABEL a BCV (with BCV=IGNORE) so that it can be used as a spare volume, e.g., a target for restore of an FDRSOS backup. More details on this procedure are in Section 200.05.

CONTINUED. . .

## 210.06 CONTINUED. . .

**OPERANDS**

**TYPE=SOS** Must be specified on the LABEL Statement.

**BCV=IGNORE** Normally, FDRSOS will identify all Symmetrix devices which are BCVs (Business Continuance Volumes). Since their usual purpose is to act as a mirror of another standard volume which can be detached for backup, they usually do not have their own volume serials and LABEL has no effect if executed against them. However, BCVs can be used as spare primary volumes.

**BCV=IGNORE** will allow BCVs to be labeled with a volume serial you specify and used as a normal Open System volume.

**PRINT=** **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being labeled. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g., PRINT=(STATUS,DIR),PRINT=UCB

## 210.07 FDRSOS VARYON/VARYOFF/VARYPATH STATEMENT

| | |
|---|---|
| **VARYON** | **TYPE=SOS** |
| **VARYOFF** | |
| **VARYPATH** | **,BCV=IGNORE** |
| | **,PRINT=STATUS** |
| | **,PRINT=(STATUS,DIR)** |
| | **,PRINT=UCB** |

### VARYON, VARYOFF AND VARYPATH STATEMENTS

These statements request a FDRSOS vary online or vary offline operation (this is not the same as a MVS VARY console command). It must be the first statement in the input; only one of these statements is allowed per execution. VARYON/VARYOFF/VARYPATH must be followed by one or more MOUNT statements with the UNIT=operand to specify the Open System devices to be varied (for VARYOFF only, the VOL= operand can be used in place of UNIT=).

The volume serial assigned to an Open System volume by the LABEL function of FDRSOS (Section 210.06) will be lost from the UCB of the device when your MVS system is reIPLed; after an IPL, the VARYON function must be executed before any FDRSOS backup, restore or print is performed. VARYON reads the stored volser from each Open System volume identified by the MOUNT statements and restores it in the UCB. Innovation recommends that you execute the VARYON function as an automatically executed task after IPL, or as the first step in any FDRSOS job.

If necessary, VARYOFF may be executed to remove the volser from the UCBs of the selected Open System volumes, making them unusable for other FDRSOS functions until another VARYON is executed. The volser remains stored on the volume itself after a VARYOFF.

The VARYPATH statement is the same as a VARYON, except that it also insures that all channel paths (CHPIDs) assigned to the referenced Open System volumes are software-enabled. This might be needed if the MVS console DS PATH command, documented in Sections 200.03 and 220.15, shows a status of * (logically off, physically on) for one or more CHPIDs for one or more Open System volumes. This might be the result of:

- A system error which set a path offline
- A MVS command of "VARY PATH(...),OFFLINE" for a Open System volume. The equivalent "VARY PATH(...),ONLINE" will fail, for the same reason that a "VARY ...,ONLINE" will fail on Open System volumes.

VARYPATH also resets some UCB flags which might prevent successful I/O; if FDRSOS jobs hang or get unexplained errors, this may solve the problem. VARYPATH will **not** change the hardware status of the CHPIDs (paths); if a CHPID is offline or has errors communicating with the EMC control unit, VARYPATH will not solve those problems.

**WARNING: Do not use the VARYPATH command unless you have the problems described above. You must execute VARYPATH only against devices which are known to be Open System volumes. The MOUNT statement for a VARYPATH command must specify only one complete unit address; multiple steps are required to VARYPATH for more than one device.**

CONTINUED. . .

## 210.07 CONTINUED. . .

**VARYON, VARYOFF AND VARYPATH STATEMENTS (continued)**

**Note:** the FDRSOS VARYON and VARYOFF functions are **not** the same as a MVS VARY console command. The Open System device will remain offline to MVS even after execution of a FDRSOS VARYON, since MVS does not natively support FBA (Fixed Block Architecture) disks like the EMC Open System disks. However, after a VARYON, the volume serial that VARYON (and LABEL) store in the UCB of the disk may be displayed by certain MVS commands (such as the console D U command) and other utilities.

**Warning: if the VARYON statement is executed with a MOUNT UNIT=* (or some other UNIT value that might include devices which are not EMC Open System disks) this might result in I/O error messages on the MVS console. FDRSOS will only attempt to VARYON devices which are generated as offline DASD devices, and uses techniques to suppress errors that result from attempting to access non-Open System devices, but in some cases this suppression may not be successful. If such error messages occur, you may be able to avoid them only by changing the UNIT= operands to include only known Open System volumes.**

**Note:** If your Symmetrix system includes BCVs for Open System volumes (see Section 200.05), you must execute a VARYON against them before you can use them with FDRSOS. However, in normal use BCVs are exact duplicates of their associated primary volume and have the same internal volser as that primary so a VARYON for a BCV volume will assign a volume serial of E#uuuu where "uuuu" is the MVS device address, e.g., E#01F0; this will allow you to identify the BCVs in a device display. BCVs are not normally used directly for FDRSOS operations; you use them indirectly to create "frozen" copies of primary volumes. However, you can use a BCV as a spare volume, assigning it a serial with the LABEL operation (see Section 210.06); if you have done so, and then reIPL your system, you must execute a VARYON with BCV=IGNORE against those BCVs in order to pickup the assigned volume serial instead of the E#uuuu serial normally assigned to BCVs. More details are in Section 200.05.

## 210.07 CONTINUED. . .

**OPERANDS**

**TYPE=SOS** Must be specified on the VARYON/VARYOFF Statement.

**BCV=IGNORE** FDRSOS will identify all Symmetrix devices which are BCVs (Business Continuance Volumes). Since their usual purpose is to act as a mirror of another standard volume which can be detached for backup, they usually do not have their own volume serials and VARYON **will assign a volume serial of E#uuuu where "uuuu" is the MVS device address, e.g., E#01F0.** However, BCVs can be used as spare primary volumes by executing a LABEL function with BCV=IGNORE to label a BCV with its own volume serial. If you reIPL your system while a BCV is in use as a spare volume, you must execute a VARYON with BCV=IGNORE against those volumes to load the true volume serial into the UCB.

**PRINT=** **STATUS** - requests that FDRSOS attempt to identify the type of Open System volume being varied. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,

PRINT=(STATUS,DIR),PRINT=UCB

## 210.08 FDRSOS ESTABLISH/RE -ESTABLISH STATEMENT

| | |
|---|---|
| **ESTABLISH** | **TYPE=SOS** |
| **RE-ESTABLISH** | **,BCV=WAIT \| <u>NOWAIT</u>** |
| | **,PRINT=STATUS**<br>**,PRINT=(STATUS,DIR)**<br>**,PRINT=UCB** |
| | **,RESTOREFROMBCV** |

### ESTABLISH AND RE-ESTABLISH STATEMENTS

These statements are used with BCVs (Business Continuance Volumes). Please read Section 200.05 for details on BCV operation and use.

ESTABLISH assigns an available BCV as a mirror of a primary volume in the same EMC Symmetrix subsystem. One or more MOUNT statements with VOL= (the primary volume serial) and BCVUNIT= (the MVS address of the BCV) are required; it is your responsibility to choose a BCV of the same size and type (Open System, FBA) as the primary. The Symmetrix hardware will copy all data on each primary volume to its BCV as a background task. ESTABLISH will fail if the BCVUNIT= address does not point to an available (unassigned) BCV volume (however, if the BCV is already assigned to the primary volume indicated, no error will result).

RE-ESTABLISH reassigns a BCV to its primary volume. Once a BCV has been ESTABLISHed, and then SPLIT (Section 210.09), RE-ESTABLISH can be used to make the BCV a valid mirror again by copying only the data which has been updated since the SPLIT occurred. One or more MOUNT statements with only VOL= (the primary volume serial) are required; FDRSOS will determine the BCV that was most recently split from this primary volume. RE-ESTABLISH will fail if the primary volume has never had a BCV assigned to it.

However, if the operand RESTOREFROMBCV is specified, the operation is essentially reversed. Data currently on the BCV volume will be copied to the primary volume. If the operation is ESTABLISH, all tracks on the BCV will be copied, but if it is RE-ESTABLISH, only those tracks on the primary which have been updated since the BCV was last SPLIT will be copied to the primary. This might be used to create an “instant backup” which will be restored a short time later and will be much faster than a normal FDRSOS restore. For example, you can preserve a volume’s contents before a test, and then restore the volume as if the test had not occurred.

## 210.08 CONTINUED. . .

The DUMP statement has an option to do an automatic RE-ESTABLISH as soon as the backup of a BCV is complete; the RE-ESTABLISH statement is not required if this option is used and the backup and automatic RE-ESTABLISH is successful. However, if the backup fails, the automatic RE-ESTABLISH is not done; you can correct the problem and resubmit backup, but if you don't intend to redo the backup, you must submit a RE-ESTABLISH step to resynchronize the BCV with its primary volume.

**Warning:** EMC provides utility software which can also issue the BCV commands (ESTABLISH, SPLIT, etc.). You must be sure that such commands are not issued while a FDRSOS function is executing, or the results may be unpredictable. Innovation recommends that the EMC Timefinder™ software not be used for Open System volumes; use the equivalent FDRSOS functions instead.

### OPERANDS

**TYPE=SOS** Must be specified on the ESTABLISH/RE-ESTABLISH Statement.

**BCV=**

**WAIT** - FDRSOS will wait until the synchronization of each BCV with its primary volume is complete. Synchronization is started for all primary volumes specified by MOUNT statements, but the step will not end until every volume is synchronized. For ESTABLISH it waits until all primary data is copied to the BCV. For RE-ESTABLISH it waits until all data that was updated on the primary since the last SPLIT is copied. The FDRSOS step will end when synchronization is complete for all volumes selected, so the step termination can be used to coordinate with other events. FDRSOS will display a FDRW20 message on the MVS console at step end.

**NOWAIT** - FDRSOS will initiate the synchronization process but will not wait for it to complete. Step end does not indicate complete synchronization and no FDRW20 message is issued.

Default is NOWAIT.

**PRINT=**

**STATUS** - requests that FDRSOS attempt to identify the type of the primary Open System volume. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB** - prints the MVS UCB (Unit Control Block) and associated control blocks for each primary volume specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g., PRINT=(STATUS,DIR),PRINT=UCB

CONTINUED. . .

## 210.08 CONTINUED. . .

**OPERANDS (continued)**

**RESTOREFROMBCV** If specified, for the primary and BCV volumes specified by the following MOUNT statements, the BCV will be assigned to the primary volume but data tracks will be copied from the BCV to the primary volume, the reverse of the normal operation. For an ESTABLISH all BCV tracks are copied to the primary, while a RE-ESTABLISH copies to the primary only tracks which were changed on the primary or the BCV since the last SPLIT.

By default, for the primary and BCV volumes specified by the following MOUNT statements, the BCV will be assigned to the primary volume and data tracks will be copied from the primary to the BCV. For an ESTABLISH all primary tracks are copied to the BCV, while a RE-ESTABLISH copies to the BCV only tracks which were changed on the primary volume since the last SPLIT.

**Warning:** RESTOREFROMBCV must be used carefully. Remember that any updated tracks on the primary will be reset to the contents of those tracks on the BCV. This is usually the original contents of those tracks, but if you have written any data to the BCV while it was SPLIT, those updated BCV tracks will also be written back to the primary volume. If the FDRSOS volume serial of the BCV was changed while it was split (via a LABEL statement) that volser will be copied from the BCV to the primary; any FDRSOS jobs referencing the old volume serial will fail.

## 210.09 FDRSOS SPLIT STATEMENT

| | | |
|---|---|---|
| **SPLIT** | **TYPE=SOS**<br>**,BCV=WAIT \| NOWAIT** | **,PRINT=STATUS**<br>**,PRINT=(STATUS,DIR)**<br>**,PRINT=UCB** |

### SPLIT STATEMENT

This statement is used with BCVs (Business Continuance Volumes). Please read Section 200.05 for details on BCV operation and use.

SPLIT detaches a BCV volume from its currently assigned primary volume. One or more MOUNT statements with only VOL= (the primary volume serial) are required; FDRSOS will determine the BCV that is currently ESTABLISHed (assigned) to this primary volume. This is usually a very quick operation. But if the primary and BCV volumes are not completely synchronized, perhaps because the ESTABLISH was recently issued, SPLIT will be **rejected** by the Symmetrix hardware and FDRSOS will wait for the synchronization to complete and reissue the SPLIT, so this could take significantly longer.

If the BCV associated with the primary volume selected is **already split**, the SPLIT statement will cause an automatic RE-ESTABLISH (to bring the BCV into synchronization with the primary) and then a SPLIT; this assumes that you want the BCV to be current as of the time of the SPLIT. Even if BCV=NOWAIT is specified or defaulted, SPLIT will wait until the RE-ESTABLISH is complete.

**Warning:** EMC provides utility software which can issue **also issue** the BCV commands (ESTABLISH, SPLIT, etc.). You must be sure that such commands are not issued while a FDRSOS function is executing, or the results may be unpredictable. Innovation recommends that the EMC Timefinder™ software not be used for Open System volumes; use the equivalent FDRSOS functions instead.

## 210.09 CONTINUED. . .

**OPERANDS**

**TYPE=SOS** Must be specified on the SPLIT Statement.

**BCV=**

**WAIT** - FDRSOS will wait until the SPLIT of each BCV from its primary volume is complete. This is usually a very quick operation, but if one or more of the primary and BCV volumes are not already in sync, SPLIT will not complete until the synchronization ends on every volume. The FDRSOS step will end when the SPLIT is complete for all volumes selected, so the step termination can be used to coordinate with other events. FDRSOS will display a FDRW20 message on the MVS console at step end.

**NOWAIT** - FDRSOS will initiate the SPLIT process but will not wait for it to complete. Step end does not indicate completion and no FDRW20 message is issued.

Default is NOWAIT.

**PRINT=**

**STATUS** - requests that FDRSOS attempt to identify the type of the primary Open System volume. If possible, it will display the type of Open System which created it, and other pertinent information about the format and contents of the volume. For more information, see the sample displays in Section 200.04.

**(STATUS,DIR)** - in addition to the PRINT=STATUS displays, for some Open System platforms it will display the files and subdirectory names in the root directory of each volume or logical volume.

**UCB -** prints the MVS UCB (Unit Control Block) and associated control blocks for each unit address specified. This is a diagnostic tool, used primarily to investigate disk access problems.

**Note:** PRINT=STATUS and PRINT=UCB must be specified separately if both are required, e.g.,
PRINT=(STATUS,DIR),PRINT=UCB

## 210.10 FDRSOS LOCALBACKUP STATEMENT

| | | |
|---|---|---|
| LOCALBACKUP<br>LB | TYPE=INIT \| UPDATE | ,MAX#PROF=nnn |
| | ,BCV=IGNORE | ,MAXBACKUPSIZE=nnnn |
| | ,DELMIGRATED=YES \| NO | ,MAXFILESIZE=nnnn \| UNLIMITED |
| | ,DYNADDPROF=YES \| NO | ,RETAINLB=YES \| NO |
| | ,MINALLOC%=nnn | ,WRAP=YES \| NO |
| | ,MAX#BACKUPS=nnn | |

### LOCALBACKUP STATEMENT

This statement is used to initialize EMC Symmetrix disk volumes for use with FDR/UPSTREAM as FDRSOS local backup volumes, or to update default parameters on an existing FDRSOS local backup volume. See Section 220.13 for more information on FDR/UPSTREAM local backups.

The disk(s) to be processed are identified by one or more MOUNT statements which follow. These disks must have been previous labeled by a FDRSOS LABEL statement. You must not change the volume serial of a local backup volume once it is in use; **if you do so, all existing backups on the volume will be lost.**

The operands on the LOCALBACKUP statement set options stored on the FDRSOS local backup volume which control FDR/UPSTREAM use of the local backup volume. For TYPE=INIT, all the operands (or their defaults) are used to set the options on the new local backup volume; TYPE=UPDATE will update options on an existing local backup volume and only those operands which appear on the LOCALBACKUP statement will be modified (other options keep their previous values).

Except for DYNADDPROF= and MAX#PROF=, all the LOCALBACKUP operands provide defaults for processing the FDR/UPSTREAM profiles which have backups on the FDRSOS local backup volume. Any profile newly added to the local backup volume will get these defaults. Facilities are provided with FDR/UPSTREAM on the Open System to add and delete profiles, change the values associated with any profile and change the defaults.

TYPE=INIT will normally initialize the entire volume for FDRSOS local backup use, but you can specify a SIZE= operand on each MOUNT statement to limit the space used to less than the physical size of the volume. The volume is initialized as a DOS disk partition with additional data used by local backup; SIZE= causes only the first part of the volume to be formatted as a DOS partition. The additional space on the volume can be used for other purposes if the Open System supports DOS partition tables; this includes all versions of Windows and Novell Netware but **does not include UNIX systems**.

## 210.10 CONTINUED. . .

To avoid overlaying valid data, TYPE=INIT will fail if the disk currently has any file system format which is recognized by FDRSOS, including local backup data. If you want to take a volume which previously held data and use it for FDRSOS local backups, you will need to ERASE at least the first 1000 blocks of data (see ERASE in Section 210.05).

TYPE=UPDATE will not normally change the size of the existing FDRSOS local backup volume. However, if you specify a SIZE= operand on a MOUNT statement, and the value is larger than the SIZE= used to initialize the volume, the local backup volume will be updated to use the new size for local backups. Note that there is no checking to be sure that the additional space is not already used for other data. TYPE=UPDATE will fail if the disk is not already formatted for local backups.

### SUN SPARC SYSTEMS

When the local backup disk is attached to a SUN SOLARIS system running on a SUN SPARC processor, there are special procedures (they do not apply to SUN SOLARIS on an INTEL-type system):

1. unless the disk is brand new, never before used, you should erase the disk completely with the FDRSOS ERASE function (Section 210.05).
2. from the SUN system, you must format the disk volume and specify a volume name of "FDRSOS". The SUN SOLARIS commands to do so are documented in Appendix F.
3. use the FDRSOS LOCALBACKUP function to make the formatted disk as a FDRSOS local backup.

**Warning: if you do not follow this procedure, in this order, the local backup disk will be usable from non-SUN systems but not from the SUN system. Disks initialized with this procedure will be usable on the SUN and most other Open Systems.**

### OPERANDS

**TYPE=** Identifies the type of operation and must be specified on the LOCALBACKUP Statement.

**INIT** – performs a full initialization of a new FDRSOS local backup volume. The operands on this statement, or their defaults, are used to set defaults on the local backup volume for use with new FDR/UPSTREAM profiles.

**UPDATE** – updates an existing FDRSOS local backup volume with new default values. Only values for the operands you specify are changed; omitted operands will retain their previous values.

## 210.10 CONTINUED. . .

**BCV=IGNORE** If the volume you are trying to initialize is defined as a BCV (Business Continuance Volume) in the Symmetrix configuration, even if it is not currently ESTABLISHed as a copy of any primary volume, FDRSOS will normally reject any attempt to initialize the BCV for local backups. If you are sure that you want to use the BCV for local backups, and will never ESTABLISH that BCV as a copy of another volume, specify BCV=IGNORE to bypass the check and initialize it for local backups.

**Note:** if a BCV was initialized for local backups with BCV=IGNORE, all subsequent VARYON statements executed against that volume must also specify BCV=IGNORE. Innovation recommends that the Symmetrix configuration be updated to make the volume a non-BCV to avoid these requirements.

**DELMIGRATED=** If FDR/UPSTREAM is unable to find sufficient space on the FDRSOS local backup volume for a new backup, this controls what action it will take.

**YES** –delete backups on the volume for the current profile which have been copied (migrated) to MVS tape or disk by FDR/UPSTREAM, then try the allocation again.

**NO** – migrated backups will not be automatically deleted. If there is insufficient free space on the local backup volume you may need to manually delete backups.

Default is NO.

**DYNADDPROF=YES** – if you do a backup to this FDRSOS local backup volume under a FDR/UPSTREAM profile which has not previously been used on this volume, it will be automatically added to the control records on the volume. The backup characteristics of the profile will be initially set to the values specified by other operands on the LOCALBACKUP statement, but they can be modified by FDR/UPSTREAM on the Open System.

**NO** – Only FDR/UPSTREAM profile names which have been manually added to the control records on this FDRSOS local backup volume can do backups to the volume. Profile names are added by FDR/UPSTREAM on the Open System.

Default is YES.

## 210.10 CONTINUED. . .

**MINALLOC%=** When FDR/UPSTREAM allocates space for a backup on a FDRSOS local backup volume, the actual space required is not known since compression will reduce the requirement but cannot be predicted. If available, the uncompressed size is allocated and any unused space released at the end. But if that much space is not available, it may allocate a smaller backup file anticipating that the backup may fit. This operand controls the minimum percentage of the uncompressed size which can be allocated. You may need to adjust it depending on your results with the FDR/UPSTREAM compression.

**nnn** – specifies the minimum percentage of the total uncompressed size which must be allocated for the backup to the local backup volume. Valid values are 10 to 100.

Default is 50.

**MAX#BACKUPS=**This controls the maximum number of backups which may be recorded on the FDRSOS local backup volume for a given FDR/UPSTREAM profile. Under a given FDR/UPSTREAM profile, backups older than the maximum will automatically be deleted to reclaim space when a new backup is created.

**nnn** – specifies the number of backups per profile allowed. Valid values are 1 to 255.

Default is 10, unless RETAINLB=NO then 0.

**MAX#PROF=** This controls the maximum number of FDR/UPSTREAM profiles which may have backups recorded on this FDRSOS local backup volume. Profiles are added automatically the first time that they do a backup to this local backup volume if DYNADDPROF=YES was specified or defaulted; profile names can also be added manually using FDR/UPSTREAM on the Open System.

**nnn** – specifies the number of backups per profile allowed. Valid values are 1 to 100.

Default is 100.

**MAXBACKUPSIZE=**This specifies the maximum size in megabytes (MB) of a backup that will be allowed on this FDRSOS local backup volume. If the total uncompressed size of the backup exceeds this size, FDR will allocate the maximum size on the local disk. If WRAP=YES is specified or defaulted, FDR will reuse the backup internally to send all the remaining files through the local disk to MVS. If WRAP=NO is specified, the remaining files will be sent over the network.

Note: If RETAINLB=NO is specified, MAXBACKUPSIZE need not be over the default of 100MB. Performance may suffer if a value under 50MB is specified for a large backup.

**nnnn** – the maximum backup size in megabytes (MB). The value must be from 1 to 4095.

Default is 100 MB.

## 210.10 CONTINUED. . .

**MAXFILESIZE=** This specifies the maximum size in megabytes (MB) of an individual Open System file that will be allowed on this FDRSOS local backup volume. Files which exceed the size will be transmitted to FDR/UPSTREAM over your network instead, although smaller files may be written to the local backup.

**nnnn** – the maximum file size in megabytes. The value must be from 1 to 4096. UNLIMITED - there is no limit to the file size.

Default is UNLIMITED.

**RETAINLB=** specifies whether the local backup is to be retained after the backup completes. As a default the local copy will be retained if the backup was not restarted and it was large enough to contain all the files written (did not wrap).

**YES** – specifies to keep the local backup.

**NO** – specifies that the local backup is to be deleted after a successful backup. If the backup fails and is restartable the local backup will be retained for the first restart.

Default is YES.

**WRAP=** specifies whether FDR/UPSTREAM on the PC/Open System can re-use the backup file on the local disk if there is insufficient space to contain the entire backup.

**YES** – FDR/UPSTREAM can re-use the backup. The backup will be deleted at the end of the backup if it wraps.

**NO** – FDR/UPSTREAM cannot re-use the backup. If the backup fills up, all the remaining files will be sent through the network.

Default is YES.

## 210.15 FDRSOS MOUNT STATEMENT

| | | |
|---|---|---|
| **MOUNT** | **BACKUPDSN=**dsname | **(RESTORE only)** |
| | **,BCVUNIT=**unit | **(ESTABLISH only)** |
| | **,SETVOL=**volser | **(LABEL only)** |
| | **,SIZE=**nnnnn | **(LOCALBACKUP only)** |
| | **,TAPEDD=**x | **(DUMP/RESTORE/ERASE/PRINT only)** |
| | **,UNIT=**unit | **(LABEL/VARYON/VARYOFF/VARYPATH only)** |
| | **,VOL=**volser | **(all except VARYON and VARYPATH)** |

**MOUNT STATEMENT**

The MOUNT statement identifies Open System volumes to be processed by FDRSOS . For DUMP, RESTORE, ERASE and PRINT operations, either MOUNT statements or DISKx DD statements can be used to identify the volumes. For all other operations, one or more MOUNT statements are required.

Please review the examples in Sections 210.20-25 for guidance on the correct usage of the various operands of MOUNT.

**OPERANDS**

**BACKUPDSN=** For RESTORE only, optionally specifies the name of a cataloged backup data set. If the data set is a GDG, a relative generation number may be included, e.g., BACKUPDSN=BACK.OPEN1(-1).The data set will be dynamically allocated by FDRSOS and it will be restored to the Open System volume specified by VOL=; in this case, no TAPEx DD statement is required. If BACKUPDSN= is omitted, then TAPEDD= and a TAPEx DD statement must be specified (unless a DISKx DD is provided, in which case the MOUNT statement is not required).

**BCVUNIT=** Used only for ESTABLISH operations, specifies the S/390 device address of an Open System BCV (Business Continuance Volume) in the same Symmetrix subsystem as the primary volume identified by the VOL= operand. The value is a 4-digit S/390 device address, e.g., UNIT=01E0 or UNIT=25C6. The device at the address given will be validated as an EMC Open System BCV and will be ESTABLISHed (assigned) as a BCV mirror of the primary volume.

**SETVOL=** Used only for LABEL operations, specifies the volume serial to be assigned to the Open System device identified by the UNIT= operand.

**SIZE=** Used only for LOCALBACKUP operations, specifies the size in MB (Megabytes) of the Open System volume to be initialized as a FDRSOS local backup volume for use with FDR/UPSTREAM. For a TYPE=INIT operation, if SIZE= is less than the size of the volume, FDRSOS will initialize the beginning of the volume as a DOS partition of that size. The additional space on the volume can be used for other purposes if the Open System supports DOS partition tables; this includes all versions of Windows and Novell Netware **but does not include UNIX systems**. For a TYPE=UPDATE operation, the existing local backup area will be expanded to the size indicated; this can be used only if the volume was initialized with a smaller SIZE= value.

Default for TYPE=INIT is the size of the volume, minus a few cylinders for system use; the entire volume will be initialized as a local backup. For TYPE=UPDATE, the existing size of the local backup volume is preserved.

## 210.15 CONTINUED. . .

**TAPEDD=** If MOUNT statements are used with DUMP, RESTORE, ERASE, or PRINT, and the FDRSOS JCL contains more than one TAPEx DD statement, this operand is required to specify which TAPEx DD is to be used for the Open System volume identified by this MOUNT statement:

- For DUMPs, it specifies the output backup data set. If omitted, TAPE1 is assumed.
- For RESTOREs, it specifies the input backup to be restored. Either TAPEDD= with a TAPEx DD statement or the BACKUPDSN= operand on MOUNT may be used.
- For ERASEs, TAPEDD=is required only if MAXTASKS=n is specified on the ERASE statement; there must be a TAPEx DD DUMMY and a matching SYSPRINx DD specifying the destination for messages for this ERASE operation.
- For PRINTs, it specifies the output print data set. If omitted, TAPE1 is assumed.

The operand is the single character "x", e.g., TAPEDD=3 specifies TAPE3. A unique value must be used on each MOUNT statement since each Open System volume must be directed to a different TAPEx DD.

## 210.15 CONTINUED. . .

**UNIT=**

Specifies the S/390 device address(s) of Open System volume(s) to be processed for LABEL, VARYON, VARYOFF and VARYPATH operations (it is not used with DUMP, RESTORE, ERASE or PRINT). For VARYOFF only, the VOL= may alternately be used to identify the volume.

For LABEL, the value is a 4-digit S/390 device address, e.g., UNIT=01E0 or UNIT=25C6. The device at the address given will be validated as an EMC Open System volume and will be labeled with the value specified by the SETVOL= operand.

For VARYON and VARYOFF, the value may be a 4-digit S/390 device address (e.g., UNIT=01E0) or may be a prefix followed by an asterisk, e.g., UNIT=01E* or UNIT=03*. You may also specify a list of units and/or prefixes by enclosing the list in parenthesis, e.g., UNIT=(01E*,01F*,01D5). You may even use UNIT=* to specify all DASD device addresses in your system. All device addresses in the I/O configuration matching the address or prefix will be validated as EMC Open System devices (offline to MVS and responding to FBA commands); for those that pass, FDRSOS will read the volume serial recorded on the volume by the LABEL function and store it in the MVS UCB for the device.

For VARYPATH, the value is a 4-digit S/390 device address, e.g., UNIT=01E0 or UNIT=25C6. All defined paths (CHPIDs) for the device are enabled (unless they are physically offline) and certain UCB flags that may inhibit I/O are reset.

UNIT= may be used with ERASE operations, if the volume to be erased has not been previously assigned a pseudo volume serial with a LABEL statement. However, Innovation recommends labeling the volume and using the VOL= parameter to identify the volume to be erased.

**WARNING: 4-digit device addresses must be used even if your operating system only supports 3-digit addresses for other functions; specify a leading zero if required.**

## 210.15 CONTINUED. . .

**VOL=** Specifies the volume serial of the Open System volume to be processed by all FDRSOS operations except VARYON and VARYPATH. A volume serial can be used to identify an Open System volume only if a FDRSOS LABEL or VARYON operation has been run since the LAST IPL to store the volume serial in the MVS UCB.

For some operations, a volume serial prefix may be specifying by following it with an asterisk, e.g., VOL=OPEN*. You can also specify a list of volsers and/or prefixes by enclosing the list in parenthesis, e.g., VOL=(OPEN*,EMC123,NTA1*). All of the specified volumes will be processed.

When used with a Symmetrix BCV (Business Continuance Volume), via the ESTABLISH, RE-ESTABLISH, SPLIT, and DUMP BCV=USE commands, VOL= must specify the FDRSOS volume serial of the primary volume. For ESTABLISH, BCVUNIT= must also be specified to identify the BCV to be assigned to the primary volume. For other operations, FDRSOS will automatically identify the BCV associated with the primary volume identified by VOL=.

The following table summarizes the use of VOL= and UNIT= with the various FDRSOS operations:

| Operation | VOL= | UNIT= | Notes |
|---|---|---|---|
| VARYON | not used | required | |
| VARYPATH | not used | required | |
| VARYOFF | optional | optional | Either VOL= or UNIT= is required. VOL= can specify a prefix and/or a volume list |
| LABEL | not used | required | |
| ESTABLISH | required | not used | specifies a primary volume |
| RE-ESTABLISH | required | not used | VOL= specifies primary volumes, can specify a prefix and/or a volume list |
| SPLIT | required | not used | VOL= specifies primary volumes, can specify a prefix and/or a volume list |
| DUMP | required | not used | serial can also be specified on DISKx DD. To backup BCV, specify primary volume |
| RESTORE | required | not used | serial can also be specified on DISKx DD |
| PRINT | required | not used | serial can also be specified on DISKx DD |
| ERASE | required | not used | serial can also be specified on DISKx DD |
| LOCALBACKUP | required | not used | |

## 210.16 FDRSOS SELECT/EXCLUDE STATEMENT

| | |
|---|---|
| **SELECT**<br>**S** | **FROMBLK=**nnnnn,**TOBLK=**nnnnn<br>**,TAPEDD=**x |
| **EXCLUDE**<br>**X** | **,VOL=**volser |

### SELECT AND EXCLUDE STATEMENTS

The SELECT and/or EXCLUDE statements are required when TYPE=PARTIAL was specified on the DUMP, RESTORE, or PRINT statement to specify the data to be dumped, restored, or printed.

SELECT identifies a range of data blocks to be processed and EXCLUDE identifies data blocks from within those selected by SELECT statements which are not to be processed. Multiple SELECT and EXCLUDE statements may be input to specify muliple ranges.

Open System volumes are formatted as a series of 512 byte data sectors, but are usually read and written as 4K (4096) byte blocks, so the SELECT and EXCLUDE statements specify a range of 4K blocks by block number. Block numbers are relative to zero, so 0 will be the first 4K block on the volume, 1 the second 4K block, etc.

**The control statements are always scanned in the order in which they were input, so in general, EXCLUDE statements should precede SELECT statements.**

### OPERANDS

**FROMBLK=**
**TOBLK=**

Specifies the beginning and end of a range of 4K data blocks to be processed for backup, restore, or print. The values for FROMBLK= and TOBLK= are block numbers relative to zero. In other words, FROMBLK=0,TOBLK=499 will select the first 500 4K blocks on the volume. TOBLK= must be equal or greater than FROMBLK=.

**TAPEDD=**

If the FDRSOS JCL contains more than one TAPEx DD statement, TAPEDD= may be used to specify to which TAPEx DD this SELECT/EXCLUDE applies. The operand is the single character "x", e.g., TAPEDD=3 specifies TAPE3.

**VOL=**

If more than one Open System volume is being processed in this step, VOL= specifies the volume serial of the Open System volume to which this SELECT/EXCLUDE applies. It may specify a complete volser (e.g., VOL=OPEN23) or a volser prefix (followed by an asterisk, e.g., VOL=OPEN*)

Either TAPEDD= or VOL= may be specified to identify the volume to which this statement applied. If both are omitted, this statement will apply to all Open System volumes being processed in this step.

## 210.17 FDRSOS PRINT MODIFIERS

FDRSOS includes the ability to scan for specified ASCII or hexadecimal strings of data while printing data from Open System volumes (the PRINT statement is described in Section 210.04). Only the data sectors containing the specified string will be printed. If scan statements are omitted, all selected data sectors are printed.

### $TRKPRIN DD STATEMENT

In order to specify scan statements, an additional DD statement must be added to the FDRSOS step. $TRKPRIN contains the scan statements and is usually a DD * (input stream) data set, separate from the normal SYSIN DD statement. The input may include a DEFAULT statement and/or a SCAN statement.

### DEFAULT STATEMENT

**DEFAULT** **CHECKSUM** | **NOCHECKSUM**

**,LINECOUNT=** nnn

**,NOZEROBLOCKS**

**,SEP** | **NOSEP**

The DEFAULT statement is optional, but must come first if present. It specifies options for the PRINT operation and can be present even if the SCAN statement is omitted to control non-scan PRINT functions.

### OPERANDS

**CHECKSUM**
**NOCHECKSUM**

**CHECKSUM** specifies that a hexadecimal checksum be calculated and printed in every separator line (see SEP below). The checksum is calculated by adding every 4-byte word in the data; the value is a running checksum, meaning that the value printed at any point is the sum of all 4-byte words printed since the beginning of the PRINT operation. This checksum might be useful for identifying sectors which have changed, or for quickly verifying that a FDRSOS restore has restored the exact same data that existed before the backup. If the last checksum printed in 2 PRINT operations is the same, the data printed is probably the same (it is possible to have offsetting changes in data that result in the same checksum, but this is usually not the case).
**NOCHECKSUM** suppresses the checksum calculation.

The default is CHECKSUM.

**LINECOUNT=** specifies the lines per page to be printed on the TAPEx DD statement.
The default is 58.

**NOZEROBLOCKS** If specified, sectors which are entirely binary zeros will not be printed. This is useful to restrict the printout to just sectors containing significant information.

**SEP**
**NOSEP**

**SEP** indicates that every 512 data sector printed will have a separator line, identifying the sector number and, if requested, displaying the current checksum including that sector.
**NOSEP** requests that the separator line be printed only at the beginning of each group of 64 data sectors, identifying the range of sectors and displaying the current checksum through the last sector range.

210.17 CONTINUED . . .

**SCAN STATEMENT**

**SCAN** **ARGUMENT**=string | 'string' | X'hexdata'

**,CASE=UPPER | LOWER**

The SCAN statement is optional. If present, it specifies that only data sectors that contain a specified ASCII or hexadecimal string should be printed.

**OPERANDS**

**ARGUMENT=** specifies the string which should be scanned. In can be in one of 3 formats:
**string** - is an alphanumeric ASCII character string. It cannot contain blanks or special characters.
**'string'** - is an ASCII string in quotes. It can contain any valid ASCII characters including blanks.
**X'hexdata'** - is a string of data in hexadecimal. It must contain an even number of valid hexadecimal digits (0-9, A-F).

**Note:** ASCII strings are actually entered on the SCAN statement in EBCDIC; they are translated to ASCII for use in the scan.

**CASE=** If an ASCII string was specified for ARGUMENT=, CASE specifies translation of that string.
**UPPER** - the string is translated to upper case ASCII
**LOWER** - the string is translated to lower case ASCII

The default is UPPER.

Note: mixed-case ASCII strings can be entered only by entering the hexadecimal equivalent of the ASCII characters using the ARGUMENT=X'hexdata' option.

## 210.20 FDRSOS INITIALIZATION EXAMPLES

### LABEL OPEN SYSTEM VOLUMES

Assign volume serials to a number of Open System volumes. The volser specified is stored in an area of the volume reserved by EMC for FDRSOS use and is also stored in the UCB (MVS Unit Control Block) of the device, allowing it to be referenced in JCL by other FDRSOS jobs. The LABEL function needs to be executed only once for a given Open System volume unless you need to change the label or the label is lost because of EMC hardware reconfiguration or replacement. PRINT=STATUS requests that FDRSOS identify the type and contents of the Open System volume, to document that you are labeling the correct volume. LABEL will also provide additional diagnostics if it cannot access the specified unit as an Open System volume.

```
//LABEL      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//SYSIN       DD     *
   LABEL     TYPE=SOS,PRINT=STATUS
   MOUNT     UNIT=01E0,SETVOL=OPEN#1
   MOUNT     UNIT=01E1,SETVOL=OPEN#2
```

### VARYON OPEN SYSTEM VOLUMES

The volume serial of an Open System volume stored in the UCB (MVS Unit Control Block) is not preserved across IPLs. The VARYON function reads the volser from the FDRSOS area of the volume and stores it in the UCB again, allowing it to be referenced in JCL by other FDRSOS jobs. This example executes the VARYON for two ranges of Open System Devices (01E0-01EF, 01F0-01FF). The VARYON function is not the same as a MVS VARY console command; the devices will remain offline to MVS. PRINT=(STATUS,DIR) requests that FDRSOS identify the type and contents of each Open System volume processed; for some volume formats the contents of the root directory is also displayed.

```
//VARYON     EXEC   PGM=FDRSOS,REGION=0M
//STEPLIB     DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD    SYSOUT=*
//SYSUDUMP    DD    SYSOUT=*
//SYSIN       DD    *
    VARYON    TYPE=SOS,PRINT=(STATUS,DIR)
    MOUNT     UNIT=01E*
    MOUNT     UNIT=01F*
```

210.20 CONTINUED . . .

## VARYON ALL OPEN SYSTEM VOLUMES

The volume serial of an Open System volume stored in the UCB (MVS Unit Control Block) is not preserved across IPLs. The VARYON function reads the volser from the FDRSOS area of the volume and stores it in the UCB again, allowing it to be referenced in JCL by other FDRSOS jobs. This example executes the VARYON for several ranges of addresses (only offline DASD devices are examined). Since the FDRSOS control statements are completely specified in the PARM data, this example could be converted into a PROC for a started task to be executed automatically at IPL time.

If your Symmetrix system includes BCVs (Business Continuance Volumes) for Open System volumes, you must include the S/390 addresses of those BCVs in the VARYON processing, in order to make them available for FDRSOS processing; they will receive a generated volume serial of E#uuuu where "uuuu" is the S/390 device address, e.g., E#01F0.

**Warning: VARYON with MOUNT UNIT=* might result in I/O error messages on the MVS console. FDRSOS will only attempt to VARYON devices which are generated as offline DASD devices, and uses techniques to suppress errors that result from attempting to access non-Open System devices, but in some cases this suppression may not be successful. If such error messages occur, you may be able to avoid them by changing the UNIT= operands to include only known Open System volumes.**

```
//VARYON        EXEC      PGM=FDRSOS,REGION=0M,
//    PARM=' VARYON TYPE=SOS/MOUNT UNIT=(01E*,01F*,03C*,03D*)'
//STEPLIB        DD       DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT       DD       SYSOUT=*
//SYSUDUMP       DD       SYSOUT=*
//SYSIN          DD       DUMMY
```

## VARYOFF OPEN SYSTEM VOLUMES

You might want to make certain Open System volumes unavailable to FDRSOS. The VARYOFF function removes the volser of specified Open System volumes from the MVS UCB. A subsequent VARYON would be required to make them available again. As shown, the volumes involved can be specified by unit address or volser. The VARYOFF function is not the same as a MVS VARY console command; Open System devices are always offline to MVS.

```
//VARYOFF    EXEC     PGM=FDRSOS,REGION=0M
//STEPLIB     DD      DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//SYSIN       DD      *
   VARYOFF   TYPE=SOS
   MOUNT     UNIT=01E*
   MOUNT     VOL=OPEN2
```

210.20 CONTINUED . . .

## VARY PATHS FOR AN OPEN SYSTEM VOLUME

When some paths (CHPIDs) are not available for an Open System volume, or I/O to a volume is not successful for an undetermined reason, the VARYPATH command may be able to correct the problem. It resets some flags which may prevent successful I/O, and enables all paths less they are physically varied offline. The MVS console command DS PATH (see Section 220.15) will show path status; a status of * for a CHPID indicates it is logically offline but physically online. The MOUNT statement must specify only a single Open System device.

```
//VARYPATH  EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB    DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD     SYSOUT=*
//SYSUDUMP   DD     SYSOUT=*
//SYSIN      DD     *
     VARYPATH TYPE=SOS
     MOUNT    UNIT=01E0
```

## INITIALIZE FDR/UPSTREAM LOCAL BACKUP

As described in Section 220.13, a FDRSOS LOCALBACKUP function is required to initialize an Open System volume in the Symmetrix as a FDRSOS local backup volume for use with FDR/UPSTREAM. This is an example of such a LOCALBACKUP initialization with all defaults taken. A LABEL step is shown since the volume to be initialized must have a FDRSOS volume serial before it can be initialized as a local backup; omit this step if the volume has previously been labeled.

```
//LABEL      EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB     DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD   SYSOUT=*
//SYSUDUMP    DD   SYSOUT=*
//SYSIN       DD   *
     LABEL    TYPE=SOS
     MOUNT    UNIT=01FC,SETVOL=LOCAL1
//LOCALB     EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB     DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD   SYSOUT=*
//SYSUDUMP    DD   SYSOUT=*
//SYSIN       DD   *
     LOCALBACKUP TYPE=INIT
     MOUNT    VOL=LOCAL1
```

210.20 CONTINUED . . .

**INITIALIZE FDR/UPSTREAM LOCAL BACKUPS**

This is an example of a LOCALBACKUP initialization with all operands specified. Two volumes are initialized; LOCAL2 will be initialized to use its entire size (as reported by the Symmetrix) but LOCAL3 will only use the first 2GB for local backup storage.These volumes were previously used for other purposes, so the step ERASE erases the first 1000 blocks of storage on the volumes in order to erase any volume Ids and file structures so that the local backup initialization can complete successfully.

**Warning: ERASE will make all previous data on the volume unusable. Be sure that the data is not required before executing this initialization.**

```
//ERASE      EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB      DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD   SYSOUT=*
//SYSUDUMP     DD   SYSOUT=*
//SYSIN        DD   *
   ERASE TYPE=PARTIAL
   SELECT FROMBLK=0,TOBLK=999
   MOUNT VOL=LOCAL2
   MOUNT VOL=LOCAL3
//LOCALB      EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB      DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD   SYSOUT=*
//SYSUDUMP     DD   SYSOUT=*
//SYSIN        DD   *
    LOCALBACKUP TYPE=INIT,
       DYNADDPROF=NO,
       DELMIGRATED=YES,
       MAX#BACKUPS=5,
       MAX#PROF=10,
       MAXBACKUPSIZE=800,
       MAXFILESIZE=25
    MOUNT     VOL=LOCAL2
    MOUNT     VOL=LOCAL3,SIZE=2048
```

**UPDATE FDR/UPSTREAM LOCAL BACKUP**

An existing FDRSOS local backup volume is updated with new default values. Only the values of the operands specified will be changed; other values will retain their previous value. The SIZE= operand causes a volume which was previously initialized to use only part of its space for local backups to expand (see previous example); it will now use 4GB for local backups.

```
//LOCALB       EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB      DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD   SYSOUT=*
//SYSUDUMP     DD   SYSOUT=*
//SYSIN        DD   *
    LOCALBACKUP TYPE=UPDATE,MAXBACKUPSIZE=1000
    MOUNT     VOL=LOCAL3,SIZE=4096
```

Warning: FDRSOS does not check if the new space allocated to the local backup is currently in use by the Open System.

## 210.21 FDRSOS BACKUP EXAMPLES

**Note:** for examples of FDRSOS backups of Symmetrix BCV volumes, see Section 210.25.

### DUMP ONE VOLUME

Dump an Open System disk volume to 3490E tape cartridges, creating 2 copies. The TAPE11 DD can be omitted if only one copy is required. A MOUNT statement is used to specify the volume to be backed up. A volume count of 99 is specified on the TAPE DD statements in case more than 5 tape volumes are required.

```
//DUMP     EXEC PGM=FDRSOS,REGION=0M
//STEPLIB   DD  DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT  DD  SYSOUT=*
//FDRSUMM   DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//TAPE1     DD  UNIT=3490,DSN=BACKUP.VE#01E0.C1,
//         VOL=(,,,99),DISP=(,CATLG)
//TAPE11    DD  UNIT=3490,DSN=BACKUP.VE#01E0.C2,
//         VOL=(,,,99),DISP=(,CATLG)
//SYSIN     DD  *
    DUMP    TYPE=FULL
    MOUNT   VOL=E#01E0
```

The report on SYSPRINT will look similar to:

```
FDR007    STARTING TIME OF FULL VOL DUMP -- 16.18.11 -- UNIT=01E0      ,IN=D#E#01E0,OUTPUT=TAPE1
FDR007    ENDING   TIME OF FULL VOL DUMP   -- 16.30.08 -- UNIT=01E0      ,IN=D#E#01E0,OUTPUT=TAPE1
FDR122    OPERATION STATISTICS FOR SOS VOLUME..............................................E#01E0
FDR122                          BYTES ON VOLUME ..........................................4,355,850,240
FDR122                          DATASETS PROCESSED.......................................0
FDR122                          BYTES READ FROM DASD.....................................4,355,850,240
FDR122                          BYTES ON BACKUP .........................................4,364,783,536
FDR122                          COMPRESSION SAVINGS (%) ..................................0
FDR122                          DASD SECTORS BACKED UP ..................................8,507,520
FDR122                          BACKUP BLOCKS WRITTEN ...................................79,760
FDR122                          DASD EXCPS ..............................................8,863
FDR122                          BACKUP FILE EXCPS .......................................8,864
FDR122                          CPU TIME (SECONDS) ......................................14.040
FDR122                          ELAPSED TIME (MINUTES) ..................................12.4
FDR122                          BACKUP TIME(EXCLUDING MOUNTS)............................12.0
FDR122                          BACKUP COPY 1 ON TAPE DSN=BACKUP.VE#01E0.C1
FDR122                                                VOL=900013,900014,900015
FDR122                          BACKUP COPY 2 ON TAPE DSN=BACKUP.VE#01E0.C2
FDR122                                                VOL=900022,900023,900024
FDR002    FDR DUMP SUCCESSFULLY COMPLETED VOL=E#01E0
```

and the report on FDRSUMM will be similar to:

| VOLSER | COMP CODE | ELAPSED TIME(MIN) | VOLUME SIZE | DASD BYTES READ FROM VOL | BYTES ON BACKUP FILE | COMP-PRESS | SECTORS DUMPED |
|---|---|---|---|---|---|---|---|
| E#01E0 | 0 | 12.0 | 4,355,850,240 | 4,355,850,240 | 4,364,783,536 | 0% | 8,507,520 |

Reports from other dump and restore operations in these examples will also be similar.

210.21 CONTINUED . . .

**DUMP MULTIPLE VOLUMES SERIALLY**

Dump three Open System disk volumes sequentially to 3590 (Magstar) tape cartridges, using IDRC hardware compression. The 3 backups will be stacked on the tape as consecutive tape data sets. DISKx DD statements are used to specify the volumes to be backed up; each DISKx will be backed up to the corresponding TAPEx (e.g., DISK3 to TAPE3). OPENSYS must be a esoteric unit name in your I/O configuration which includes the EMC Open System device addresses; if such a name does not exist, you must use UNIT=3390 (or 3380 depending on how the devices were defined).

```
//DUMP        EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB      DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD   SYSOUT=*
//FDRSUMM      DD   SYSOUT=*
//SYSUDUMP     DD    SYSOUT=*
//DISK3        DD    UNIT=OPENSYS,VOL=SER=OPEN#1,DISP=OLD
//DISK5        DD    UNIT=OPENSYS,VOL=SER=OPEN#2,DISP=OLD
//DISK8        DD    UNIT=OPENSYS,VOL=SER=OPEN#3,DISP=OLD
//TAPE3        DD    DSN=BACKUP.OPEN1,DISP=(,CATLG),UNIT=3590,
//            DCB=TRTCH=COMP
//TAPE5        DD    DSN=BACKUP.OPEN2,DISP=(,CATLG),VOL=REF=*.TAPE3,
//           DCB=TRTCH=COMP,LABEL=2,DISP=(,CATLG)
//TAPE8        DD    DSN=BACKUP.OPEN3,DISP=(,CATLG),VOL=REF=*.TAPE5,
//           DCB=TRTCH=COMP,LABEL=3,DISP=(,CTLG)
//SYSIN        DD    *
  DUMP       TYPE=FULL
```

**DUMP MULTIPLE VOLUMES CONCURRENTLY**

Dump three Open System disk volumes concurrently to 3 different tape drives. The backup data sets are GDGs. MOUNT statements are used to specify which volumes to backup and to which TAPEx DD they are written. A job like this might be used to backup volumes which must be backed up at the same time, such as members of a AIX/6000 volume group. PRINT=STATUS requests that FDRSOS identify the type and contents of each Open System volume dumped, to document that the correct volumes were backed up.

```
//DUMP          EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB        DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT       DD   SYSOUT=*
//SYSPRINA       DD   SYSOUT=*
//SYSPRINB       DD   SYSOUT=*
//SYSPRINC       DD   SYSOUT=*
//FDRSUMM        DD   SYSOUT=*
//SYSUDUMP       DD   SYSOUT=*
//TAPEA          DD   DSN=BACK.UP.OPEN1(+1),DISP=(,CATLG),
//              UNIT=TAPE,VOL=(,,,99)
//TAPEB          DD   DSN=BACKUP.OPEN2(+1),DISP=(,CATLG),
//              UNIT=TAPE,VOL=(,,,99)
//TAPEC          DD   DSN=BACKUP.OPEN3(+1),DISP=(,CATLG),
//              UNIT=TAPE,VOL=(,,,99)
//SYSIN          DD   *
    DUMP        TYPE=FULL,MAXTASKS=3,PRINT=STATUS
    MOUNT       VOL=EMC120,TAPEDD=A
    MOUNT       VOL=EMC123,TAPEDD=B
    MOUNT       VOL=EMC128,TAPEDD=C
```

**Note:** FDRSOS software compression (invoked by the COMPRESS= operand on DUMP statements) is recommended only when the output backup data set is on disk or is on tapes connected by slower parallel (bus/tag) channels, such as 3480 tape drives. For disk output it reduces the backup data set size, while on parallel-attached tapes it reduces channel utilization, allowing for more concurrent backups. For tapes connected via ESCON (serial) channels, the channel is fast enough that channel utilization is not a problem, so hardware IDRC compression is recommended.

210.21 CONTINUED . . .

## DUMP SELECTED DATA BLOCKS

Selected ranges of data blocks are dumped from an Open System volume to a backup data set on disk. FDRSOS compression is used to reduce the size of the backup (this is recommended when outputting to disk backup data sets). Since is it is difficult to tell what data resides in specific blocks on an Open System volume, this procedure will not be used for normal operations.

**Note:** FDRSOS software compression is not supported in this release (V5.3/05). It will be supported in a future release. COMPRESS=will be ignored.

```
//DUMP       EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB      DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD    SYSOUT=*
//FDRSUMM      DD    SYSOUT=*
//SYSUDUMP     DD    SYSOUT=*
//TAPE1        DD    DSN=TECH.BACKUP.OPEN1,UNIT=SYSDA,
//           SPACE=(CYL,(20,5),RLSE),DISP=(NEW,CATLG)
//DISK1        DD    UNIT=3390,VOL=SER=AIX123,DISP=OLD
//SYSIN        DD    *
   DUMP      TYPE=PARTIAL,COMPRESS=ALL
   SELECT    FROMBLK=0,TOBLK=499
   SELECT    FROMBLK=1050,TOBLK=1070
```

## 210.22 FDRSOS RESTORE EXAMPLES

**FULL VOLUME RESTORE**

Restore an Open System volume. By default, the operator will be prompted for permission before beginning the restore.

```
//RESTORE    EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//FDRSUMM     DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//TAPE1       DD     DSN=BACKUP.VOP3132.C1,DISP=OLD
//DISK1       DD     UNIT=3390,VOL=SER=OP3132,DISP=OLD
//SYSIN       DD     *
   RESTORE   TYPE=FULL
```

**SERIAL VOLUME RESTORE**

Restore several Open System volumes serially. CONFMESS=NO indicates that the operator will **not** be prompted for permission before beginning the restores. Each of the indicated backup data sets will be dynamically allocated and restored to the indicated Open System volume. This might be used to restore volumes which must be processed together, such as members of an AIX/6000 volume group. Because BACKUPDSN= is used to specify the backups, the restores will be done serially, one at a time. PRINT=(STATUS,DIR) requests that FDRSOS identify the type and contents of each Open System volume both before and after the restore; for some volume types the contents of the root directory is also displayed. This will help document that the proper backup has been restored.

```
//RESTORE    EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//FDRSUMM     DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//SYSIN       DD     *
   RESTORE   TYPE=FULL,CONFMESS=NO,PRINT=(STATUS,DIR)
   MOUNT     VOL=EMC120,BACKUPDSN=BACKUP.OPEN1(0)
   MOUNT     VOL=EMC123,BACKUPDSN=BACKUP.OPEN2(0)
   MOUNT     VOL=EMC128,BACKUPDSN=BACKUP.OPEN3(0)
```

**CON-CURRENT VOLUME RESTORE**

Restore several Open System volumes concurently. By default, the operator **will** be prompted for permission before beginning the restores. Each TAPEx backup will be restored to the corresponding DISKx volume. Messages from each restore are printed on the SYSPRINx DD. This might be used to restore volumes which must be processed together, such as members of an AIX/6000 volume group.

```
//RESTORE     EXEC   PGM=FDRSOS,REGION=0M
//STEPLIB      DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD    SYSOUT=*
//FDRSUMM      DD    SYSOUT=*
//SYSUDUMP     DD    SYSOUT=*
//TAPE1        DD    DSN=BACKUP.VAIX01A,DISP=OLD
//DISK1        DD    UNIT=3390,VOL=SER=AIX01A,DISP=OLD
//SYSPRIN1     DD    SYSOUT=*
//TAPE2        DD    DSN=BACKUP.VAIX01B,DISP=OLD
//DISK2        DD    UNIT=3390,VOL=SER=AIX01B,DISP=OLD
//SYSPRIN2     DD    SYSOUT=*
//SYSIN        DD    *
   RESTORE    TYPE=FULL,MAXTASKS=2
```

## 210.22 CONTINUED. . .

### SELECTED BLOCK RESTORE

Restore selected data blocks to an Open System volume. The backup must contain the indicated data blocks; it might be a full backup of the volume, or a selected block backup that includes those blocks. The blocks will always be restored to their original locations on the disk; there is no facility for restoring to a different location. The backup data set is dynamically allocated.

```
//RESTORE    EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//FDRSUMM     DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//SYSIN       DD     *
   RESTORE   TYPE=PARTIAL
   MOUNT     VOL=E#01E3,BACKUPDSN=PROD.BACKUP.OPEN3
   SELECT    FROMBLK=0,TOBLK=5
```

### ALTERNATE VOLUME RESTORE

Restore several Open System volumes concurrently to alternate (spare) volumes. This technique can be used for recovery of individual files; see Section 220.14 for more details. The operator will be prompted for permission before beginning the restores. Each TAPEx backup will be restored to the corresponding DISKx volume, but the FDRSOS Open System volume serial of the output volumes will be preserved, and processing will be performed to preserve the physical volume IDs of the volumes and make the volumes usable on the Open System. Messages from each restore are printed on the SYSPRINx DD. PRINT=STATUS requests that FDRSOS identify the type and contents of each Open System volume restored, to document that the correct backups were restored.

```
//RESTORE     EXEC   PGM=FDRSOS,REGION=0M
//STEPLIB      DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD    SYSOUT=*
//FDRSUMM      DD    SYSOUT=*
//SYSUDUMP     DD    SYSOUT=*
//TAPE1        DD    DSN=BACKUP.VAIX01A,DISP=OLD
//DISK1        DD    UNIT=3390,VOL=SER=AIX27A,DISP=OLD
//SYSPRIN1     DD    SYSOUT=*
//TAPE2        DD    DSN=BACKUP.VAIX01B,DISP=OLD
//DISK2        DD    UNIT=3390,VOL=SER=AIX27B,DISP=OLD
//SYSPRIN2     DD    SYSOUT=*
//SYSIN        DD    *
   RESTORE     TYPE=FULL,MAXTASKS=2,CONFMESS=YES,
            CPYVOLID=NO,VOLRESET=NO,PRINT=STATUS
```

## 210.23 FDRSOS PRINT EXAMPLES

### PRINT SELECTED BLOCKS

Print selected blocks from an Open System Volume.

```
//PRINT      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//TAPE1       DD     SYSOUT=*
//SYSIN       DD     *
   PRINT     TYPE=PARTIAL
   MOUNT     VOL=EMC123
   SELECT    FROMBLK=0,TOBLK=49
   SELECT    FROMBLK=500,TOBLK=510
```

### PRINT NONZERO BLOCKS

Print selected blocks from an Open System Volume, but only print those blocks which are not all binary zeros. Innovation may ask you to run a PRINT job similar to this to help us understand new or unusual disk formats.

```
//PRINT      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//TAPE1       DD     SYSOUT=*
//SYSIN       DD     *
  PRINT      TYPE=PARTIAL
  MOUNT      VOL=SUN01X
  SELECT     FROMBLK=0,TOBLK=300
//$TRKPRIN    DD   *
  DEFAULT  NOZEROBLOCKS
```

### SCAN FOR ASCII DATA

An entire Open System volume will be scanned for a specified ASCII string. Only those data sectors containing the string will be printed.

```
//PRINT      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//DISK1       DD     UNIT=3390,VOL=SER=EMC123,DISP=OLD
//TAPE1       DD     SYSOUT=*
//SYSIN       DD     *
   PRINT     TYPE=FULL
//$TRKPRIN    DD*
   SCAN      ARGUMENT='test data',CASE=LOWER
```

## 210.23 CONTINUED. . .

### SCAN FOR HEX DATA

Selected data blocks on an Open System volume will be scanned for a specified hexadecimal string. Only those data sectors containing the string will be printed. No data checksums will be printed.

```
//PRINT      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//TAPE1       DD     SYSOUT=*
//SYSIN       DD     *
   PRINT     TYPE=PARTIAL
   MOUNT     VOL=EMC123
   SELECT    FROMBLK=0,TOBLK=999
//$TRKPRIN    DD     *
   DEFAULT   NOCHECKSUM
   SCAN      ARGUMENT=X'00134790'
```

## 210.24 FDRSOS ERASE EXAMPLES

**WARNING: the ERASE function will overwrite all data and preformatting on an Open System volume, and must be used with great care. After an ERASE, the volume will not be usable again until it is reformatted by utilities on the Open System involved. ERASE is most likely to be useful after a disaster test or when replacing an EMC Symmetrix containing Open System Data, to insure that all corporate data is removed. Although the CONFERASE=NO operand can be used to suppress the prompt for operator permission, Innovation recommends that you do not use it except at a disaster recovery site (and even then with care).**

**You should run a VARYON function with PRINT=STATUS before the ERASE to confirm that the correct volume has been selected for erasure (see example in Section 210.20).**

### ERASE A VOLUME

The volume specified by the MOUNT statement will be erased, entirely overwritten with binary zeros. The operator will be prompted for permission before erasing the volume. PRINT=STATUS requests that FDRSOS identify the type and contents of each Open System volume before it is erased, to document the data that was erased (after the erase it will identify it as "UNKNOWN VOLUME TYPE" to confirm that the erasure took place.

```
//ERASE      EXEC    PGM=FDRSOS,REGION=0M
//STEPLIB     DD     DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD     SYSOUT=*
//FDRSUMM     DD     SYSOUT=*
//SYSUDUMP    DD     SYSOUT=*
//SYSIN       DD     *
  ERASE      TYPE=FULL,PRINT=STATUS
  MOUNT      VOL=EMC123
```

### ERASE SEVERAL VOLUMES CONCURRENTLY

The volumes specified by the DISKx DD statements will be erased, entirely overwritten with binary zeros. The operator will be prompted for permission before erasing the volumes.

```
//ERASE       EXEC   PGM=FDRSOS,REGION=0M
//STEPLIB      DD    DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD    SYSOUT=*
//FDRSUMM      DD    SYSOUT=*
//SYSUDUMP     DD    SYSOUT=*
//DISK1        DD    UNIT=3390,VOL=SER=OPEN#1,DISP=OLD
//TAPE1        DD    DUMMY
//SYSPRIN1     DD    SYSOUT=*
//DISK2        DD    UNIT=3390,VOL=SER=OPEN#2,DISP=OLD
//TAPE2        DD    DUMMY
//SYSPRIN2     DD    SYSOUT=*
//DISK3        DD    UNIT=3390,VOL=SER=OPEN#3,DISP=OLD
//TAPE3        DD    DUMMY
//SYSPRIN3     DD    SYSOUT=*
//SYSIN        DD    *
  ERASE       TYPE=FULL,MAXTASKS=3
```

## 210.25 FDRSOS TIMEFINDER™ BCV EXAMPLES

Timefinder™ is an optional feature of EMC Symmetrix subsystems which allows you to create instant "frozen" copies of disk volumes, so that the frozen copy can be backed up even though updates are occuring to the primary volume. This facility is described in detail in Section 200.05.

### ESTABLISH BCV MIRRORS

The two volumes identified by the MOUNT statements (AIX102 and AIX103) are assigned BCVs at S/390 addresses 01F3 and 01F4 respectively. If the BCV address specified is not an Open System BCV, or if it has a different size than the primary volume, the ESTABLISH will fail. The Symmetrix hardware will completely synchronize each volume with its BCV by copying all data from the primary volume to the BCV, as a background task. BCV=WAIT causes FDRSOS to initiate all the ESTABLISH operations and then wait until the synchronization is compete for all volumes. If you don't need to know when synchronization is complete, specify BCV=NOWAIT (or omit BCV= since NOWAIT is the default).

```
//ESTAB     EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD       DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD       SYSOUT=*
//SYSIN      DD       *
  ESTABLISH TYPE=SOS,BCV=WAIT
  MOUNT   VOL=AIX102,BCVUNIT=01F3
  MOUNT   VOL=AIX103,BCVUNIT=01F4
```

### SPLIT BCV MIRRORS

All Open System volumes whose volume serial starts with "AIX10" will have their BCVs detached. Once a BCV is split, it becomes a "frozen" copy of the primary volume which can be backed up even though further updates are taking place to the primary volume. You may need to quiesce updates to the primary volumes until the SPLIT is complete (see Section 220.11); BCV=WAIT will cause FDRSOS to wait until the SPLIT is complete on all selected volumes, so when the SPLIT step ends you know that you can re-enable updates.

```
//SPLIT     EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD       DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD       SYSOUT=*
//SYSIN      DD       *
  SPLIT TYPE=SOS,BCV=WAIT
  MOUNT   VOL=AIX10*
```

## 210.25 CONTINUED. . .

### BACKUP AND RE-ESTABLISH BCV MIRRORS

This is essentially a combination of the following two examples, in one step. BCV=(USE,RET) causes FDRSOS to backup the BCVs most recently assigned to the primary volumes selected; at the end of each backup an automatic RE-ESTABLISH is done so no separate RE-ESTABLISH step is required. A SPLIT must be done before the backup, so this backup step will probably immediately follow the SPLIT step in the previous example.

```
//DUMP      EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD       DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD       SYSOUT=*
//SYSPRIN1   DD       SYSOUT=*
//SYSPRIN2   DD       SYSOUT=*
//FDRSUMM    DD       SYSOUT=*
//SYSUDUMP   DD       SYSOUT=*
//TAPE1      DD       DSN=BACKUP.AIX102(+1),UNIT=TAPE,
//            VOL=(,,,99),DISP=(,CATLG)
//TAPE2      DD    DSN=BACKUP.AIX103(+1),UNIT=TAPE,
//            VOL=(,,,99),DISP=(,CATLG)
//SYSIN      DD    *
  DUMP      TYPE=FULL,PRINT=STATUS,BCV=(USE,RET)
  MOUNT      VOL=AIX102,TAPEDD=1
  MOUNT     VOL=AIX103,TAPEDD=2
```

### BACKUP BCV MIRRORS

The BCV “frozen” copies of the Open System volumes specified by the DISKx DD statements are backed up. BCV=USE causes FDRSOS to backup the BCVs most recently assigned to the primary volumes selected. A SPLIT must be done before the backup, so the backup step will probably immediately follow the SPLIT step in the earlier example. The backup of a given disk will fail if there is no BCV which was previously ESTABLISHed and SPLIT from this primary volume.

```
//DUMP      EXEC  PGM=FDRSOS,REGION=0M
//STEPLIB    DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD   SYSOUT=*
//SYSPRIN1   DD   SYSOUT=*
//SYSPRIN2   DD   SYSOUT=*
//FDRSUMM    DD   SYSOUT=*
//SYSUDUMP   DD   SYSOUT=*
//TAPE1      DD   DSN=BACKUP.AIX102(+1),UNIT=TAPE,
//       VOL=(,,,99),DISP=(,CATLG)
//DISK1      DD   UNIT=OPENSYS,VOL=SER=AIX102,DISP=OLD
//TAPE2      DD   DSN=BACKUP.AIX103(+1),DISP=(,CATLG),UNIT=TAPE,
//      VOL=(,,,99),DISP=(,CATLG)
//DISK2      DD   UNIT=OPENSYS,VOL=SER=AIX103,DISP=OLD
//SYSIN      DD   *
  DUMP    TYPE=FULL,MAXTASKS=2,PRINT=STATUS,BCV=USE
```

## 210.25 CONTINUED. . .

### RE-ESTABLISH BCV MIRRORS

The Open System volumes whose volume serial starts with “AIX10” are resynchronized with their BCVs. FDRSOS will determine the address of the BCV most recently associated with each primary volume; RE-ESTABLISH will fail if there is no BCV which was previously ESTABLISHed and later SPLIT from this primary volume. The Symmetrix hardware will bring the BCV and primary volumes back in synchronization by copying to the BCV all primary volume data that was updated since the last SPLIT, as a background task. It will not wait for synchronization to be completed; if you need to know when it is complete, add BCV=WAIT to the RE-ESTABLISH statement.

```
//REESTAB   EXEC      PGM=FDRSOS,REGION=0M
//STEPLIB    DD       DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT   DD       SYSOUT=*
//SYSUDUMP   DD       SYSOUT=*
//SYSIN      DD       *
   RE-ESTABLISH TYPE=SOS
  MOUNT   VOL=AIX10*
```

### RESTORE BCV MIRROR

The Open System volume “TST012" is restored from the current contents of its BCV, which must currently be SPLIT from that volume. FDRSOS will determine the address of the BCV most recently associated with the primary volume; RE-ESTABLISH will fail if there is no BCV which was previously ESTABLISHed and later SPLIT from this primary volume. The Symmetrix hardware will bring the BCV and primary volumes back in synchronization by copying to the primary volume all BCV data that was updated on the primary since the last SPLIT, as a background task. This restores the primary volume to the exact contents it had at the time of the last SPLIT. However, if tracks were directly updated on the BCV while it was split, they will also be copied to the primary.

```
//REESTAB     EXEC PGM=FDRSOS,REGION=0M
//STEPLIB      DD  DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT     DD  SYSOUT=*
//SYSUDUMP     DD  SYSOUT=*
//SYSIN        DD  *
   RE-ESTABLISH TYPE=SOS,BCV=WAIT,RESTOREFROMBCV
   MOUNT   VOL=TST012
```

## 210.25 CONTINUED. . .

### FDR/UPSTREAM WITH BCV

Customers who have Innovation's FDR/UPSTREAM LAN-to-MVS backup software as well as FDRSOS can use the ability of FDR/UPSTREAM to execute programs and scripts on many Open Systems to coordinate the backups of the BCVs. More detail on this example is found in Section 200.05. USTBATCH is a FDR/UPSTREAM utility which executes a program or script on a specified Open System; consult FDR/UPSTREAM manuals for more information on this facility.

```
//QUIESCE    EXEC     PGM=USTBATCH
//*  EXECUTE A SCRIPT ON THE OPEN SYSTEM TO QUIESCE UPDATES
//*  ON THE VOLUMES TO BE DUMPED AND WAIT FOR QUIESCE TO COMPLETE
//STEPLIB     DD      DISP=SHR,DSN=upstream.loadlib
//USTLOG      DD      SYSOUT=*
//USTPARM     DD      *
CONV=WAIT                 wait for updates to be quiesced
TARGNAME=SERVER1
ACTION 5                  run a job
SPECNUMBER 1
FILES C:\DB\CLOSEALL.BAT execute this script
//SPLIT      EXEC  PGM=FDRSOS,COND=(0,NE)
//* SPLIT THE BCV AND WAIT FOR THE SPLIT TO COMPLETE
//STEPLIB     DD      DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD      SYSOUT=*
//SYSIN       DD      *
   SPLIT TYPE=SOS,BCV=WAIT
    MOUNT  VOL=AIX10*
//REENABLE  EXEC  PGM=USTBATCH
//*  EXECUTE A SCRIPT ON THE OPEN SYSTEM TO REENABLE UPDATES
//*  ON THE VOLUMES TO BE DUMPED.
//STEPLIB     DD      DISP=SHR,DSN=upstream.loadlib
//USTLOG      DD      SYSOUT=*
//USTPARM     DD      *
CONV=KEEP                   wait for request to be accepted
TARGNAME=SERVER1
ACTION 5                    run a job
SPECNUMBER 1
FILES C:\DB\OPENALL.BAT     execute this script
//DUMP       EXEC  PGM=FDRSOS,REGION=0M,COND=(0,NE)
//*  BACKUP THE SPLIT BCV VOLUMES, RE-ESTABLISH WHEN DONE
//STEPLIB     DD      DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD      SYSOUT=*
//SYSPRIN1    DD      SYSOUT=*
//SYSPRIN2    DD      SYSOUT=*
//FDRSUMM     DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//TAPE1       DD      DSN=BACKUP.AIX102(+1),DISP=(,CATLG),UNIT=TAPE,
//       VOL=(,,,99)
//TAPE2       DD      DSN=BACKUP.AIX103(+1),DISP=(,CATLG),UNIT=TAPE,
//      VOL=(,,,99)
//SYSIN       DD   *
   DUMP       TYPE=FULL,MAXTASKS=2,BCV=(USE,RET)
   MOUNT      VOL=AIX102,TAPEDD=1
   MOUNT      VOL=AIX103,TAPEDD=2
```

## 220.01 FDRSOSTC BACKUP COPY UTILITY

**OVERVIEW** The FDRSOS tape copy utility (FDRSOSTC) has been specifically designed to copy FDRSOS-formatted backups on tape or disk. **As noted earlier, FDRSOS backups cannot be correctly copied by any non-Innovation utilities.**

FDRSOSTC may be used to copy any FDRSOS backup, from disk or tape, to disk or tape, optionally creating a second copy at the same time. Backups in FDRSOS compressed format can be copied, and FDRSOSTC can create compressed output from uncompressed input, and vice versa. Simple JCL and control statements are used to specify the input and output backup data sets.

## 220.02 FDRSOSTC JCL REQUIREMENTS

FDRSOSTC requires the following JCL to execute:

**EXEC STATEMENT**

Specifies the program name (FDRSOSTC), and region requirement (a region of 4M is adequate for all functions) and optional PARM= field which may contain the FDRSOS control statement (a COPY statement). For example,

```
//SOS EXEC PGM=FDRSOSTC,REGION=0M,PARM=' COPY COMPRESS=NONE'
```

**STEPLIB DD STATEMENT**

If required, specifies the load library in which FDRSOSTC resides. It must be an APF authorized library.

**SYSPRINT DD STATEMENT**

Specifies the output message data set. Normally a SYSOUT data set.

**SYSUDUMP DD STATEMENT**

Specifies the abend dump data set. Although not required, we strongly urge you to always include this DD statement, so that we can help you diagnose error conditions. Usually specifies a SYSOUT data set.

**TAPEIN DD STATEMENT**

Specifies the input tape or disk FDRSOS backup data set. At least DSN= and DISP= must be given; if the input data set is not cataloged then UNIT=, VOL=, and possibly LABEL= must also be given.

**TAPEOUT DD STATEMENT**

Specifies the primary output tape or disk to be created.

If more than 5 output tape volumes may be required, you must specify a volume count in the VOL= parameter, e.g., "VOL=(,,,99)". Innovation recommends always specifying a volume count when the output is on tape.

You may want to specify the RETPD= or EXPDT= JCL parameters to specify the expiration of the output file.

If the output is on disk, the DD statement can create the output file (with DISP=(NEW,KEEP) or (NEW,CATLG) and SPACE=) or can refer to an existing output file (DISP=OLD).

TAPEOUT may be DUMMY; TAPEIN will still be read. This is useful to verify the readability of the input files, since FDRSOSTC will validate the contents and format of the backup file (Note: if the input file is compressed, specify COMPRESS=NONE for full validation).

**TAPE2OUT DD STATEMENT**

(Optional) Specifies that a second output copy is to be created. All comments about 'TAPEOUT' above apply to 'TAPE2OUT'. If a TAPEOUT2 DD is present, it will be treated as an alias for TAPE2OUT. If the TAPE2OUT and TAPEOUT2 DDs are omitted, then the TAPEOUT file will be the only copy created.

**SYSIN DD STATEMENT**

Specifies the control statement data set. Usually an input stream or DD * data set. It can be omitted (or specified as DUMMY); if so, and no PARM= was specified on the EXEC statement, a COPY statement with no operands will be assumed.

## 220.03 FDRSOSTC CONTROL STATEMENTS

**COPY BUFNO=MAX | nn**

**,COMPRESS=ALL | COPY1 | COPY2 | NONE**

**COPY STATEMENT**

The COPY control statement is the only statement accepted by FDRSOSTC. All operands are optional. If the COPY statement itself is omitted, the defaults for all operands are assumed. So, unless you need to override one of the operands, you can execute most FDRSOSTC steps with no control statements at all (in which case the SYSIN DD statement can be omitted or specified as DUMMY).

**OPERANDS**

**BUFNO=** specifies how many buffers will be used while copying each Open System backup. Each buffer is 36K in length. The buffers acquired will be divided into 2 sets in order to overlap input and output I/O operations; each I/O will read or write one half of the buffers. Reducing the number of buffers will reduce the amount of below-the-line storage required while copying, but will also reduce the efficiency of the copy and increase the elapsed time.
**MAX** - specifies that 32 buffers will be acquired.
nn - the specified number of buffers is acquired. The value may be from 1 to 32 but it will be rounded up to the next higher even number. Values over 32 are treated as 32.

The default is MAX (32).

**COMPRESS=** **ALL** specifies the backup copy on TAPEOUT and TAPE2OUT (if present) are both to be compressed using FDRSOS software compression.
**COPY1** specifies that only the TAPEOUT backup is to be compressed. TAPE2OUT (if present) will be uncompressed.
**COPY2** specifies that only the TAPE2OUT backup (if present) is to be compressed. TAPEOUT will be uncompressed.
**NONE** specifies that neither backup will be compressed.
Compression of the output backups is independent of whether the input backup was compressed or not. The input data will be uncompressed or compressed as appropriate to create the output.

The default is ALL if the backup on TAPEIN was compressed, or NONE if it was not compressed.

## 220.04 FDRSOSTC EXAMPLES

**COPY ONE BACKUP**

One Open System backup on tape or disk is to be copied to another tape. The input data set is cataloged and the output data set will be cataloged. Since no SYSIN DD statement is present, the defaults for all operands of the COPY statement are assumed. By default, if the input backup was compressed by FDRSOS software (the COMPRESS= option on the DUMP statement) the output backup will also be compressed; otherwise, it will not be compressed.

```
//COPY          EXEC  PGM=FDRSOSTC,REGION=4M
//STEPLIB        DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT       DD   SYSOUT=*
//SYSUDUMP       DD   SYSOUT=*
//TAPEIN         DD   DSN=BACKUP.OPEN1,DISP=OLD
//TAPEOUT        DD   DSN=BACKUP.OPEN1.COPY2,DISP=(,CATLG),UNIT=CART
```

**COPY BACKUP CREATING 2 COPIES**

One Open System backup on tape is to be copied creating 2 identical outputs on tape. The input data set is cataloged and the output data sets will be cataloged. The outputs will not be compressed by FDRSOS software compression (COMPRESS=NONE) but will be compressed by IDRC hardware compression (TRTCH=COMP).

```
//COPY       EXEC PGM=FDRSOSTC,REGION=4M
//STEPLIB     DD  DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT    DD  SYSOUT=*
//SYSUDUMP    DD  SYSOUT=*
//TAPEIN      DD  DSN=BACKUP.OPEN1,UNIT=3490,VOL=SER=B01234,DISP=OLD
//TAPEOUT     DD  DSN=BACKUP.OPEN1.COPY2,DISP=(,CATLG),
//           UNIT=3490,DCB=TRTCH=COMP
//TAPE2OUT    DD  DSN=BACKUP.OPEN1.COPY3,DISP=(,CATLG),
//           UNIT=3490,DCB=TRTCH=COMP
//SYSIN       DD  *
    COPY  COMPRESS=NONE
```

CONTINUED . . .

## 220.04 CONTINUED . . .

### COPY MULTIPLE BACKUPS

Three Open System backups are to be copied to 3 consecutive files on the output tape. The input backup files are cataloged, but if they happen to be files on the same input tape, the RETAIN parameter insures that the input tape is not dismounted between steps. The COPY statement is specified via the PARM= on the EXEC statement, so no SYSIN DD statements are required.

```
//COPY1        EXEC  PGM=FDRSOSTC,REGION=4M,PARM=' COPY'
//STEPLIB       DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT      DD   SYSOUT=*
//SYSUDUMP      DD   SYSOUT=*
//TAPEIN        DD   DSN=BACKUP.OPEN1,DISP=OLD,VOL=(,RETAIN)
//TAPEOUT       DD   DSN=BACKUP.OPEN1.COPY2,DISP=(,CATLG),
//         UNIT=3490,VOL=(,RETAIN),LABEL=1
//COPY2        EXEC  PGM=FDRSOSTC,REGION=4M,PARM=' COPY'
//STEPLIB       DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT      DD   SYSOUT=*
//SYSUDUMP      DD   SYSOUT=*
//TAPEIN        DD   DSN=BACKUP.OPEN2,DISP=OLD,VOL=(,RETAIN)
//TAPEOUT       DD   DSN=BACKUP.OPEN2.COPY2,DISP=(,CATLG),
//          UNIT=3490,VOL=(,RETAIN,REF=*.COPY1.TAPEOUT),LABEL=2
//COPY3        EXEC  PGM=FDRSOSTC,REGION=4M,PARM=' COPY'
//STEPLIB       DD   DISP=SHR,DSN=fdrsos.loadlib
//SYSPRINT      DD   SYSOUT=*
//SYSUDUMP      DD   SYSOUT=*
//TAPEIN        DD   DSN=BACKUP.OPEN3,DISP=OLD,VOL=(,RETAIN)
//TAPEOUT       DD   DSN=BACKUP.OPEN3.COPY2,DISP=(,CATLG),
//          UNIT=3490,VOL=(,RETAIN,REF=*.COPY2.TAPEOUT),LABEL=3
```

## 220.05 FDRSOS OPEN SYSTEM UTILITIES

Several utilities are provided which run on the Open System to provide functions such as mounting and dismounting volumes. Documentation is found in the Appendix of this manual.

## 220.10 FDRSOS SPECIAL CONSIDERATIONS

The following sections detail some special considerations and procedures for the use of FDRSOS, including:

- Coordination of FDRSOS backups and restores with operations on the Open System.
- Coordination of FDRSOS backups with other backup products
- Use of FDRSOS backups with Innovation's FDR/UPSTREAM LAN-to-MVS backup product
- Recovering individual files from FDRSOS backups
- Diagnostic techniques, especially for verifying and correcting connectivity between the S/390 system and the EMC Symmetrix Open System devices.

## 220.11 FDRSOS COORDINATION

Since there is no communication between FDRSOS and the Open System (UNIX, Windows NT, etc.) actually using the data, some manual coordination may be necessary to ensure that you have valid backups and can successfully do restores.

### QUIESCING OPEN SYSTEM DATA

In general, you must ensure that the data on the Open System volumes has been completely written to disk (called "hardening") and is not being updated while the FDRSOS backups are running ("quiesced").

Data on Open System volumes may be in many formats, including various file system formats and database formats. Remember that an FDRSOS backup is a physical backup, reading the volume from the lowest addressed data block to the highest; it is not aware of file structures or database formats. Since the data belonging to a given file, database or directory may actually be written in several nonadjacent areas of the disk, it is possible that FDRSOS will backup one part of a file and backup a second part of it many minutes later. If the data is updated between those two points, a restore from that backup may create an unusable version of the file. Even worse, if directory information or database indexes are updated during the backup, a great deal of data may be unusable or inaccurate when restored.

Also, some systems may buffer updated data in memory or cache of some sort, not writing it to disk immediately. In this case a backup may not get the most recent version of the data.

In most cases, you must take whatever steps are required on the Open System to see that all buffered data is flushed out and written to disk (hardened), and that no updates are allowed to take place during the backup. The necessary steps will vary depending on the type of system or database involved.

If you are using EMC's Timefinder™ option to backup BCVs (Business Continuance Volumes) instead of the live primary volumes, you will need to quiesce updates and harden data only for the brief period required to split the BCV from its primary volume and create a "frozen" copy of the data to be backed up. Details on BCVs are in Section 200.05.

Some database systems are architected to allow updates to take place during backups, using an update journal file to keep track of updates. After a restore, you specify the time that the backup began and the database system will reapply all updates from the journal, even if some of them are already in place from the backup. **You must be sure that your database software supports this function (update from journals after a physical volume restore) before you allow updates to occur during the FDRSOS backup; be very careful, not all journaled databases will support this type of restore.** Consult with your database vendor and Innovation if you are not sure.

CONTINUED . . .

220.11 CONTINUED . . .

**BACKUP COORDINATION**

You may need to coordinate your FDRSOS backups to ensure that you get a consistent backup image of the Open System data.

For some types of Open System data, each EMC logical disk (which looks like a SCSI physical disk) is an independent volume, including its own directory and not containing any data which must be coordinated with data on any other disk volume. Such volumes can be backed up and restored independently.

But other types of Open System data is spread across multiple Open Systems volumes, which requires that all Open System volumes involved must be backed up at about the same point-in-time, and all must be restored if any are restored.

An example of this is IBM AIX/6000 volume groups. A volume group is a collection of physical SCSI disks (EMC logical disks in this case) on which logical volumes are defined. These logical volumes (which are treated like physical volumes by the end-user) may be spread across any or all of the physical volumes in the volume group, as a collection of smaller areas called partitions. It is necessary to backup and restore all of the volumes in a given volume group together; otherwise the data may be totally unusable when restored.

Another example is Windows NT "volume sets", where a Windows "drive letter" is actually defined across several physical SCSI disks (or EMC logical disks).

**RESTORE COORDINATION**

Since FDRSOS will restore all data blocks on the Open System volume, the volumes being restored must usually be dismounted (disconnected) from the Open System CPU. The procedures for doing this will vary depending on the type of system, but it is necessary that you do this or the system may not properly recognize that FDRSOS has restored the volume. For Novell Netware and Windows NT, see the appendix of this manual.

In extreme cases, it may be necessary to shut down the Open System during the restores and restart it when they are complete.

If you are restoring a volume group or volume set (see notes under "Backup Coordination" above), you may need to predefine the target volumes as a volume group on your Open System before you do the restore. If you are restoring back to the original volumes this is usually not a problem since they are already properly defined. But if you are restoring to alternate target volumes as described in Section 220.14, you may need to define the group to your Open System **before** you restore. This is known to be true for Windows NT systems and may be true for other systems.

Innovation intends to provide guidance on the restore procedures for each type of system supported.

CONTINUED . . .

220.11 CONTINUED . . .

## HARDWARE RESERVES

RESERVE is a disk hardware instruction which is used by one system to prevent other systems from accessing a given disk volume, even though they have a channel connection to the disk device. The lockout continues until the RESERVing system explicitly RELEASEs the device.

The EMC Symmetrix supports two sorts of RESERVEs for Open System disks:

- a S/390 RESERVE
- a SCSI RESERVE

Either kind of RESERVE prevents any other system, whether S/390 or SCSI, from accessing the disk. S/390 I/Os to a RESERVEd disk will simply wait, but the issuing job; may eventually time out and abend. SCSI I/Os to a RESERVEd disk may cause waits or may cause I/O errors.

To the best of Innovation's knowledge, only IBM's AIX/6000 (UNIX for the RS/6000) issues a SCSI RESERVE to mounted disks. Since the AIX RESERVE would prevent FDRSOS from backing up the volumes, EMC has provided a special I/O technique which will allow FDRSOS to access the Open System volumes even though AIX (or any other system) has a SCSI RESERVE on the volumes.

FDRSOS has a option to issue a S/390 RESERVE on Open System volumes during backups, restores, and prints (the RESERVE= operand). Since this RESERVE will cause SCSI operations on the volumes (even reads) to wait or possibly fail, Innovation does not recommend its use during backups unless you have no other way of inhibiting updates to the volumes during backups. For restores, the default is RESERVE=YES since you generally want to prevent access to the volume during a restore.

**Note** that if RESERVE=YES is used and the Open System volume is currently RESERVEd to a SCSI channel, FDRSOS will fail the operation with an error message; it will not attempt to wait. If an AIX/6000 system has a volume RESERVEd, you cannot use the FDRSOS RESERVE unless you unmount the AIX volume group from AIX (the ‘varyoffvg’ command).

## 220.12 FDRSOS AND OTHER BACKUPS

FDRSOS backups are primarily for hi-speed full-volume backup and recovery. Depending on the type of data on the Open System volumes being processed by FDRSOS, you may need to use other backup products to provide all of the recovery that you need.

**Innovation's FDR/UPSTREAM, a hi-speed LAN-to-MVS backup/recovery system, can provide the necessary additional recovery in many cases. FDR/UPSTREAM is being enhanced to record and coordinate FDRSOS backups with normal FDR/UPSTREAM backups to provide a complete automated backup solution. See Section 220.13 for details on FDRSOS usage with FDR/UPSTREAM.**

### DATABASES

If the data on the Open System volumes is large database files, you will need to review the recovery options that are provided for the database system you are using.

Many database systems provide logging of all updates, creating a separate log file in which every change to the database is recorded. In this case, recovery of the database is relatively simple: you must restore the FDRSOS backup of every volume involved in the database, then instruct the database system to reapply all updates that were recorded **after** the FDRSOS backup was taken. However, you will need to provide for separate backups of the log files, especially if you intend to use them at a disaster recovery site. One way to do so is to put the log files on a separate, small logical volume within the EMC Symmetrix system, and backup that log disk with FDRSOS on frequent intervals, while backing up the databases themselves once a day or even less.

Some database backup products allow you to backup only the changed records within a database. This can also be used with FDRSOS backups if the database recovery product allows you to restore only the records that were changed after a given point. You can run FDRSOS backups on some schedule (perhaps weekly) and do the database backup of changed records more frequently (perhaps daily). If recovery is required, restore the most recent FDRSOS backup and then restore the database records that changed after the point of the FDRSOS backup. Note that this is practical only if the update activity in the database is relatively low; a database where a large percentage of the data is changed daily should be backed up daily in its entirety by FDRSOS.

### FILE SYSTEMS

Open System volumes containing many smaller files may also need additional backups with a product that can backup only the files that change each day (such as FDR/UPSTREAM, see Section 220.13).

To recover such a volume you would first do a full-volume restore from the most recent FDRSOS backup, then instruct the other backup product to restore any files that changed after the time of the FDRSOS backup.

**Warning: because FDRSOS restores the entire physical volume, all files that existed on that volume at the time of the backup will be restored. If some of those files were deleted from the volume after the FDRSOS backup, they will reappear on the restored disk. In the worse case, this may not leave enough room on the volume for the restore of updated files from the other backup product.**

Such a scheme also allows for the recovery of individual files from the backups. Although recovery of individual files from FDRSOS backups may be possible (see Section 220.14), it is much less convenient than the service provided by a file backup product.

## 220.13 FDRSOS AND FDR/UPSTREAM

### WHAT IS FDR/UPSTREAM?

FDR/UPSTREAM is Innovation Data Processing's LAN-to-MVS file backup and recovery product. It uses SNA/APPC and/or TCP/IP communications to backup files from servers and workstations to disk or tape backup data sets on MVS.

FDR/UPSTREAM supports many of the systems which may be using Open System volumes on EMC Symmetrix, including Windows NT, Novell Netware, OS/2, AIX/6000, Sun Solaris and HP-UX. Other versions of UNIX will be supported in the future.

Please consult the FDR/UPSTREAM documentation for complete details on the procedures described below.

### FDRSOS USE WITH FDR/UPSTREAM

FDR/UPSTREAM V2.5 and above can be used with FDRSOS backups for Open System disks which contain file systems:

- The Open System volumes should be backed up with a FDR/UPSTREAM first-time full MERGE backup, one time. This will backup all files on the volumes to provide a baseline backup for future MERGE backups.
- FDR/UPSTREAM incremental backups to disk should be done on a regular basis, probably daily.
- On a regular basis, perhaps weekly, USTMIGRT should be run with the FORWARD option. This will move any incrementals on disk to tape, plus it will copy all previous incrementals to the same output tape.

Since FDR/UPSTREAM will always have the current backup of individual files on the volume, individual files can be restored at any time.

For full-volume recovery, you must first restore the most recent FDRSOS full-volume backup. FDR/UPSTREAM V2.5.4 and above contains a "Restore back to FDRSOS Full" option. When enabled by an option in FDR/UPSTREAM on the Open System, it records a file containing the date/time of the last FDR/UPSTREAM backup on the Symmetrix volume. This timestamp file is restored by FDRSOS (along with the rest of the data on the volume). When the "Restore back to FDRSOS Full" option is selected, the FDR/UPSTREAM restore recognizes that timestamp and restores only backups created after that time automatically.

As a convenience, the FDR/UPSTREAM-MVS ISPF panels support an option to quickly generate FDRSOS batch jobstreams. One or many volumes can be selected, and it provides flexible options for stacking backups on tape.

FDR/UPSTREAM V2.5.5 has been further enhanced to support FDRSOS:

- FDR/UPSTREAM can now record the backups taken under FDRSOS. The backups must still be done by batch jobs independently of FDR/UPSTREAM, but after the backups are complete, a FDR/UPSTREAM utility (USTREGEN) can be run to read each of the backups and create an entry in the FDR/UPSTREAM repository documenting the date, time, and disk volume serial of each backup.
- If a restore of an Open System volume is required, the best way to do it is by a FDRSOS full-volume restore, using a batch job as described in this manual. However, it may not be possible to do a FDRSOS restore, especially at a disaster site where the disks may not be connected to the MVS host, or the restore disk may not even be EMC Symmetrix disks. In these cases, FDR/UPSTREAM can do a "raw" restore from the FDRSOS backup, transmitting the data over the network where it is rewritten by FDR/UPSTREAM on the workstation. This provides the equivalent of a FDRSOS restore, but at much lower speed.

CONTINUED . . .

220.13 CONTINUED . . .

## REMOTE INITIATED FUNCTIONS

The USTBATCH function of FDR/UPSTREAM allows a batch job or TSO user on the MVS mainframe to initiate functions on the workstation, server or Open System including:

- FDR/UPSTREAM backups and restores
- programs or batch files

The latter allows you to run a MVS job which can execute the necessary procedures on the server or Open System to close databases, mount volumes, or whatever functions are required to prepare for FDRSOS or FDR/UPSTREAM operations. This can include the utility programs described in Section 220.05.

## LOCAL BACKUPS

Version 2.5.3 of FDR/UPSTREAM introduced local backups, allowing FDR/UPSTREAM running on the Open System to create a local copy of backup data on local Open System disks specified by the user. The backup data was still sent over the user's network to FDR/UPSTREAM on MVS in order to write it to MVS tape or disk storage. The local backups provided for faster restore when recently backed up files are restored.

FDR/UPSTREAM V2.5.6 introduces a new local backup concept, using the facilities of EMC Symmetrix disks and FDRSOS. When EMC Open System volumes are formatted as special FDRSOS local backup volumes, the data does **not** need to be sent over the network, so the backups will run much faster.

On the Open System, FDR/UPSTREAM will write the data only to the SCSI local backup volume in the EMC Symmetrix, limited only by SCSI channel and device speed. FDR/UPSTREAM on MVS will read the local backup data directly from the Open System volume, using the same techniques built into FDRSOS, and will make a copy of the backup data on MVS tape or disk as indicated by the associated FDR/UPSTREAM profile. The speed of this copy is limited only by the MVS channel and device speeds. The FDR/UPSTREAM network connection between MVS and the Open System is used only to send control records. This new feature makes it practical to combine periodic FDRSOS full-volume backups with FDR/UPSTREAM file backups for complete recoverability with maximum backup speed.

Before it can be used with FDR/UPSTREAM, an Open System volume must be initialized as a FDRSOS local backup volume using the LOCALBACKUP TYPE=INIT statement of FDRSOS (see Section 210.10). This formats the volume as a DOS partition and initializes the control information required by FDR/UPSTREAM. This control information includes the maximum number of different FDR/UPSTREAM profile names which can have backups on the volume, plus default values to be associated with each profile when it is first used (such as the number of backups to keep on the local backup volume). These defaults can be updated by a LOCALBACKUP TYPE=UPDATE statement and can also be changed by FDR/UPSTREAM on the Open System.

220.13 CONTINUED . . .

**LOCAL BACKUPS (continued)**

When a Open System volume is initialized as a local backup, it will normally use the entire volume for local backup storage. So, a 4GB disk will use all 4GB for local backups. However, if your expected use of local backup does not require the entire volume, you can limit the space used by FDR/UPSTREAM by specifying the SIZE= operand on the MOUNT statement (size in megabytes). In this case, the DOS partition created at the beginning of the Open System volume will use only the size specified. The additional space on the volume can be used for other purposes if the Open System supports DOS partition tables; this includes all versions of Windows and Novell Netware **but does not include UNIX systems**. It is possible to expand the space used on a disk volume for local backups but you must insure that the additional space is not already in use for some other purpose.

As backups are directed to the local backup volume, by specifications in FDR/UPSTREAM on the Open System, space is allocated to the backup file and the backup written to it. For each FDR/UPSTREAM profile name which writes backups to this volume, there will be control records on the volume documenting each profile name, its backups, and its options. If the DYNADDPROF=YES operand was specified or defaulted when initializing the disk, new profiles are automatically added to the disk and use the default options set by the LOCALBACKUP statement. Facilities are provided in FDR/UPSTREAM on the Open System to view, add, and modify profiles on the volume.

Each profile has a maximum number of backups to be kept on the FDRSOS local backup. When a new backup is created for a profile, older backups under that profile may be deleted so that the maximum is not exceeded. So, the local backups are self-maintaining. If the FDRSOS local backup volume is short on free space, the DELMIGRATED=YES option will cause backups to be deleted as long as they have been successfully copied to MVS tape or disk.

When a FDR/UPSTREAM restore is requested, the files required will be restored directly from the local backup if the backup file still exists on the local backup volume. If it has been deleted, FDR/UPSTREAM will restore from its backups on tape or disk, sending the data over the network.

**Note:** unlike FDRSOS backups, which can only backup data stored on Open System volumes within the EMC Symmetrix disk subsystem, the FDRSOS local backup volumes in the Symmetrix can be used to backup and restore data from other non-Symmetrix disk volumes attached to the Open System.

## 220.14 FILE RESTORES FROM FDRSOS BACKUPS

Although FDRSOS is designed for full-volume backup and recovery of Open System volumes, it is possible to recover individual files from FDRSOS backups in many cases. This procedure **must be tested** in your environment to insure that it works.

### ALTERNATE VOLUME RESTORE

To perform this file recovery, you must reserve "spare" logical Open System volumes in the EMC Symmetrix system on which you can restore the FDRSOS backups of the "live" volumes from which you wish to recover. In other words, you must be able to restore the backup to a logical volume which is not currently in use but which can be accessed by the Open System to which the data belongs.

Obviously the spare volumes must be equal in size to the volumes whose backups you are restoring. If the data involved is spread across several Open System volumes (e.g., a AIX/6000 Volume Group), then you must have spare volumes of the proper number and size to restore **all** of the Open System volumes involved.

Most Open Systems record a volume identification (sometimes called a PVID or signature) in various locations on an Open System volume; this is similar to a MVS volume serial, but it is usually assigned by the operating system, not by the user. The volume identification may be used by the Open System software to identify the volume when it is mounted (or during IPL). In most cases, the Open System will not allow 2 volumes with the same identification to be mounted. Note that some volume formats, such as "raw volumes" and some database formats, do not have volume identifications.

Since FDRSOS normally backs up and restores all data on the volume, including the identification values, this would mean that a normal restore to an alternate volume would make the alternate volume unusable since the original volume with the same identifier is still mounted. To address this, FDRSOS has special support for alternate volumes:

- first, you must format the target volumes (the alternate volumes) with the appropriate Open System utility. If the system supports various disk formats (different types of file systems), it must be formatted as the same type as the volumes whose backup you are about to restore. The volumes should be the same size as the original volumes. This will assign unique identifiers to the volumes.
- second, you must specify the VOLRESET=NO operand on the RESTORE statement for the alternate volume restore (see Section 210.03). With VOLRESET=NO, FDRSOS will attempt to identify the format of the output volumes, locate the volume identification fields on the volumes, and read them. Then, as the backups are being restored, the identifier value preserved will be inserted into the identifier fields of the appropriate data sectors. In other words, the target volumes will still have the same unique identifiers after the restore. If the IDs are not valid on the disk, FDRSOS will take the ID from the backup and make it unique by adding an increment. The Open System will see these volumes as having identifiers distinct from the original volumes whose backup you just restored, and will be able to access them. VOLRESET=NO will also modify some other identification fields, such as logical volume IDs and UNIX mount points, to make them

unique.

## 220.14 CONTINUED. . .

Currently, VOLRESET=NO only supports volumes with certain formats:

- volumes used by AIX/6000 (UNIX for IBM RS/6000 systems). VOLRESET=NO will change the PVID on the physical volumes and the VGID (volume group ID) associated with each logical volume. It will also attempt to identify the "mount point" name associated with each logical volume (if any) and modify it by adding "_SOS" to the end of the name, allowing it to be mounted even though the original volumes are still mounted.
- volumes used by HP/UX are processed similarly to AIX volumes. If the volumes contain a Veritas file system, there are no mount points; for a standard HP file system, mount points are modified as shown above.
- volumes used by Sun Solaris systems. The volume ID and label are preserved from the output disk.
- volumes used by Windows NT.
- volumes used by Novell Netware.

### MOUNT THE RESTORED VOLUMES

Once the Open System volumes are restored to the spare volumes, you must make those alternate volumes accessible to the Open System. The details of this will vary depending on the type of system. On some, they may simply appear as extra "drive letters" (e.g., the K: drive). On some, special procedures are required to "mount" the volumes. For systems where no native commands to do this function are provided, the Innovation utilities documented in the Appendix of this manual provide the proper functions for most Open Systems.

### COPY THE FILES

Once the restored volumes are mounted, the files on them can be accessed. The required files should be copied to the proper locations on the normal volumes, using normal copy utilities. If necessary, the original files should be overlaid.

### DISMOUNT THE VOLUMES

Once the files have been recovered, the spare volumes should be "dismounted", if such a procedure exists on the system, to insure that files on the restored volume are not accidentally accessed in place of the live files. For systems where no native commands to do this function are provided, the Innovation utilities documented in the Appendix of this manual provide the proper functions for most Open Systems.

### TESTING

**WARNING: Because of the many variables involved, Innovation cannot be sure that this procedure will work in your environment. You should test this process on your system before you rely on it for file restores.**

## 220.15 FDRSOS DIAGNOSTIC TECHNIQUES

*This section primarily concerns diagnosis of problems which sometimes occur during the initial installation and configuration of an EMC Symmetrix for use with FDRSOS. It discusses the different types of problems which can occur, documents some diagnostic tools, and suggests solutions to common problems.*

It assumes that you (and your EMC representative) believe that you have taken all the proper steps, outlined in Section 200.03, to configure the Symmetrix, attach it to the S/390 system, and update the S/390 I/O configuration.

### MVS CONSOLE COMMANDS

The first tool for checking the configuration and connection is the MVS DEVSERV command (abbreviated DS) on the console. It will indicate whether MVS can successfully connect to the Symmetrix control unit and to the Open System volumes in particular, without involving FDRSOS.

You can verify the connectivity from MVS to the Symmetrix with the DEVSERV PATHS command (abbreviated DS P). If your Open System volumes were at addresses 1E0-1EF, you could issue the command:

```
DS  P,1E0,16
```

which would produce a display similar to:

```
    IEE459I 16.49.32 DEVSERV PATHS 321
 UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
 01E0,3390  ,F,000,E#01E0,20=+ 21=<
 01E1,3390  ,F,000,E#01E1,20=+ 21=<
 01E2,3390  ,F,000,E#01E2,20=& 21=<
                  ...
************************ SYMBOL DEFINITIONS
F = OFFLINE
< = PHYSICALLY UNAVAILABLE
+ = PATH AVAILABLE
& = RESERVED TO ANOTHER PATH/GROUP
* = LOGICALLY OFF, PHYSICALLY ON
- = LOGICALLY OFF, PHYSICALLY OFF
```

It displays:

- the device address
- the device type (make sure it is 3380 or 3390)
- device status (make sure it is F for offline)
- the volume serial (which will be blank unless you have previously used the volume with FDRSOS)
- the channel paths (CHPIDs) defined for this device (up to 4) and the status of each. A good connection is indicated by a status of:

  + path available

  & path available but the device is currently reserved to another system, such as an AIX/6000 system

## 220.15 CONTINUED. . .

**MVS CONSOLE COMMANDS (continued)**

Review the display to be sure that every path to every device has a good connection (except for paths which have not yet been physically connected, of course). Connection problems might be indicated by:

* path available but marked offline for this device. This might be due to a VARY PATH console command; VARY PATH is not recommended for Open System volumes. This might also occur if connection errors occurred at IPL time or when the device was dynamically added to the configuration. The FDRSOS command VARYPATH may be able to reactivate these paths (See Section 210.07)

- path physically offline to the hardware. You may be able to vary it online with the MVS console command: CF CHP(xx),ONLINE. Functions on the hardware console may also be used to re-enable the CHPID.

< path online but device could not be contacted, control unit not responding. This is the most common connection error status. See "Connection Errors" below for hints on finding the problem.

D a "device error" occurred when validating this path. This may indicate a problem or configuration error in the Symmetrix.

Other error status codes are possible but less common. See the IBM documentation for message IEA459I if you get other status codes.

Another useful console command is the DISPLAY M command (abbreviated D M). It will display the connection status of individual devices or all devices on a path, but unlike DEVSERV, does not actually attempt to verify the connection.

To display individual devices, issue:

```
D M=DEV(1E0)    or    D M=DEV(1E0-1EF)
```

specifying the device address or address range. To display all devices on a channel path in a compact format, issue:

```
D M=CHP(20)
```

specifying the CHPID of the channel. The information displayed is similar to that displayed by the DEVSERV command.

## 220.15 CONTINUED. . .

**FDRSOS DIAGNOS-TICS**

Most of the time, when a device address or address range is specified on a FDRSOS MOUNT statement, e.g.,:

```
MOUNT UNIT=01E4
MOUNT UNIT=01E*
```

the FDRSOS UCB scan routine will ignore any UCBs which are not candidates for Open System volumes. The tests it applies are:

- is the device a disk?
- is the device offline?
- is the device enabled for I/O?
- does the device have at least one enabled path (CHPID)?

This is done in case you specify addresses or ranges which are not Open System volumes on EMC Symmetrix subsystems, so that FDRSOS will not generate useless I/Os and error messages. If it passes the above tests, FDRSOS will then attempt to do I/O to the device, using CCW chains that will work on EMC Open System devices but which will fail on all other devices (including CKD disks from EMC and all other vendors). In most cases, if this I/O fails then FDRSOS will bypass the device, assuming that it is not an Open System volume.

If some configuration error prevents the test I/O from completing successfully on a valid Open System volume, FDRSOS may not report the error, making it difficult to determine what happened. This is done so that you specify a range of device addresses FDRSOS does not generate a large number of error messages for devices which are not Open System volumes. If a given MOUNT statement references multiple devices (e.g., MOUNT UNIT=01E*), FDRSOS will only report results on devices that have been identified as Open System volumes. However, if all devices specified by one MOUNT statement fail to pass the tests, you will get a FDR316 message indicating that no devices matched your MOUNT.

To make it easier to diagnose such problems, there is one exception to the above. When processing a LABEL statement, which requires a MOUNT statement specifying a single unit address, FDRSOS bypasses the above tests and goes straight to the test I/O. If the test I/O fails, then you will get a message and a I/O trace showing the results, similar to this example:

```
FDR303  CARD IMAGE --  LABEL  TYPE=SOS
FDR303  CARD IMAGE --  MOUNT UNIT=014F
FDR212** FDRSOS ERROR UNIT=014F REASON=4 - NOT A RESPONDING SOS UNIT
FDR149  I/O TRACE SECT=X'00000000'      3.504 MSEC - 109,367.476 - DD=D#FDRSOS - UNIT=014F
FDR149  CCW/DATA ---- IOB C6008000 41009290 00008BD8 0E000000 00008BD0 5000921C 00000000 00000000
                  -- SENSE 80000000 CF000004 00000032 95000004 020085BD 03010F04 00004CE3 00000000
008BD0  AF008E08 6000000C E4008EA8 60000007 64008EB0 20000040
        1F000000 00000000 00000000 00000000 00000000 00000000
```

## 220.15 CONTINUED. . .

**FDRSOS DIAGNOSTICS (continued)**

In the "I/O trace", there are several fields of interest:

- the IOB documents the termination status of the test I/O. The first byte of the second word is the terminating status indicator; it is X'41' in this example, indicating an I/O error. Another possible value is X'6D' indicating no paths were available (CC=3 on the Start Subchannel).
- The third and fourth words of the IOB are the CSW (Channel Status Word) documenting the I/O termination. The first 2 bytes of the fourth word are the device and channel hardware status of the I/O (X'0E00' in our example).
- If the device status include the X'02' flag (Unit Check), a device I/O error occurred, and the SENSE line will be printed showing detailed status information returned by the device. In this example, the sense starts with X'80' (Command Reject) indicating that the control unit did not recognize our test CCW chain, which may mean that it is not properly configured as an Open System device.
- The lines below the IOB and SENSE are the actual CCW chain executed.

Your EMC representative may be able to help you interpret the I/O status indicators. However, you may always call Innovation Technical Support for assistance if you are unable to determine the cause.

## 220.15 CONTINUED. . .

### FDRSOS UCB DISPLAY

FDRSOS has another diagnostic tool built in, a UCB display. Since FDRSOS does a number of tests against the UCB (Unit Control Block) to determine if it is a valid Open System volume, some problems can be diagnosed by examining the fields in the UCB. In fact, if you are calling Innovation Technical Support because of connection problems, we would prefer that you get this UCB display and have it available when you call.

To get the UCB display, simply add the parameter PRINT=UCB onto your first FDRSOS control statement. This will cause FDRSOS to format the UCB of every device address you specify. The UCB is formatted in the joblog and system messages of the FDRSOS job. In ESA 5.1 and above (including OS/390) it will be similar to (some sections abbreviated to conserve space):

```
UCB PREFIX AT 01E49BD0
    -0008  LOCK..... 00000000  IOQ...... 00000000
UCB AT 01E49BD8
    +0000  JBNR..... 00        FL5...... 88        ID........FF
    +0003  STAT..... 04        CHAN..... 01FF      FL1.......00
    +0007  FLB...... 00        NXUCB.... 00000000  WGT.......04
    +000D  NAME..... 1FF       TBYT1.... 30        TBYT2.....30
    +0012  DVCLS.... 20        UNTYP.... 0F        FLC.......00
    +0015  EXTP..... E49BB1
    +0018  VTOC..... 00000000  VOLI.....           STAB......00
    +0023  DMCT..... 00        SQC...... 00        FL4.......00
    +0026  USER..... 0000
UCB COMMON EXTENSION AT 01E49BB0
                ...
UCB EXTENDED PREFIX AT 01E50080
    +0000  RSTEM.... 00        MIHKY.... 04        MIHTI.....00
    +0003  HOTIO.... 00        IOQF..... 00000000  IOQL......00000000
    +000C  SIDA..... 0001      SCHNO.... 0090      PMCW1.....2898
    +0012  MBI...... 00A0      LPM...... C0        RSV.......00
    +0016  LPUM..... 00        PIM...... C0        CHPID.....2021FFFF
    +001C          FFFFFFFF    LEVEL.... 01        IOSF1.....40
    +0022  IOTKY.... 00        MIHFG.... 00        LVMSK.....00000001
DCE AT 01E49B80
                ...
CMB AT 053FE400
                ...
```

On systems before ESA 5.1, the Extended Prefix does not exist, so those fields are in the prefix, as shown below:

```
UCB PREFIX AT 00FCECE0
   -0030  RSTEM.... 00        RSV...... 00        MIHTI.....00
   -002D  HOTIO.... 40        IOQF..... 00000000  IOQL......00000000
   -0024  SIDA..... 0001      SCHNO.... 0010      PMCW1.....2898
   -001E  MBI...... 0018      LPM...... F0        RSV.......00
   -001A  LPUM..... 10        PIM...... F0        CHPID.....01020607
   -0014          00000000    LEVEL.... 01        IOSF1.....00
   -000E  IOTKY.... 00        MIHFG.... 00        LVMSK.....00000001
   -0008  LOCK..... 00000000  IOQ...... 00FC4880
UCB AT 00FCED10
   +0000  JBNR..... 00        FL5...... 8A        ID........FF
   +0003  STAT..... 84        CHAN..... 014F      FL1.......40
   +0007  FLB...... 00        NXUCB.... 00FCEDB0  WGT.......04
   +000D  NAME..... 14F       TBYT1.... 30        TBYT2.....30
   +0012  DVCLS.... 20        UNTYP.... 0E        FLC.......04
                ...
```

## 220.15 CONTINUED. . .

**FDRSOS UCB DISPLAY (continued)**

Some things that FDRSOS checks:

- in field STAT, X'80' indicates it is online; this must not be on
- field DVCLS must be X'20' (disk device class)
- in field FLB, the X'40' or X'20' bits indicate no path to device; they must be off
- field LPM (logical path mask) must be non-zero otherwise there are no enabled paths (CHPIDs) to the device.

You can verify these fields in the UCB display. If the UCB fails one of the first 2 checks, check your I/O configuration (HCD or MVSCP). If it fails the last 2 tests, see "Connection Errors" below.

Normally the LPM and PIM (path installed mask) are the same. The field CHPID shows the CHPIDs assigned to this device.

## 220.15 CONTINUED. . .

### FBACHK

The FDRSOS program library contains a program FBACHK. This is not an Innovation-written program. It was written by EMC to help identify problems with Open System FBA disk volumes. It is distributed with FDRSOS since it is only used by FDRSOS installations. It validates that the SOS disk device can be accessed, verifies that the Symmetrix is correctly configured for SOS and tests each of the special facilities used by FDRSOS.

If you are having problems accessing Open System volumes with FDRSOS, you should execute this program and give the results to your EMC representative, who can deal with the EMC support center to resolve the problems it reports. You may also want to execute FBACHK immediately after installation of the SOS disks, even before running any FDRSOS jobs.

To execute FBACHK, use this JCL:

```
//FBACHK  EXEC PGM=FBACHK,PARM='uuuu,VERIFY'
//STEPLIB  DD  DISP=SHR,DSN=fdrsos.library
```

No SYSPRINT is required; all displays are done to the job log of the job. "uuuu" is the 4-digit device address of the Open System device to be tested.

FBACHK displays the microcode and patch levels of the control unit, verifies that the device is offline and that it responds to FBA commands. If it successfully passes all tests the job log will contain text similar to:

```
FBAU000I DEVICE 0727, UCBADDR 00F21CF0, UCBPRFX 01ECBB08
FBAU002I CNTLTYPE SYM3, MCLEVEL 5063, PATCH LEVEL 3408,SYMDEV# 0027
FBAU003I DEVICE CONFIGURED CORRECTLY
```

Other messages indicate a problem of some sort. If you are unable to identify and resolve the problem from the text of the error message, contact your EMC representative for assistance.

One common error message is:

```
FBAU012E CUU DOES NOT SPECIFY A SHARED FBA DEVICE
```

This indicates an internal configuration error in the Symmetrix. When a device is assigned both a S/390 and a SCSI address, this turns on a "shared" flag in the Symmetrix configuration which enables the special FDRSOS support. If it has only a S/390 address, this error is produced (it has nothing to do with "shared DASD" or any options in the S/390 I/O configuration). If you have not yet assigned a SCSI address to a volume but want to test FDRSOS on it, ask your EMC representative to manually turn on this flag for the device.

There is one more way that FBACHK can be used:

```
//FBACHK EXEC  PGM=FBACHK,PARM='uuuu,FIXUCB'
//STEPLIB  DD  DISP=SHR,DSN=fdrsos.library
```

This will reset various software indicators for the device "uuuu". It is similar to the FDRSOS VARYPATH command described in Section 210.07.

## 220.15 CONTINUED. . .

### FDRDEBUG: ONE MORE TOOL

FDRSOS includes a utility debugging program called FDRDEBUG; it is also distributed with FDR. FDRDEBUG has a variety of functions, some of which are useful with FDRSOS.

FDRDEBUG can be called from TSO or executed as a batch program. From TSO enter:

```
CALL 'fdrsos.library(FDRDEBUG)' 'FDRDEBUG,function'
```

In batch, execute:

```
//DEBUG   EXEC  PGM=FDRDEBUG,PARM='FDRDEBUG,function'
//STEPLIB  DD  DISP=SHR,DSN=fdrsos.library
```

no SYSPRINT is required; all displays are done to the job log of the job.

The useful functions include:

| | |
|---|---|
| LLA | display linklist libraries |
| APF | display APF authorized libraries |
| modulename | locate the named module |
| ?hexaddress | locate module at that address |
| *hexaddr,len | display storage at that address |
| -device | display UCB for the named device (e.g., -1E4 or -374E) |
| _device | monitor CMD for the named device |

For example:

```
CALL 'fdrsos.library(FDRDEBUG)' 'FDRDEBUG,APF'

//DEBUG  EXEC  PGM=FDRDEBUG,PARM='FDRDEBUG,-1E8'
//STEPLIB  DD  DISP=SHR,DSN=fdrsos.library
```

The last function (_device) is useful to monitor the performance of an Open System volume once it is operating normally. Every 5 seconds it displays fields from the CMB (Channel Measurement Block) allowing you to see how performance it changing. It terminates after 5 minutes or when you do ATTENTION under TSO. The display will be similar to:

```
UNIT VOLSER  SECS  CONN PEND DISC CUQ  - TOTC TOTP TOTD TOTQ
1EC  E#01EC  005    46%   1%  52%   0%    40%   0%  60%   0%
1EC  E#01EC  005    47%   1%  53%   0%    40%   0%  60%   0%
1EC  E#01EC  005    47%   0%  53%   0%    40%   0%  60%   0%
1EC  E#01EC  005    45%   0%  52%   0%    40%   0%  60%   0%
```

The Connect, Pending, Disconnect and CU Queue values are the percentage of active I/O time spent in each state in the preceding 5 second interval; the equivalent TOT fields are values since the last IPL. In this example of a real FDRSOS backup, disconnect is mostly time spent by the Symmetrix fetching disk data into the cache, and connect is time spent transferring data to the channel. Larger values for pending (waiting for I/O initiation) or CU queue (waiting for free control unit) might indicate that the channels to the Symmetrix are overloaded. Larger values for disconnect may indicate contention problems within the Symmetrix.

## 220.15 CONTINUED. . .

### CONNECTION ERRORS

These hints may help you resolve your connection problems. If they do not, please contact your EMC representative and Innovation Technical Support for assistance.

If **all** devices on the Symmetrix subsystem are failing (including any S/390 CKD disks in the same subsystem):

1. Make sure that the physical channel connections have been made properly (ESCON or parallel channels properly connected on each end).
2. Check that the proper channel paths (CHPIDs) have been specified in your I/O configuration.
3. If your Symmetrix is connected by an ESCON director, be sure that the director configuration is correct.
4. Ask your EMC representative to verify that the Symmetrix configuration is correct.

If CKD disks in the same Symmetrix subsystem are working but all of the FBA Open System volumes are failing:

5. Check your I/O configuration (HCD or MVSCP) to be sure that they are properly defined as 3380 or 3390 disks and are marked offline.
6. Ask your EMC representative to be sure that the FBA disks in the Symmetrix have been assigned both a S/390 address and a SCSI address.

If the Open System volumes are assigned only a S/390 address, FDRSOS support in the Symmetrix microcode will not be enabled and they will respond only to standard IBM FBA CCWs, not the special CCWs used by FDRSOS. This will result in a Command Reject as shown in the sample I/O trace earlier in this section. If they are assigned only a SCSI address, they will not respond to the S/390 at all. They must be assigned both a SCSI and S/390 address to be eligible for FDRSOS processing.

If the DEVSERV PATH command shown earlier in this section shows a status of < for a path, this means that the control unit (the Symmetrix channel adapter) responds to the device address, but indicates that no such device exists. This is similar to a short string of IBM disks where some device addresses do not exist. This is probably due to some error in the Symmetrix configuration.

If I/O errors are occurring on the Open System volumes for no obvious reason, you can try the FDRSOS VARYPATH command (See Section 210.07); it resets certain flags which may prevent I/O from completing successfully, and reenables all defined paths to the device.

As always, if you and EMC cannot determine the problem, please call Innovation Technical Support.

### I/O ERRORS

Because of the high reliability of the EMC Symmetrix subsystem, and additional recovery provided by the mirroring and RAID-S options of the subsystem, unrecoverable I/O errors will be rare. If they do occur, you will receive an I/O trace printout similar to that shown earlier in this section. The IOB, sense, and CCW printouts will identify the error that occurred. You can discuss these errors with your EMC representative, or call Innovation Technical Support.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

## 230.01 FDRSOS INSTALLATION

Section 230 contains the simple instructions for installing the FDRSOS product on your system.

**DISTRIBUTION TAPE FORMAT**

FDRSOS is distributed on a standard label (SL) magnetic tape volume. The volume serial and status of the distribution tape will be clearly marked on the external tape label. The volume serial of the tape will be:

FDR53P - a production (licensed) version of FDRSOS.

FDR53T - a trial (evaluation) version of FDRSOS. The trial programs will expire (stop functioning) on the expiration date indicated on the external label.

The files on the tape are:

File 2: DSN=LOAD
IEBCOPY-unloaded library of programs.

File 4: DSN=ICL
IEBCOPY-unloaded library of installation jobstreams and supplementary documentation.

File 10: DSN=JCL
IEBCOPY-unloaded library of all example JCL from this user manual.

Since the FDRSOS distribution tape shares a common format with other Innovation distribution tapes, many files on the tape are not used at this time. Any files not listed above will have a data set name of DUMMY and will be empty.

**INSTALLATION INSTRUCTIONS**

The following sections document installation of the FDRSOS libraries.

**WARNING: If you are licensed for other FDR components (such as FDR, FDRABR, COMPAKTOR, and others) you will receive separate distribution tapes for FDRSOS and the other products. You can install them into the same program library, as long as all the modules are at the same maintenance level (for example, V5.3 level 34), since some programs are shared between FDRSOS and other FDR components. If you must install different levels of FDRSOS and other FDR components, then FDRSOS must be installed into a separate program library which cannot be in the system linklist; a STEPLIB DD statement will be required for FDRSOS execution. If you are installing a trial tape for FDRSOS and/or FDR, you must keep them in separate program libraries.**

## 230.02 INSTALLING THE INSTALLATION CONTROL LIBRARY

File 4 of the installation tape contains IEBCOPY input that will create an Installation Control Library. This library supplies the user with JCL streams and supplementary documentation. The member named 'INDEX' contains a brief functional description of each member within the Installation Control Library.

**DISK SPACE REQUIREMENTS**

The Installation Control Library must be loaded to a partitioned data set on disk. You may load it to an existing data set (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the Installation Control Library:

| DATA SET | RECFM | LRECL | BLKSIZE | BLOCKS | PDS DIR BLOCKS |
|---|---|---|---|---|---|
| IDP.ICLFDR53 | FB | 80 | 3120 | 1500 | 45 |

This data set is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this data set, you should adjust the number of blocks accordingly.

**INSTALLATION INSTRUCTIONS**

Execute the following JCL to load the Installation Control library. The following changes must be made to reflect your environment:

- 'DSN=IDP.ICLFDR53'on the SYSUT2 DD statement should be changed to the name you wish to use for the Installation Control Library.
- 'VOL=SER=vvvvvv' on the SYSUT2 DD statement must specify a disk volume where the Installation Control Library will be allocated, or may be omitted if the data set will be SMS-managed.
- 'UNIT=TAPE' on the SYSUT1 statement must specify a tape drive capable of reading the installation tape that you have received.
- 'VOL=SER=FDR53T' on the SYSUT1 statement must be changed to 'VOL=SER=FDR53P' if you are loading from a production installation tape.

**JOB CONTROL FOR LOADING LIBRARY**

```
//ICLLOAD     EXEC  PGM=IEBCOPY,REGION=1024K
//SYSPRINT     DD   SYSOUT=*
//SYSUT2       DD   DSN=IDP.ICLFDR53,          <--USER-CHANGE
//             VOL=SER=vvvvvv,                 <--USER-CHANGE
//             UNIT=SYSDA,DISP=(,CATLG),       SEE NOTE 1
//             DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//             SPACE=(3120,(1500,100,45),,,ROUND)
//SYSUT1       DD   DSN=ICL,
//             UNIT=TAPE,                      <--USER-CHANGE
//             DISP=OLD,LABEL=(4,EXPDT=98000),
//             VOL=SER=FDR53T  CHANGE T TO P IF PRODUCTION TAPE
//SYSIN        DD   *
  COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))
```

**NOTE 1:To load the members to an existing library,**

```
//SYSUT2       DD   DISP=OLD,DSN=IDP.ICLFDR53 <--USER-CHANGE
```

CONTINUED . . .

## 230.02 CONTINUED . . .

**MEMBERS** The members in the Installation Control Library provide you with jobstreams for completing the installation, as well as supplementary documentation. It will also contain members for other members of the FDR product line which do not apply to FDRSOS.

## 230.03 INSTALLING THE PROGRAM LIBRARY

**INSTALLATION**

The FDRSOS program library must be loaded to a partitioned data set on disk. You may load it to an existing dataset (if it has sufficient space and proper DCB) or allocate and load a new one. The following table shows the allocation parameters for the program library:

**DISK SPACE REQUIREMENTS**

| DATA SET | RECFM | LRECL | BLKSIZE | BLOCKS | PDS DIR BLOCKS |
|---|---|---|---|---|---|
| IDP.MODFDR53 | U | n/a | 6144 | 240 | 35 |

This data set is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this data set, you should adjust the number of blocks accordingly.

**JOB CONTROL FOR LOADING LIBRARY**

JCL similar to that below is supplied on the Installation Control Library (see section 230.02) with a member name of 'SOSLOAD'. The JCL in the Installation Control Library reflects the type of product installation tape from which it was loaded.

```
//LOAD          EXEC  PGM=IEBCOPY,REGION=1024K
//SYSPRINT       DD   SYSOUT=*
//SYSUT2         DD   DSN=IDP.MODFDR53,              <--USER-CHANGE
//               VOL=SER=vvvvvv,                     <--USER-CHANGE
//               UNIT=SYSDA,DISP=(,CATLG),           SEE NOTE 1
//               DCB=(BLKSIZE=6144,RECFM=U),
//               SPACE=(6144,(240,20,35),,,ROUND)
//SYSUT1         DD   DSN=LOAD,
//               UNIT=TAPE,                          <--USER-CHANGE
//               DISP=OLD,LABEL=(2,EXPDT=98000),
//               VOL=SER=FDR53T  CHANGE T TO P IF PRODUCTION TAPE
//SYSIN          DD   *
  COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))
```

**NOTE 1:To load the members to an existing library,**

```
//SYSUT2         DD   DISP=OLD,DSN=IDP.MODFDR53      <--USER-CHANGE
```

1. ***** CAUTION CAUTION CAUTION *****
   **If you are licensed for other FDR components (such as FDR, FDRABR, COMPAKTOR, and others) you will receive separate distribution tapes for FDRSOS and the other products. You can install them into the same program library, as long as all the modules are at the same maintenance level (for example, V5.3 level 34), since some programs are shared between FDRSOS and other FDR components. If you must install different levels of FDRSOS and other FDR components, then FDRSOS must be installed into a separate program library which cannot be in the system linklist; a STEPLIB DD statement will be required for FDRSOS execution. If you are installing a trial tape for FDRSOS and/or FDR, you must keep them in separate program libraries.**
2. **If you are installing a production tape (which has no date protected modules), you must change the installation JCL to reflect this; change the serial from FDR53T to FDR53P. You may need to change UNIT=TAPE to a unit name capable of reading the distribution tape you received (reel or cartridge).**
3. **The FDRSOS program library *must* be an APF authorized library. A library is APF authorized if it is accessed via the LINKLIST or its name is in SYS1.PARMLIB member IEAAPFxx. Under MVS/XA, MVS/ESA and OS/390, if the LNKAUTH=APFTAB option is specified in IEASYSxx, then even a library accessed via the LINKLIST has to be listed in IEAAPFxx to be authorized. For testing purposes, it is possible to dynamically authorize a library.**

## 230.04 INSTALLING THE JCL LIBRARY

File 12 of the installation tape contains IEBCOPY input that will create a JCL Library. This library supplies the user with copies of all of the example JCL shown in this manual.

**DISK SPACE REQUIREMENTS**

The JCL Library must be loaded to a partitioned data set on disk. You may load it to an existing data set (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the Installation Control Library:

| DATA SET | RECFM | LRECL | BLKSIZE | BLOCKS | PDS DIR BLOCKS |
|---|---|---|---|---|---|
| IDP.JCLFDR53 | FB | 80 | 3120 | 500 | 100 |

This data set is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this data set, you should adjust the number of blocks accordingly.

**INSTALLATION INSTRUCTIONS**

Execute the following JCL to load the Installation Control library. The following changes must be made to reflect your environment:

- DSN=IDP.JCLFDR53' on the SYSUT2 DD statement should be changed to the name you wish to use for the JCL Library.
- 'VOL=SER=vvvvvv' on the SYSUT2 DD statement must specify a disk volume where the JCL Library will be allocated, or may be omitted if the data set will be SMS-managed.
- 'UNIT=TAPE' on the SYSUT1 statement must specify a tape drive capable of reading the installation tape that you have received (reel or cartridge).
- 'VOL=SER=FDR53T' on the SYSUT1 statement must be changed to 'VOL=SER=FDR53P' if you are loading from a production installation tape.

**JOB CONTROL FOR LOADING LIBRARY**

The JCL below is supplied on the Installation Control Library (see section 230.02) with a member name of 'SOSJCL'. The JCL in the Installation Control Library reflects the type of product installation tape from which it was loaded.

```
//JCLLOAD     EXEC PGM=IEBCOPY,REGION=1024K
//SYSPRINT     DD  SYSOUT=*
//SYSUT2       DD  DSN=IDP.JCLFDR53,                    <--USER-CHANGE
//              VOL=SER=vvvvvv,                         <--USER-CHANGE
//              UNIT=SYSDA,DISP=(,CATLG),               SEE NOTE 1
//              DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//              SPACE=(3120,(500,50,100),,,ROUND)
//SYSUT1       DD  DSN=JCL,
//              UNIT=TAPE,                              <--USER-CHANGE
//              DISP=OLD,LABEL=(10,EXPDT=98000),
//             VOL=SER=FDR53T  CHANGE T TO FDR53P IF PRODUCTION TAPE
//SYSIN        DD    *
  COPY OUTDD=SYSUT2,INDD=((SYSUT1,R))
```

**NOTE 1: To load the members to an existing library,**

```
//SYSUT2       DD  DISP=OLD,DSN=IDP.JCLFDR53           <--USER-CHANGE
```

## 230.10 FDRZAPOP GLOBAL OPTION CHANGE FACILITY

Some options which affect FDRSOS may be changed permanently. All such options are kept in a load module called FDROPT, in the FDRSOS program library. Some of these options affect the defaults for various operands on FDRSOS control statements and can be overridden at execution time; others may be changed only in FDROPT and cannot be overridden.

All options may be set with program FDRZAPOP, the Global Option Change facility.

Note that FDRZAPOP and FDROPT are also used with the FDR system. Many of the options that can be modified or printed by FDRZAPOP have no meaning for FDRSOS. Only those options documented in the following sections are valid for FDRSOS.

**BATCH EXECUTION**

To execute FDRZAPOP as a BATCH job use the following JCL:

**EXEC STATEMENT**

Must specify the program name of the Global Option Change Facility -- FDRZAPOP.

**STEPLIB or JOBLIB DD STATEMENT**

Must specify the load module library in which FDRSOS resides.

**SYSPRINT DD STATEMENT**

Specifies the output message data set. This is a required DD statement and usually is a SYSOUT data set.

**SYSLIB DD STATEMENT**

Must also specify the load module library in which FDRSOS resides. The user must have authority to update this library.

**SYSIN DD STATEMENT**

Specifies the control statement data set required for all functions. Usually an input stream or DD * data set.

**TSO EXECUTION**

FDRZAPOP can be executed under TSO. The program will prompt you for the commands. The 'END' command will save the modified options, if any, and terminate the program. The 'CANCEL' command will discard any modified options and terminate. The allocations required to execute FDRZAPOP in the TSO Foreground are as follows:

```
ALLOC      F(SYSLIB)     DA('fdrsos.library') SHR
ALLOC      F(SYSPRINT)   DA(*)
ALLOC      F(SYSIN)      DA(*)
CALL 'fdrsos.library(FDRZAPOP)'
```

## 230.11 FDRZAPOP STATEMENTS AND OPTIONS

**AVAILABLE FUNCTIONS**

FDRZAPOP accepts the following statements:

**HELP –** The HELP command will print a menu of the FDRZAPOP options and related documentation. Note that it will include **all** FDR options, not just FDRSOS options.

The HELP command format is:

**HELP** or **HELP SUB(command)** or **HELP ALL**

**PRINT –** The current values in the FDRSOS Global Options Table will be printed. Note that the report will contain **all** FDR options, not just FDRSOS options.

The PRINT command format is:

**PRINT**

**RESET –** The FDRSOS Global Options Table will be reset to the original values on the installation tape.

The RESET command format is:

**RESET**

**ZAP –** Modify options in the FDRSOS Global Options Table. This command enables the installation to enable or disable specified functions and set control statements defaults and other processing options for FDRSOS. The operands for ZAP are documented below.

The ZAP command format is:

```
ZAP   [  ENABLE=(option1,...,optionn) ]
      [ ,DISABLE=(option1,...,optionn) ]
      [ ,operand=value,...,operand=value ]
```

## 230.11 CONTINUED. . .

**ZAP OPTIONS**

These are the operands that can be specified for the ZAP command, to set various permanent options for FDRSOS. If specified as a subparameter of the ENABLE operand, the option is enabled; specifying them as a subparameter of DISABLE disables them.

**ALLCALL** if enabled, FDRSOS does RACF-compatible security checks for dump, restore and erase operations. For each Open System volume, it issues a security check with CLASS=DASDVOL and ENTITY=the Open System volume serial assigned by the FDRSOS LABEL function. READ authority will be required for backups, and ALTER authority for restore and erase. ALLCALL is disabled by default. To change it, enter ZAP ENABLE/DISABLE=ALLCALL

ALLCALL security checking is effective only if your installation has implemented the DASDVOL security class (which can also be used to control access to normal MVS volumes) and defined the Open System volume serials under that class. FDRSOS will permit all access to a volume if the DASDVOL class is not active or if the volume is not controlled under that class. Consult your security system documentation for guidance on defining volume rules that are compatible with this DASDVOL check.

Note that FDRSOS also supports a volume security exit in which you can program your own security checks for volume access. If you need this function, contact Innovation for more information.

**BUFNO** As a default, during a DUMP FDRSOS will acquire 16 32K data buffers (ZAP BUFNO=MAX). You can specify that a smaller number of buffers be used (ZAP BUFNO=nn) but this will impact FDRSOS performance. Since this can be overridden at execution time by the BUFNO= operand, Innovation does not recommend changing this option.

**LINECNT** You can override the maximum number of lines to be printed on any report page produced by the FDR system. The default is 58 lines per page. To change this default, specify ZAP LINECNT=nn, where nn may be a number from 10 to 99.

**MESSAGE ROUTING AND DESCRIPTOR CODES**

The routing and descriptor codes used for the operator console messages in the FDR system can be changed. The default value for the routing codes is (2,11), and for the descriptor code is 2. The routing and descriptor codes are documented in the IBM manual *Routing and Descriptor Codes* and are used by MVS to control which consoles the messages will appear on, and how they will be displayed. To change them, specify ZAP ROUTECODE=(nn,...,nn) and/or ZAP DESCRIPTCODE=(nn,...,nn).

## 230.20 FDRSOS MESSAGES

This section describes the various printer and console messages which may be output by FDRSOS system and the various ABEND codes with which it may terminate.

**FDRSOS MESSAGE FORMAT**

The general format of FDRSOS messages and WTORs is as follows:

```
FDRnnn        Message-text.
or
FDRnnn**      Message-text.
```

When ** appears the message is an error message. The step will ABEND, or will pass a return code of 8 or higher unless indicated otherwise in the specific message.

**RETURN CODES**

FDRSOS always passes a return code (step completion code) at the end of the step, unless it ABENDs. A return code of zero indicates that FDRSOS has performed all functions successfully. A return code of 4 indicates that the testing period for a trial version of the product has expired. Any other return code indicates that errors of some kind have occurred during this execution of FDR. Check the listing for the error messages.

**ABEND CODES**

FDRSOS ABEND codes range from a U0100 to a U0900. Most user ABENDs are preceded by an FDR message. Note that some FDRSOS ABENDs will result in the entire FDRSOS step abending (for errors such as invalid JCL and control statement errors), while some ABENDs only affect a subtask under FDRSOS (such as a dump or restore subtask). Subtask ABENDs will be documented by a FDR319 message but they will usually not cause the entire step to abend; other subtasks may continue to execute and the step will end with a non-zero return code to indicate that some unusual condition occured.

## 230.21 FDRSOS CONSOLE MESSAGES

The following may appear on an MVS operator's console during FDRSOS operations.

**FDRW01 CONFIRM REQUEST TO *function* RESTORE UNIT=*uuuu*,VOL=*vvvvvv*,JOB=*jjjjjjjj***

**Reason**: The job indicated in the message wishes to restore or erase (as indicated by "function") the Open System volume specified. This confirmation can be suppressed by specifying the "CONFMESS=NO" operand on the RESTORE statement or "CONFERASE=NO" operand on the ERASE statement.

**Action:** Reply YES—Operator approves the operation.
Reply NO—Operator does not approve the operation; the operation is terminated. The subtask will terminate with a U0801 ABEND, but other restore or erase requests in the same FDRSOS step may proceed.

**FDRW20 FDR BCV SYNCHRONIZATION COMPLETED SUCCESSFULLY | WITH ERRORS**

**Reason:** EMC Timefinder™ BCV volumes are being used for FDRSOS backups (See Section 200.05). For a ESTABLISH, RE-ESTABLISH or SPLIT statement, BCV=WAIT was specified. This message is output to the MVS console to document that the requested operation is complete. If SUCCESSFUL appears, all the volumes in the step were completed successfully. If it says WITH ERRORS, one or more volumes had errors and the FDRSOS output should be reviewed.

**Action:** If you have an automation tool which recognizes console messages, you may use this message to recognize that the requested operation is complete and invoke additional processing.

**FDRW89 FDR - TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN 'nnn' DAYS - PLEASE CONTACT INNOVATION**

**Reason**: This is a trial version of FDRSOS. The number of days the trial will remain active is displayed. The product will stop operating when the trial expires.

**Action:** When there are 10 or fewer days before the trial is due to expire this message will become non-deletable. Call INNOVATION for assistance.

## 230.22 FDRSOS PRINTER MESSAGES

These messages may appear in FDRSOS printed output.

**FDR001 FDR SAFEGUARD OPEN STORAGE -- FDRSOS VER *v.r/mmb* -- INNOVATION DATA PROCESSING DATE=*yy.ddd* PAGE *nnn***

**Reason:** This is the FDRSOS page heading, specifying the version level of FDRSOS. "v.r" indicates the version and release (e.g., 5.3), "mm" is a 2-digit number indicating the maintenance level and "b" will be "P" for a production version, "T" for a trial, and "B" for a beta test version.

**FDR002 *function* SUCCESSFULLY COMPLETED VOL=vvvvvv**

**Reason**: The Open System volume with volume serial vvvvvv was successfully processed by the function indicated.

**FDR006 function OPERATION CANCELLED BY OPERATOR**

**Reason**: An operator replied 'NO' to an FDRSOS request for approval to restore a disk (message FDRW01). A U0801 Abend follows.

**FDR007 STARTING|ENDING TIME OF *function* -- *hh.mm.ss* --UNIT=*address*, IN=*inputdd* ,OUTPUT=*outdd1 outdd2***

**Reason**: Documents the time that FDRSOS began or ended the function indicated. "address" is the S/390 device address of the Open System volume.
"inputdd" is the DD name of the DD name of the Open System volume for DUMP, PRINT, and ERASE; this may be a DISKx name or a DD name dynamically allocated by FDRSOS. For RESTORE, it is the TAPEx DD name of the backup being restored.
"outdd1" is the output DD name. If a duplicate backup was requested, it is shown by DD name "outdd2".

**FDR008 OPEN ERROR OR NO DD STATEMENT DD=*ddname* - *function* BYPASSED**

**Reason**: 1. A tape DD statement specified by "ddname" was missing. A TAPEDD=x operand appeared on a MOUNT statement but the corresponding TAPEx DD statement was not found.
2. The DD statement SYSPRINx is missing. If MAXTASKS= was specified, there must be a SYSPRINx DD for every TAPEx DD.
3. An OPEN error occured for the specified ddname. Check the job log for IBM error messages.

**Action**: The operation associated with the failing DD statement will be bypassed. Other operations in the same FDRSOS step may execute.

## 230.22 CONTINUED. . .

### FDR012 NOT AN FDR *type* TAPE DSN=dsname

**Reason**: 1. The backup dataset input for a restore is not a FDRSOS-created backup.
2. The JCL specifies the wrong tape or disk volume serial numbers, or specifies the serials in the wrong order for a multi-volume backup dataset.
3. The FDRSOS backup file may have been written over.
"type" will be BACKUP if the tape is not in FDR format at all, and SOS if it is in FDR format but is not a FDRSOS backup (i.e., is a regular FDR backup). The data set name printed is the backup data set name. A MINI DUMP is printed displaying the first tape block read, which may help you understand what is really in this backup.

**Action**: The restore subtask will abend with U0205. If possible, correct the JCL to point to the proper backup and resubmit.

### FDR014 BACKUP HAS MORE | LESS SECTORS THAN RESTORE DEVICE - RESTORED TERMINATED|CONTINUING

**Reason**: During a RESTORE, FDRSOS determined that the size of the Open System volume that was backed up to the input backup data set was a different size than the target Open System volume. A MINI DUMP is printed displaying the FDRSOS header record.

**Action**: If the backup contained more data sectors, the restore is terminated. If it contained fewer data sectors, the restore will continue but the restored volume may or may not be usable, depending on the system which uses the data.

### FDR020 RACF VOLUME PROTECTION FAILED ON VOL=*vvvvvv*

**Reason**: A security check was done in the DASDVOL class for the volume serial indicated. The user was not authorized to READ that volser for DUMP and PRINT, or ALTER the volser for RESTORE.

**Action**: The restore subtask will abend with U0801.

### FDR031 DD=*ddname* - NUMBER OF CHARACTERS EXCEEDS 5 - STATEMENT IGNORED

**Reason**: FDRSOS encountered a DD statement starting with "DISK" but the total length of the DD name was more than 5 (more than one character after DISK, e.g., DISKAA).

**Action**: The DD statement is ignored. Correct the JCL and resubmit the job.

### FDR032 NO VALID DISK(X) DD STATEMENTS WERE FOUND

**Reason**: FDRSOS did not find any Open System volumes to process. The volumes to process must be specified either by DISKx DD statements or MOUNT VOL= statements. If MOUNT statements were used it may that the volume serials were misspelled, or the volumes have not been made available by a FDRSOS VARYON or LABEL operation.

**Action**: Correct the jobstream and resubmit the job.

CONTINUED. . .

## 230.22 CONTINUED. . .

**FDR042 RESTORE FROM BACKUP OF VOL=*vvvvvv* UNIT=*uuuu* CREATED ON DATE=*yy.ddd* TIME=*hh.mm.ss***

**Reason**: Documents the creation time and date of the backup from which a restore is being done, as well as the volume serial (vvvvvv) and original device address (uuuu) of the disk that was dumped.

**FDR090 DIAGNOSTIC MINI DUMP-FDRDMPRT VER *v.r/nnb*-INNOVATION DATA PROCESSING DATE/TIME – *yydddlhh.mm.ss* PAGE -- *nn***

**Reason**: General page heading for the MINI DUMP processor listing the version, level, date, time and page.

**FDR091 REGS**

**Reason**: General register contents formatted by the MINI DUMP processor.

**FDR092 *type* LEN=*nnnn*(*hhhh)* LOC=*llllll***

**Reason**: Control block formatted by the MINI DUMP processor. 'type' is the name of the control block, 'nnnn' is the length in decimal, 'hhhh' is the length in hex, and 'llllll' is the storage location in hex.

**FDR093 CCWS LEN=*nnnn*(*hhhh*)**

**Reason**: CCWs formatted by the MINI DUMP processor.

**FDR094 *reason* – CCWS NOT PRINTED(TERMINATED)**

**Reason**: CCW formatting was requested in a MINI DUMP. However, for the reason printed, CCW printing was terminated.

**Action**: Contact Innovation technical support for further problem determination.

**FDR099 NEAR RELATIVE DATA LOCATION nnn -- error description**

**Reason**: The common parsing routine encountered an error in parsing the user specified control statements. The approximate location of the error was position nnn, counting the first position as 000.

**Action**: Correct error described in 'error description' and re-submit job.

CONTINUED. . .

## 230.22 CONTINUED . . .

### FDR122 OPERATION STATISTICS FOR SOS VOLUME...*volser*

**Reason**: This is the header for a table of statistics about the dump, restore, or print operation that was performed on the Open System volume indicated. This may include:

**BYTES ON VOLUME** - total size, in bytes, of the Open System volume being processed

**BYTES READ FROM DASD** - total number of bytes actually backed up from the Open System volume (should be the same as "bytes on volume" for full backups).

**BYTES WRITTEN TO DASD** - total number of bytes actually restored to the Open System volume (should be the same as "bytes on volume" for full restores from full backups).

**BYTES ON BACKUP** - total number of bytes actually written to or read from the backup. For uncompressed backups, this will be slightly higher than the DASD bytes because of control data. For compressed backups (COMPRESS= on the DUMP statement), this is the bytes after compression.

**COMPRESSION SAVINGS** - percentage of the backup file size saved by FDRSOS compression (COMPRESS= on the DUMP statement). This is calculated as the "Bytes on Backup" divided by the bytes that would have been written to the backup without compression (not displayed). This will be zero unless COMPRESS=ALL was specified; compression savings from IDRC tape hardware compression is not displayed.

**DASD SECTORS BACKED UP|WRITTEN** - number of 512 byte data sectors read from or written to the Open System volume.

**BACKUP BLOCKS WRITTEN|READ -** number of physical blocks written to or read from the backup file.

**DASD EXCPS** - number of write/read I/O requests issued to the Open System volume.

**BACKUP FILE EXCPS** - number of write/read I/O requests issued to the backup file.

**CPU TIME (SECONDS)** - the CPU (TCB) time required to process this volume, in seconds and thousandths of a second.

**ELAPSED TIME (MINUTES)** - the actual time, in minutes and tenths, required to process this volume.

**BACKUP|RESTORE TIME (EXCLUDING MOUNTS)** - the actual time, in minutes and tenths, required to process this volume, excluding the time spent waiting for input or output tape volumes to be mounted (at OPEN or EOV).

**BACKUP COPY n ON type DSN=dsname VOL=volumes** - documents the location of the backup file created. "n" is the copy number (1 or 2) and "type" is DISK or TAPE.

**RESTORE FROM DSN=dsname** - documents the name of the backup file being restored.

## 230.22 CONTINUED . . .

### FDR129 I/O ERROR ON DISK PACK--LAST SEEK ADDRESS READ X'bbbbbbbb'

**Reason**: An I/O error or logical error was detected by FDRSOS on this volume. "bbbbbbbb" is the sector number (in hex) of the first sector referenced by the I/O that got the error (the actual sector in error may follow that sector). An IBM IOS000I message may also have been printed on the Job log. A FDR149 message will be printed detailing the I/O error

**Action**: FDRSOS will attempt to continue processing from this point, bypassing the data in error, but this may make the backup or the restored disk unusable. If more than MAXERR= errors occur the operation will be terminated with a U0101 ABEND. Contact Innovation technical support for assistance.

### FDR149 I/O TRACE SECT|CCHH=X'*xxxxxxxx*' *mmm.mmm* MSEC *ttt* DD=*ddname* UNIT=*unit*

**CCW/DATA - IOB *iobdata***

**CCW/DATA - SENSE *sensedata***

***ccwdata***

**Reason**: This message is produced for FDRSOS internal traces and is also printed for I/O errors on either the Open System Volume (in which case SECT= documents the relative sector number first referenced by the I/O operation on which the error occured) or the backup file (CCHH= documents the disk location for backup files on DASD, and is meaningless for tape backups).

The elapsed time of the I/O in milliseconds is shown as "mmm.mmm".

"ttt" is an internal clock value.

"ddname" and "unit" document which device this message relates to.

"iobdata" documents the I/O termination status.

If a unit check (device error) occured, "sensedata" will document the sense bytes returned by the device.

"ccwdata" documents the CCW chain that was issued, along with 8 (occasionlly 16) bytes of data associated with each CCW.

**Action**: Check the job log of the FDRSOS job for IBM messages which may relate to the error. If the cause of the error is not obvious, contact Innovation tech support for assistance in interpreting and correcting the error.

### FDR200 BLOCK DROPPED–(synadaf info)–(data block)

**Reason**: FDRSOS or FDRSOSTC has encountered a permanent BSAM read or write I/O error on a backup data set. The cause of the error is documented as "synadaf info" as provided by the IBM SYNAD exit. "data block" is the first 20 bytes of backup data block, if available. An IBM IOS000I I/O error message may also appear in the job's JOB LOG.

**Action**: The step will be terminated with a U0200 abend.

## 230.22 CONTINUED. . .

### FDR203 PREMATURE TAPE END OF FILE DSN=dsname

**Reason**: FDRSOS detected an end-of-file on the backup data set without encountering the FDR trailer identification record. Possible causes are:
– The backup that created this file did not complete successfully.
– The backup run completed successfully, but the backup file was not correctly cataloged.
– The JCL for the restore specifies volume serial numbers for the backup file, but the last volume has been omitted, or the volume serials are out of order.

**Action**: In the last two cases, you can do the restore by specifying the correct volume serial numbers on the DD statement for the backup file.

### FDR204 TAPE BLOCK LENGTH CHECK--BLOCK BYPASSED

**Reason**: FDRSOS records the length of the block written internally. FDRSOSTC detected that the length of the block read from the input did not match the length recorded. An FDR MINI DUMP is printed. If Message FDR204 occurs many times, the cause is probably that you are copying from a tape that was created by using a utility program (such as IEBGENER) to copy an FDRSOS backup file. Most copy utilities cannot copy FDR backup tapes. FDRSOSTC or FATAR (Ver. 4.4) must be used to copy FDRSOS backups.

**Action**: FDRSOSTC will continue processing bypassing this block. Data may be lost from the tape.

### FDR206 TAPE BLOCK COUNT ERROR

**Reason**: The number of blocks read from a backup tape did not match the block count in the tape trailer label. A MINI DUMP is printed of the DCB, UCB and registers. The registers are from the block count exit. Register zero (0) contains the block count from the trailer label, which reflects the number of blocks written during the backup run. The fourth word of the DCB contains the count of blocks read during the restore.

**Action**: FDRSOS will ignore the error and continue processing.

### FDR210 I/O ERROR ON BACKUP DD=*ddname*

**Reason**: An I/O error has occured writing the backup data set. A FDR149 message will follow to document the error.

**Action**: The backup subtask will abend immediately with a U0200 abend.

## 230.22 CONTINUED. . .

### FDR211 FDRSOS ERROR ON DD=*ddname* REASON=*reason*

**Reason**: FDRSOS RESTORE had an error attempting to initialize the DD statement shown, for the reason shown. Reason codes include:
**1 - NO MATCHING MOUNT TAPEDD** - a TAPEx DD was found in the JCL but there was no matching DISKx DD or MOUNT statement to indicate where to restore the backup.
**2 - UNIT/VOLSER NOT SPECIFIED** - UNIT= and/or VOL= must be specified on the MOUNT statement.
**3 - UNIT ADDRESS NOT FOUND** - UNIT= was specified but no offline DASD device with that address was found.
**4 - DD STATEMENT IS MISSING** - TAPEDD=x was specified but no TAPEx DD statement was found.
**5 - VOLSER NOT FOUND** - VOL= was specified but that volser was not found in the UCB of any offline DASD device. You may need to execute the VARYON or LABEL function of FDRSOS.
**6 - TAPEDD NOT SPECIFIED** - TAPEDD= was not specified
**7 - NOT A VALID SOS DEVICE** - the unit or volume specified was not a valid Open System volume.

**Action**: FDRSOS will bypass this DD statement and continue processing.

## 230.22 CONTINUED. . .

### FDR212 FDRSOS ERROR UNIT=*uuuu* REASON=*reason*

**Reason**: FDRSOS had an error attempting to initialize or process the Open System volume on the device address shown as "uuuu", for the reason shown. Reason codes include:

**1 - VARY FAILED - VOLSER vvvvvv IS A DUPLICATE** - operation failed because the volume serial of the Open System volume is already assigned to another Open System or S/390 volume.

**2 - VOLSER FILLED IN BUT DOES NOT RESPOND** - a Open System UCB has a volume serial set, but the device does not respond. Check the EMC Symmetrix subsystem.

**3 - DEVICE IS RESERVED TO ANOTHER MACHINE** - RESERVE=YES was specified (or defaulted) but the Open System volume is reserved to another system.

**4 - NOT A RESPONDING SOS UNIT** - unit is not responding to I/O requests.

**5 - UNIT IS DUPLICATE OR INCLUDED IN PRIOR MOUNT** - the UNIT= value specified on the MOUNT statement is a duplicate of that on a previous MOUNT (or is included in a prefix on a previous MOUNT).

**6 - UNIT DOES NOT CONTAIN SPECIFIED VOLSER** - both UNIT= and VOL= were specified, but the volume serial was not found on any of the devices specified.

**7 - TAPEDD=X FOR MOUNT COMMAND NOT FOUND** - TAPEDD= was omitted when required.

**8 - ERROR READING/WRITING THE LABEL SECTOR** - an I/O error occuring on the volume label from an Open System volume during a VARYON or LABEL operation.

**9 - VOLUME DOES NOT CONTAIN A LABEL** - for a VARYON operation, the volume was not previously intialized by a LABEL operation.

**A - SETVOL MISSING REQUIRED ON LABEL COMMAND** - for a LABEL operation, the SETVOL= operand was omitted from the MOUNT statement.

**B - I/O ERROR ACQUIRING STATUS** - PRINT=STATUS was specified but an I/O error occuring reading the Open System volume.

**C - I/O ERROR ERASING SECTORS** - an I/O error occuring during an ERASE operation.

**D - SETVOL VOLSER CONTAINS INVALID CHARACTERS** - the volume label contained other than alphanumeric and national characters.

**E - UNIT ADDRESS IS INVALID OR NOT 4 CHARACTERS** - in a UNIT= parameter, either non-hexadecimal characters were specified, or the length was less than 4 and did not end with *. Device addresses must be 4-digit addresses.

**F - BCV VOLUME NOT FOUND OR NOT CONNECTED TO BCV** - a BCV operation was requested but FDRSOS was not able to resolve internal Symmetrix device numbers to S/390 device addresses for the primary volume or BCV. Possible causes: your Symmetrix is not configured for BCVs, the BCV=UNIT address is invalid, VARYON was not executed against the devices.

**G - BCV VOLUME NOT SPLIT OR SYNCHRONIZED** - a RE-ESTABLISH was requested, but the BCV volume associated with the primary volume is either still connected (never SPLIT) or a previous SPLIT has not completely synchronized the BCV.

**FDR212 FDRSOS ERROR UNIT=*uuuu* REASON=*reason* (continued)**

**H - MATCHING BCV VOLUME NOT FOUND IN MVS SYSTEM** - a BCV operation was requested, but FDRSOS was unable to identify the S/390 (MVS) device address of the BCV. This will occur if you have not run a VARYON against the BCV device addresses; if this was done, check your Symmetrix and MVS configurations.
**J - CANNOT DO A ESTABLISH/RE-EST FROM BCV VOLUME** - an ESTABLISH/RE-ESTABLISH BCV operation was requested, but the volume specified was the BCV volume, not the primary volume. Change your MOUNT statement to point to the primary.
**K - EMC CODE nn** - **reason** - a BCV operation was requested, but the BCV request to the Symmetrix subsystem failed with the indicated reason code “nn”. “reason” is a brief text explanation of the reason.
**L - CANNOT SPLIT BCV VOLUME PAIR NOT ESTABLISHED** - a SPLIT operation was requested, but the primary volume was never ESTABLISHed with a BCV volume or the BCV has been reused with a different primary volume.
**M - ESTABLISH TO BCVUNIT MISSING OR INCORRECT** - an ESTABLISH operation was requested, but the Symmetrix device specified by BCVUNIT= was not a Symmetrix device or BCVUNIT= was omitted.
**N - ESTABLISH TO BCVUNIT ALREADY IN USE** - an ESTABLISH operation was requested but the Symmetrix device specified by BCVUNIT= was already assigned to another primary volume.
**P - MAXIMUM # OF BCV VOLUMES IN ONE STEP EXCEEDED** - more than 255 BCV volumes were processed in a single FDRSOS step.
**Q - CANNOT INITIALIZE LOCAL BACKUP VOL IN USE** - A LOCALBACKUP TYPE=INIT attempted to initialize a volume which already contained data. If you are sure that data is not required, ERASE the first 1000 blocks of the disk (see example in Section 210.20) and re-execute the LOCALBACKUP.
**R – FULL UNIT ADDRESS MUST BE SPECIFIED** - for a LABEL or LOCALBACKUP operation, the UNIT= operand on the MOUNT statement was a prefix. It must specify a full device address.
**S - SIZE OF LOCAL BACKUP CANNOT BE REDUCED** - a LOCALBACKUP TYPE=UPDATE specified SIZE= on a MOUNT statement, but the local backup volume was already larger than the value given.
**T - VOLUME NOT INITIALIZED FOR LOCAL BACKUP** - a LOCALBACKUP TYPE=UPDATE was executed, but the volume specified by the MOUNT statement did not point to a volume previously initialized as a FDR/UPSTREAM local backup volume.

**Action**: FDRSOS will bypass this Open System volume and continue processing.

**FDR213 FDRSOS PSEUDO MOUNTED UNIT=*uuuu* VOL=*volser* - SYM-NUMBER=X'*ssss-ssss*'**

**Reason**: FDRSOS successfully processed a VARYON statement for the Open System volume on device address "uuuu". Its volume serial is "volser" and its internal EMS Symmetrix ID is "ssss-ssss" (subsystem number and internal device number).

## 230.22 CONTINUED. . .

**FDR214 FDRSOS PSEUDO DIS-MOUNTED UNIT=*uuuu* VOL=*volser***

**Reason**: FDRSOS successfully processed a VARYOFF statement for the Open System volume on device address "uuuu". Its volume serial was "volser".

**FDR215 FDRSOS LABELED UNIT=*uuuu* TO VOL=*volser***

**Reason**: FDRSOS successfully processed a LABEL statement for the Open System volume on device address "uuuu". Its volume serial has been set to "volser".

**FDR216 STATUS OF SOS SCSI VOLUME UNIT=*uuuu* VOL=*volser* - SYM-NUMBER=X'*ssss-ssss*'**
**UNIT=*uuuu* status message**
**UNIT=*uuuu* IS UNKNOWN VOLUME TYPE**

**Reason**: If PRINT=STATUS is specified on a FDRSOS control statement, FDRSOS will attempt to determine the type of each Open System volume processed. A block of FDR216 messages are printed for each volume, identified by its device address "uuuu" and volume serial "volser" and internal EMC Symmetrix ID "ssss-ssss" (subsystem number and internal device number). The header line shown will always appear, following by one or more additional FDR216 messages. For a RESTORE, this information is printed for the volume both before and after the restore. This information may be useful to you to confirm which Open System volumes correspond to which S/390 device addresses.
When the volume type is successfully identified, FDRSOS will show the type of Open System which created the volume (e.g.,. IBM AIX, HP-UX, SUN SOLARIS, NOVELL, OS/2, etc.) plus additional information about the contents of the volume (volume or volume group identifications, volume group information, UNIX mount points, and other information; the details will vary depending on the type of Open system).
If "UNKNOWN VOLUME TYPE" is printed, FDRSOS was unable to identify the type of Open System volume. However, FDRSOS is not sensitive to the format of the volume and will still be able to process it successfully.

**NOTE: The volume information displayed by FDRSOS is based on Innovation's current understanding of the format of volumes created by each of the Open Systems supported, as obtained by research and experimentation. Innovation will update the PRINT=STATUS display in subsequent releases of FDRSOS based on experience. If PRINT=STATUS does not display your Open System volumes correctly, please contact Innovation so that we can obtain information necessary to improve the displays.**

## 230.22 CONTINUED. . .

**FDR217 FDRSOS UNIT=*uuuu* IS A BCV VOLUME stat FROM VOL=*volser* X'status bytes'**

**Reason:** During a VARYON operation, the device identified by the device address ("uuuu") was identified as a EMC Symmetrix Timefinder™ BCV (Business Continuance Volume, see Section 200.05).

"stat" is a brief description of the status of the BCV:

**NEVER BEEN PAIRED** - this BCV has never been ESTABLISHed with any primary volume.

**PAIRED AND SYNCED** - ESTABLISH or RE-ESTABLISH was issued and the BCV is fully synchronized with the primary volume.

**PAIRED AND SYNCING** - ESTABLISH or RE-ESTABLISH was issued but the BCV is not yet fully synchronized with the primary volume.

**SPLIT IN PROGRESS** - a SPLIT command is executing.

**SPLIT AND SYNCED** - a SPLIT command was executed and the BCV volume is complete split and ready for use.

**SPLIT AND SYNCING** - a SPLIT command was executed but the BCV is not yet completely split and ready for use.

The "status bytes" are 16 hex bytes of status from a special QUERY to the BCV.

**FDR218 FDRSOS UNIT=*uuuu* WAS stat BCV UNIT=*bbbb* - WAIT FOR SYNCHRONIZATION DEFERRED**

**Reason:** EMC Timefinder™ BCV volumes are being used for FDRSOS backups (See Section 200.05). This message indicates the status of the BCV volume as a result of the FDRSOS operation requested (ESTABLISH, RE-ESTABLISH, SPLIT or RESTORED). "uuuu" is the S/390 address of the primary volume and "bbbb" is the address of the BCV.

FDRSOS will initiate the requested operation on all volumes specified by MOUNT statements but will not wait for the function to complete within the Symmetrix subsystem. If BCV=WAIT was specified on the FDRSOS statement, the step will not end until the requested operation has completed on all volumes (indicated by MVS console message FDRW20); if omitted, the step will end as soon as all operations have been accepted by the Symmetrix.

. "stat" is text which indicates the volume status:

| | |
|---|---|
| **CONNECTED TO** | ESTABLISH was executed |
| **RE-CONNECTED TO** | RE-ESTABLISH was executed |
| **SPLIT FROM** | SPLIT was executed |
| **RE-SPLIT FROM** | SPLIT discovered that the BCV was already split from the primary, so it RE-ESTABLISHed it to refresh the BCV and SPLIT it again |
| **ALREADY SPLIT** | SPLIT discovered that the BCV was already split |
| **ALREADY PAIRED** | ESTABLISH or RE-ESTABLISH discovered that the BCV was already connected to the primary volume. |

## 230.22 CONTINUED. . .

### FDR219 FDRSOS UNIT=*uuuu* IS BEING DUMPED FROM BCV UNIT=*bbbb*

**Reason:** EMC Timefinder™ BCV volumes are being used for FDRSOS backups and BCV=USE was specified on the DUMP statement (See Section 200.05). This message indicates that primary volume "uuuu" is actually being dumped from its frozen copy on BCV unit "bbbb".

### FDR220 FDRSOS UNIT=*uuuu* BCV SYNCHRONIZATION COMPLETE

**Reason:** EMC Timefinder™ BCV volumes are being used for FDRSOS backups and BCV=WAIT was specified on the ESTABLISH, RE-ESTABLISH or SPLIT statement (See Section 200.05). FDRSOS will initiate the requested operation on all volumes in parallel and will print this message as each one completes. When all are complete, the step will end.

### FDR221 TAPEOUT SYNAD ERROR, DCB ADDRESS UNIDENTIFIED.

**Reason:** FDRSOSTC entered its SYNAD routine for TAPEOUT and the DCB passed was not the expected value. This is an internal error and should not occur.

**Action:** A U0200 ABEND will be issued with Reason Code 221 (in register 15). Examine Joblog and Syslog for any messages that may be the cause of the error on the TAPEOUT device. Correct these errors and resubmit. If necessary, contact Innovation Technical Support for assistance.

### FDR222 ERROR UPDATING VOLINFO REASON=*reason*

**Reason:** During a RESTORE with VOLRESET=NO, FDRSOS encountered an error attempting to preserve all of the volume ID information on the output Open System Volume. “reason” documents the error:

**1 - AIX MAXIMUM LOGICAL VOLS EXCEEDED** - while attempting to update the logical volumes on an IBM AIX/6000 volume group, more than 255 logical volumes were encountered. This probably indicates that the Open System volume, although it appears to be formatted by AIX, is not properly formatted or has been overlaid.

**3 - NO LOGICAL VOLUME MANAGER ON BACKUP** - a HP/UX disk did not contain logical volume manager footprints.

**4 - PVID IN PHYSICAL HEADER MISMATCH** - a HP/UX disk contained different PVIDs (Physical Volume IDs) in several control blocks.

**5 - VGDA VOLUME GROUP ID DOES NOT MATCH** - a HP/UX disk contained different VGIDs (Volume Group IDs) in several control blocks.

### FDR223 FDRSOS CREATED/UPDATED LOCAL BACKUP PARTITION

**Reason:** A LOCALBACKUP statement was executed successfully. An Open System volume was initialized for use as a FDR/UPSTREAM local backup (TYPE=INIT) or an existing local backup volume was updated with new defaults (TYPE=UPDATE)

## 230.22 CONTINUED . . .

### FDR302 CONTROL STATEMENT ERROR NEAR REL LOCATION nn -- REASON x -- JOB TERMINATED

**Reason**: An error was encountered during the processing of a user-supplied control statement.If "NEAR REL LOCATION nn" appears, the keyword or operand causing the error is at or near column "nn" on the input statement. The single character reason code defines the error:

**1** – A SELECT/EXCLUDE statement did not specify any operands.Control statement was blank after the Command name.

**2** – Command on the first control statement was incorrectly specified; i.e.: was not DUMP, RESTORE, PRINT, ERASE, LABEL, VARYON, or VARYOFF.

**3** – Operand on the first control statement was incorrectly specified

**4** – Operand did not end with a blank or comma.

**5** – SYSIN data set was empty.

**6** – Expected continuation statement was not found. The previous statement ended with a comma and a blank.

**7** – Invalid or incompatible operands were specified or a required operand was omitted.

**8** – An operand on a SELECT/EXCLUDE statement specified a blank or comma after the equal sign.

**9** – One of the operand values exceeded the maximum permitted number of characters or digits.

**F** – An operand requiring a numeric value had non-numeric characters.

**G** – An operand requiring a numeric value had too many digits.

**I** – Keyword is invalid under the operation indicated.

**J** – Control statement was completely blank.

**K** – A required operand such as STARTBLK= was not specified on the preceding SELECT/EXCLUDE.

**Q** – Keyword exceeded maximum value or was negative.

**R** – TYPE=xxx was specified multiple times.

**S** – An operand on the SELECT-type Command was specified multiple times or was mutually exclusive with another operand.

**U** – The TYPE=xxx operand was missing or invalid on the DUMP, RESTORE, ERASE or PRINT statement.

**V** – The FROMBLK= value was greater than the TOBLK= value.

**Action**: The FDRSOS step is terminated; no further control statements are read. Correct error and resubmit job.

### FDR303 CARD IMAGE--control statement image PARM ENTRY

**Reason**: Display all input control statements from the SYSIN data set. If 'PARM ENTRY' appears after the control statement, entry was passed from the PARM FIELD on the EXEC statement.

## 230.22 CONTINUED . . .

### FDR316 FDR DID NOT FIND REQUESTED *item*

**Reason**: This message is displayed when the statement identified by "item" did was not processed. This could occur when:
1) a MOUNT did not select any Open System volumes, perhaps because the UNIT= and/or VOL= operands were incorrectly specified or the volume specified by VOL= has not been mounted by a VARYON statement.
2) a SELECT/EXCLUDE statement specified a TAPEDD= or VOL= which did not match any volume being processed.

**Action**: Be sure that you specified 4-digit device addresses (for example, UNIT=01E0 or UNIT=01F*) even if your operating system supports only 3-digit addresses; add a leading zero if necessary. Verify that the UNIT=, TAPEDD=, or VOL= specifies values appropriate for your configuration and jobstream.

### FDR319 FDR OPERATION ABNORMALLY TERMINATED COMP CODE=S*sss* U*uuuu*

**Reason**: A FDRSOS dump or restore subtask has abnormally terminated with either a system ABEND code of "sss" or a user ABEND code of "uuuu" (which ever is non-zero). Check IBM documentation for the meaning of system ABENDs.

**Action**: FDRSOS will terminate this Open System volume and continue processing. If your JCL includes a SYSUDUMP or SYSABEND DD statement, a diagnostic dump will be produced in most cases. Check for other FDR messages which may be printed prior to the ABEND; these may help you understand the error. Check IBM documentation for the meaning of system ABENDs, see Section 230.23 for the meaning of user ABENDs. If you are unable to determine the cause of the ABEND, contact Innovation tech support; have all messages and the diagnostic dump available.

### FDR324 DDNAME=ddname MISSING OR IN ERROR REASON x

**Reason**: The DD statement identified by "ddname" had an error identified by the single character reason code:

**1** – The SYSIN DD statement was missing or could not be opened. A U0401 ABEND will occur.

**E** – The "DISKx" DD statement appeared more than once.

**F** – DISP=MOD was specified on a TAPEx or TAPExx DD statement. This may also mean that you directed multiple backups to the same TAPEx DD by the same TAPEDD=x on multiple MOUNT statements.

**G** –Both TAPExx and TAPExx point to backup data sets on disk. You cannot create 2 simultaneous disk backups. You can create simultaneousbackups on tape and disk.

## 230.22 CONTINUED . . .

### FDR336 DYNAMIC ALLOCATION ERROR COMP=cc, CODE=nnnn, INFO=iiii, DDNAME=ddname

**Reason**: Dynamic allocation was requested for a disk volume or a data set (such as a disk or tape backup data set), but the allocation failed. "cc" is the return code in REG 15, "nnnn" and "iiii" are from the dynamic allocation parameter list. These are documented in the IBM manuals *SPL: System Macros and Facilities* for MVS/XA, and *Authorized Assembler Services Guide* for MVS/ESA and OS/390. Frequently encountered CODE values include:

210– requested data set not available (e.g., another job had the required backup data set allocated with DISP=OLD).

220– requested volume not available (e.g., another job is using a tape volume required).

484– the operator replied "NO" to Message IEF235D.

1708– data set not found in the catalog (e.g., BACKUPDSN= was misspelled).

**Action**: This volume or data set operation will be bypassed. If necessary, contact INNOVATION for assistance.

### FDR370 OPEN ERROR -- TAPE BYPASSED DSN=dsname

**Reason**: An error occured opening the backup data set for a dump or restore.

**Action**: The operation is bypassed. Check for IBM messages in job log. If necessary, contact Innovation tech support for assistance.

### FDR390 NO SECTORS MET SELECTION CRITERIA

**Reason**: A TYPE=PARTIAL operation was executed, but the range of data specified by FROMBLK= and TOBLK= did not exist on the Open System volume processed. The values specified were probably larger than the size of the volume.

## 230.22 CONTINUED . . .

**FDR400 FDR SAFEGUARD OPEN STORAGE -- FDRSOSTC VER *v.r/mmb* –INNOVATION DATA PROCESSING DATE=*yy.ddd* PAGE *nnn***

**Reason**: This is the FDRSOSTC page heading, specifying the version level of FDRSOSTC. "v.r" indicates the version and release (e.g., 5.3), "mm" is a 2-digit number indicating the maintenance level and "b" will be "P" for a production version, "T" for a trial, and "B" for a beta test version.

**FDR402 INVALID CONTINUATION**

**Reason**: FDRSOSTC detected that the last control statement ended with a comma but no continuation statement was found.

**Action**: Correct and resubmit job.

**FDR446 1ST RECORD NOT FDR HEADER**

**Reason**: The first record read from the backup data set was not the expected special FDRSOS header record.

**Action**: A U0626 ABEND will be issued. Check that the input data set is a FDRSOS backup and resubmit. Contact Innovation Technical Support for assistance if the problem recurs.

**FDR447 CONTROL RECORD NOT FOUND**

**Reason**: Each set of backup records is preceded by a control record describing them. A control record was not found when it was expected. This may be due to tape errors.

**Action**: A U0626 ABEND will be issued. Contact Innovation Technical Support for assistance.

**FDR450 TAPEIN SUCCESSFULLY COPIED TO TAPEOUT--BLOCKS READ nnnnnn BLOCKS WRITTEN nnnnnn**

**Reason**: One file has been successfully copied from TAPEIN to TAPEOUT (and optionally to TAPE2OUT as well) by FDRSOSTC. Blocks read and blocks written should be the same. This message will be followed by Message FDR615 detailing the data sets read and written.

**FDR451 BLOCK GREATER THAN 32760 TO OUTPUT ON DISK ...JOB TERMINATED**

**Reason**: FDRSOSTC detected the input file was on tape and blocked at over 32k and the output file to TAPEOUT is on Disk (Initial release of FDRSOSTC does not support this function. A future release will eliminate this restriction)

**Action**: A U0502 ABEND is issued. Contact INNOVATION for assistance after obtaining a storage dump.

## 230.22 CONTINUED . . .

**FDR465 function FAILED ON DDNAME=ddname-- MISSING OR MISSPECIFIED.**

**Reason**: a) ddname printed is missing or misspelled.
b) OPEN failure occurred for ddname specified in message.
c) READ JFCB failure occurred for ddname specified in message.

**Action**: A U0629 ABEND occurs. Correct the specified DDNAME and resubmit the job.

**FDR484 INTERNAL LOGIC ERROR--JOB TERMINATED**

**Reason**: FDRSOSTC has encountered an unexpected error condition.

**Action**: If this message is accompanied by another error message see that message for appropriate action. Otherwise insure the SYSUDUMP dd was included in the job and obtain a storage dump then call Innovation Technical Support.

**FDR498 program (version) PROCESSING COMPLETED WITH ERRORS**

**Reason**: The program listed in the message completed the required processing but recoverable errors were encountered.

**Action**: Check the output, correct and rerun or restart as appropriate.

**FDR499 program (version) PROCESSING COMPLETED**

**Reason**: The program listed in the message completed the required processing.

**FDR516 Function REGISTERS -- R0=xxxxxxxx R1=xxxxxxxx R15=xxxxxxxx**

**Reason**: A CAMLST catalog request or an OBTAIN request failed. The type of CAMLST function and the registers make diagnosing the problem relatively simple. The return codes from catalog management are documented in the IBM manuals SPL: DATA MANAGEMENT for MVS or CATALOG ADMINISTRATION GUIDE for MVS/XA and MVS/ESA.

**Action**: Check the return code(s) and take corrective action if the error is apparent or call INNOVATION technical support for assistance.

## 230.22 CONTINUED . . .

### FDR604 ERROR OCCURED POSITIONING TAPEIN REASON CODE-x

**Reason**: FDRSOSTC detected an error when positioning the input tape. The reason code x may be:

| REASON | EXPLANATION |
|---|---|
| 1 | Tape mark not read when expected. |
| 2 | Tape mark read when not expected. |
| 3 | Trailer label read is not EOV1/EOF1. |
| 4 | Header label read is not HDR1. |
| 5 | Tape drive unit check during positioning. |
| 6 | Other I/O error during positioning. |

**Action**: A U0659 ABEND follows. This may be due to hardware problems with the tape drive. If it occurs with the same input tape on more than one tape drive, contact INNOVATION for technical support.

### FDR615 ddddddd DSN=dsname FILE=fn VOL=vvvvvv...

**Reason**: Printed by FDRSOSTC at the end of each file copied. One FDR615 message will be printed for TAPEIN, TAPEOUT, and TAPE2OUT (if present). It documents the input or output DDNAME, data set name, file number (0 if on disk), and up to 20 volume serials.

### FDR997 FDR ABNORMALLY TERMINATED VOL=vvvvvv

**Reason**: This FDRSOS subtask for the volume indicated has encountered an error from which it cannot recover. Other messages may be printed to detail the error.

**Action**: The subtask will terminate with a user ABEND; see Section 230.23 to interpret the abend code. If necessary, contact Innovation tech support for assistance.

### FDR998 FDR DUMP|RESTORE COMPLETED WITH ERRORS VOL=vvvvvv

**Reason**: This FDRSOS subtask for the volume indicated ran to completion but errors occured during the operation. Other messages may be printed to detail the error.

**Action**: Review the messages to determine if corrective action is necessary. If necessary, contact Innovation tech support for assistance.

### FDR998 FDR COMPLETED WITH ERRORS

**Reason**: The FDRSOS step had one or more errors. Other messages may be printed to detail the errors. The step completion code will be set to 12.

**Action**: Review the messages to determine if corrective action is necessary. If necessary, contact Innovation tech support for assistance.

### FDR999 FDR SUCCESSFULLY COMPLETED

**Reason**: The FDRSOS step completed without errors. The step completion code will be set to 0 (zero).

## 230.23 FDRSOS ABEND CODES

FDRSOS or its utility programs may abend with any of the following user abend codes. In many cases, a diagnostic message is printed before the abend, so look up any error messages that were printed first. If no message was printed that relates to this abend, then read the explanation below. Call Innovation technical support if you need assistance understanding or resolving the error.

U0100 **Open Error Trying to Open a DASD or Tape DCB**
Usually preceded by a FDR324 message. Check the job log for IBM messages which may indicate the reason for the error.

U0101 **Maximum I/O Errors Exceeded on a Direct-Access Device**
More disk I/O errors than are allowed by the MAXERR= operand occurred. If you want to complete the operation in spite of the errors, specify a larger MAXERR= value. However, many data sets may be invalid.

U0200 **I/O Error on Backup Data Set**
A permanent I/O has occured on the backup tape or disk data set. The backup is not usable.

U0201 **Premature End-of-File on Input**
The end of a backup file was reached without finding the special FDR trailer record.

U0204 **Invalid Block read**
An invalidly formatted block was read from a backup.

U0205 **Not an FDRSOS Backup**
A restore detected that the backup was not created FDRSOS.

U0210 **Internal error in Compress.**

U0401 **SYSIN DD Statement Error**
SYSIN DD statement missing or incorrectly specified or I/O error on SYSIN data set.

U0402 **SYSPRINT/SYSPRINn DD Statement Error**
One of the above DD statements is missing or incorrectly coded or I/O error occurred processing the data set. There must always be SYSPRINT DD statement, and there must be a SYSPRINn DD for each TAPEx DD if MAXTASKS=n was specified

U0502 **One or more Control Statements are in Error**

U0600 **Required DD Statement is Missing or in Error**

U0626 **An FDR/DSF/ABR Tape Format is in Error**

U0659 **Internal logic error**

U0660 **Stack storage management error**
An internal error occured.

U0801 **Restore Cancelled by Operator or security failure**

U0888 **Copy completed with errors**
Errors occured during a FDRSOSTC step but they were not severe enough to cause the entire COPY to be terminated. The listing should be examined to see what errors occured.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

## A.1. FDRSOS - Windows NT

**Introduction**

FDRSOS includes several Windows NT utility programs which facilitate the use of FDRSOS on volumes used by Windows NT. It includes:

- A facility where a Windows NT attached EMC volume (one or more) can be locked throughout an FDRSOS backup or restore. This facility also assures that if the operating system definition of the drive is changed because you have restored a different drive to it, the operating system information about the drive is refreshed
- A utility to display FDRSOS information about a volume from Windows NT
- A utility to generate test data
- The volume locking facility is useful when implementing FDRSOS during:
- A backup when you wish to assure that the drive(s) are not being accessed and you wish to guarantee the integrity of the data
- A restore when you wish to assure that no users are going to be affected during the restore process.
- A restore when you are restoring an older version of an entire drive (or drives) or when you are moving the backed up contents of a drive to a different drive and wish the operating system to be notified without having to shut down the operating system

The drive can remain locked until:

- A user presses a key
- A predefined timeout occurs
- Another program is run that tells the first program to unlock the drive.

***WARNING: during a FDRSOS restore, you must have the volume locked to insure that the Windows NT system definitions for the drive are reset after the restore.***

There is also a program to determine the host volume name of an EMC disk labeled by FDRSOS. This program, HOSTDISK displays the host volume label when supplied the Windows NT physical disk number.

**INSTALLATION**

The diskette or CD-ROM provided with FDRSOS has only a few files on it. Create a directory on your Windows NT system and copy the NT files into it. Here are the programs (the names shown are relative to the root directory of the diskette or the SOS\ directory of the CD):

| File Name | Description |
|---|---|
| GENDATA.EXE | test data generation utility |
| NT\HOSTDISK.EXE | utility to display the FDRSOS volume label |
| NT\REMOUNT.EXE | utility to unlock a previously locked set of volumes |
| NT\UNMOUNT.EXE | utility to lock and dismount one or more volumes |
| NT\UNMOUNTX.EXE | internal program called by UNMOUNT. |

## APPENDIX A. (CONTINUED...)

### VOLUME SETS

Windows NT provides a facility where a number of disk partitions across one or more physical disks can be grouped together to form a single logical drive. Microsoft calls this a volume set.

The definition of a volume set is not stored on the volume, but in the Windows NT registry, in the DISK subkey (HKEY_LOCAL_MACHINE\SYSTEM\DISK). Thus, when using FDRSOS to backup a volume set, you must be sure to:

- Backup all the physical drives in the set. Note that HOSTDISK.EXE (described below) may help you to identify these disks. We recommend that the volume set be UNMOUNTed (see the UNMOUNT program below) or the system be down when the volume set is backed up. This is not required but recommended to assure file and system integrity.
- Backup the Windows NT registry.

To backup the Windows NTregistry you can use the Windows NTprogram RDISK, or a full featured Windows NT backup product such as FDR/UPSTREAM.

There are a number of different scenarios for restoring volume sets:

- Restoring to their original location. This assumes that you are restoring to the same system and have registry backups.
- Restoring to a different machine.
- Restoring to a different location on the same machine. On the original machine, restoring the disks of a volume set to a different group of physical disks. This procedure allows you to be able to copy files off of an FDRSOS backup without having to backlevel your existing disks.

***WARNING: This entire process is VERY DANGEROUS and should be practiced on non-production machines by trained personnel before using it on production systems.***

### RESTORING TO THEIR ORIGINAL LOCATION

To restore a volume set to its original machine use the following procedure:

- Restore the registry. You must at the minimum restore the DISK subkey of the SYSTEM registry hive. You can use the Windows NT setup utility and your emergency repair disks, or a full featured backup/restore product such as FDR/UPSTREAM. If you are using FDR/UPSTREAM you should restore the \WINNT\System32\Config\SYSTEM file and reboot your machine.
- With the machine powered off, perform the FDRSOS restore of all the disks in the volume set.
- Restart your machine. Your volume set drive letter should be restored. If you have any questions, you should contact Innovation Technical Support.

## APPENDIX A. (CONTINUED...)

### RESTORING TO A DIFFERENT MACHINE

The following procedure can be used to restore a set of disks to a different machine than the one it was backed up from or to restore them to the original machine if you do not have a backup of the DISK key in the registry.

The target disks must be exactly the same size as the source disks. Note that the Windows NTDisk Administrator may display these disks with slightly different sizes. If the disks appear to be different sizes, use the FDRSOS PRINT=STATUS to verify that the sizes are the same.

Note that this procedure is very similar to the procedures described in Chapter 22 of the Microsoft Windows NT Workstation Resource Kit, in the section titled *FTEdit - Recovering Volume Sets* and *Stripe Sets*. You will need the programs included on the Resource Kit CD including DISKMAP, FTEDIT and DSKPROBE.

On the machine which contains the original backup, you will need to know the following:

- The physical disk numbers of the disks in the volume set.
- The signatures of each of the disks.
- Their order in the volume set by partition and signature.
- The DISK key disk number.

There are a number of steps to this process.

Run the **Windows NT Disk Administrator** by selecting it from the Administrative Tools group of the Programs menu of the Start button. If the Disk Configuration button is pressed, you will see the physical disks on your system. Note the Physical Disk number of the disks in your volume set.

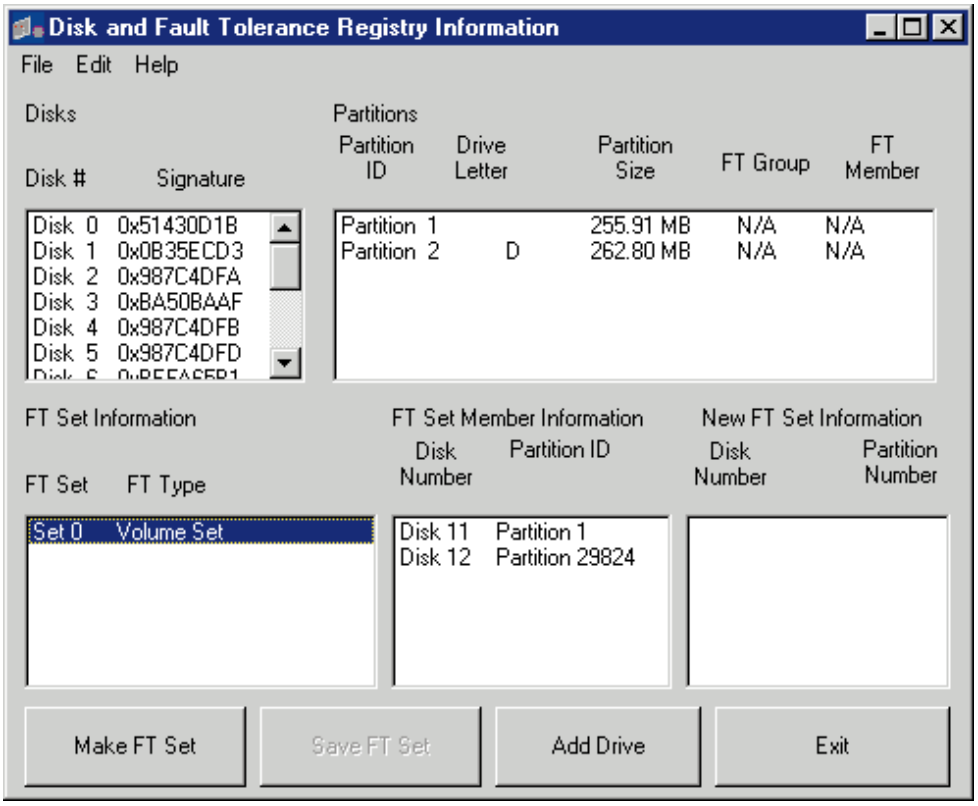
The Windows NT Resource Kit CD contains a number of programs which are used in this process. The command line program DISKMAP is used to be able to obtain the disk signature (which Windows NT places on a disk to uniquely identify it) for a given physical disk number. From a DOS prompt, run the program DISKMAP, with the physical drive number of the disk you wish to examine. For example, obtain the signature for disk #13, enter:

```
DISKMAP /D13
```

The signature is on the 4th line, left justified. You will need the signatures for all the disks in the volume set.

## APPENDIX A. (CONTINUED...)

Next, execute the Windows NT Resource Kit program FTEDIT (see figure below).




In the top left listbox (Disks), disks are listed by their order in the DISK key in the registry - not by their physical disk number on your system. In the bottom left list box (FT Set Information) highlight a volume set. The bottom center listbox (FT Set Member Information) will list the disk number (by registry DISK number) and the partition that the disk is in. From the FT Member Set Information you get the DISK key disk number, which you then use to obtain the Signature for the disk from the Disks list.

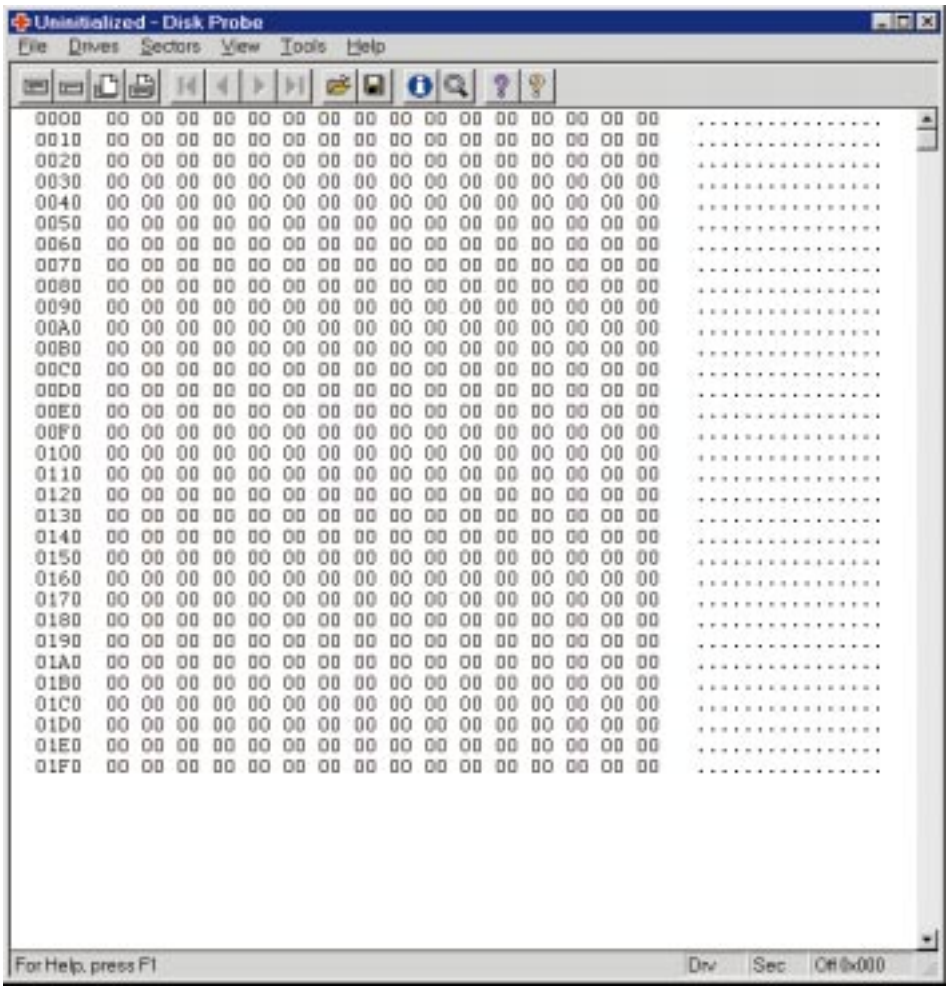
You will need the order of the disks and partitions in the volume set by their signature.

On the machine that you are restoring to, we recommend that BEFORE you perform the restore, you clear out the boot sectors/partition tables on the disks and run Disk Administrator to have the DISK key reflect that the disk is uninitialized. This is an optional step but one which will assure that the system is not confused by prior disk information.

Run the Disk Administrator, and note the physical disk numbers of the target disks and verify that they are the same size as the original disk. If they are slightly different sizes, this may be due to the way that Windows NT calculates sizes; use the FDRSOS PRINT=STATUS command to obtain their exact sizes.

Run the Windows NT Resource Kit program DSKPROBE from a DOS prompt (see figure below). DSKPROBE will be used to initialize the boot sector/partition table to all zeros.

## APPENDIX A. (CONTINUED…)




Pull down the **Drives** menu and Select **Physical Drive**. Highlight the Physical Drive you wish to initialize, uncheck the **Read** only box in the Handle 0 frame, and press the **Set Active** button. Press the Ok button to return to the edit screen. Pull down the Sectors menu and select **Read**. Enter a Starting Sector of 0, a number of sectors as 1 and press the **Read** button. On the DSKPROBE main screen, type in 00 for every byte in the sector. When all bytes are zeroed out, pull down the **Sector** menu, select **Write**, validate that the handle and all other parameters are correct and press the **Write it** button. Exit DSKPROBE.

Run Disk Administrator, do not allow it to write a signature for the modified disks and exit the program normally to have it update the DISK key in the registry with the modified information.

Run the FDRSOS restore of all the disks in the Volume Set. You must either not specify a VOLRESET statement or specify it as VOLRESET=YES (which keeps the data on the volumes from being modified by FDRSOS).

## APPENDIX A. (CONTINUED...)

Run the Disk Administrator to have it update the DISK key with the new disk signatures.

Run FTEDIT from a DOS prompt to create the volume set. Press the **Make FT Set** button and press the **Volume Set** radio button. Identify the first disk by signature in the Disks list and highlight the disk. In the Partitions list, double-click the partition of the disk used for the volume set which will add it to the Volume Set Information frame. Repeat this process for all the disks in the volume set. It is critical that they be added in the same order as they were in their initial configuration. Press the **Save FT Set** button to save your changes to the registry. Exit FTEDIT.

Restart your system.

Enter the disk administrator, highlight the volume set, and using the right mouse button, select **Assign Drive Letter** on the pop-up menu. Assign a drive letter to the volume set.

The Volume Set should now be created successfully. You should verify that files from a variety of locations on the disk are fully accessible.

### RESTORING TO A DIFFERENT LOCATION ON THE SAME MACHINE

The procedure is almost the same as Restoring to a different machine with the following exceptions:

- When you run Disk Administrator after running the DSKPROBE and before performing the FDRSOS restore, allow it to write the signature to the disk.
- In the FDRSOS restore, you must specify VOLRESET=NO to assure that the volumes are restored with different signatures.

### NOTES

Some notes on this facility:

- The DISK key in the registry (HKEY_LOCAL_MACHINE/SYSTEM/DISK) can actually get so mangled that backups and restores of it will not work correctly. In these cases you will need to use the registry editor to delete the DISK key, enter Disk Administrator to rebuild the key, and then use FTEDIT to rebuild the volume sets.
- You may need to activate/deactivate extended LUN support in the SCSI BIOS on your target system to have it match the original system to have the disks be visible with the correct size.

## APPENDIX A. (CONTINUED...)

### UNMOUNT.EXE

UNMOUNT.EXE is a 32-bit MS-DOS program that can lock one or more NT drives and force a refresh of the NT drive information when it is unlocked. If you wish, the program can be run as a service; contact Innovation for more information. UNMOUNT is a necessity when you are restoring an NT volume so that NT will recognize the structure of the restored data. UNMOUNT can also be used during backups to insure that no one can update the volume during the backup.

Unless you specify the /p option, UNMOUNT must be left running during the backup or restore to assume the the drive remains locked. During testing you should verify that the drive is locked during SOS operations by attempting to access the drive from Windows NT (such as a DIR command). You should get a "access is denied" error message.

UNMOUNT will fail if **ANY** programs are accessing the drive when it is executed. You must insure that all users and programs have closed all open files on the drive before running UNMOUNT to lock it. If this or any other error occurs, the program will write an error to the screen and set a non-zero return code.

Run this program from an MS-DOS prompt or batch file. The command format is:

UNMOUNT <drive letter:> ... [/n<name>] [/k] [/w<time>] [/p]

Where:

- **<drive letter:>** is a list of one or more drives to be locked, separated by spaces. For example **C**: or **E**: **F**: **G**:. You must specify at least one drive letter.
- **/n<name>** is an optional parameter which allows you to specify the semaphore name used when running REMOUNT. You need to specify this parameter only if you will be running REMOUNT and you wish to run more than one copy of UNMOUNT.EXE concurrently. For example, /nREMOUNT2.
- **/k** is an optional parameter that locks the drive until the user presses a key.
- **/w<time>** is an optional parameter that locks the drive for a given number of seconds. For example, for 5 minutes, specify /w300.
- **/p** is an optional parameter that causes UNMOUNT to terminate when the drive is locked. The drive will remain locked until the REMOUNT program (see below) is run. The /p option is particularly useful since it returns an immediately return code indicating if the drives were successfully locked.
- **/f** is an optional parameter which instructs UNMOUNT to open the drives specified, flush all buffers to disk and return without locking the drive. This may be useful during backups when you cannot afford to lock the drive but wish to have the most current information possible backed up.

UNMOUNT returns a code of 0 if all drives were successfully locked and 1 if any errors occurred.

Unless you specify the /f, /k or /w operands, the drive will remain locked until the REMOUNT program is executed.

## APPENDIX A. (CONTINUED...)

If you wish to lock drives D, E, and F until REMOUNT is run, specify:

```
UNMOUNT D: E: F:
```

If you wish to lock drive D until you press a key, specify:

```
UNMOUNT D: /k
```

If you wish to lock D and E but unlock them with REMOUNT at separate times (see below for REMOUNT examples), specify:

```
UNMOUNT D: /nDDRIVE
UNMOUNT E: /nEDRIVE
```

In a batch file, the code to unmount the C drive, check the return code and return might look like:

```
UNMOUNT C: /p
if ERRORLEVEL 1 goto ERROR
Echo Successful lock – can begin SOS backup
goto SOS
:ERROR
Echo Unsuccessful lock – perform the SOS backup later
goto SOSLATER
```

### REMOUNT.EXE

REMOUNT.EXE is a 32-bit MS/DOS program which will unlock an NT drive and refresh the operating system's view of the contents of the drive. If you have executed UNMOUNT without the /f, /k and /w operands, you must execute REMOUNT to terminate UNMOUNT and activate the drive. Unless you used the /p operand on the UNMOUNT statement, you must run REMOUNT in a separate MS-DOS session from UNMOUNT.

The command format is:

```
REMOUNT [/n<name>]
```

Where:

- **/n<name>** is an optional parameter. If the UNMOUNT program was executed with the /n option, the same value must be specified on REMOUNT to unlock those drives. If omitted, REMOUNT will unlock all drives locked by an UNMOUNT with no /n parameter.

REMOUNT returns a code of 0 if all drives were successfully unlocked and 1 if any errors occurred.

In the first UNMOUNT example above, where drives D, E and F were locked, run this to unlock those drives:

```
REMOUNT
```

In the second UNMOUNT example above, where drives D and E were locked separately, unlock those drives with:

```
REMOUNT /nDDRIVE
REMOUNT /nEDRIVE
```

## APPENDIX A. (CONTINUED...)

### HOSTDISK.EXE

HOSTDISK.EXE is a program which displays, from Windows NT, information about a disk used with FDRSOS. At the minimum, it will display the MVS volume serial which you have assigned to the disk via the FDRSOS LABEL statement. This helps to verify that you are properly indentifying the FDRSOS volume serial of a given Windows NT disk.

What HOSTDISK does is use EMC-specific SCSI commands to read special tracks reserved for use by FDRSOS.

***WARNING: When run on BCV volumes, HOSTDISK may either hang or report the label for the last disk it was connected to. Thus, we recommend that it not be used for BCV volumes.***

HOSTDISK is a Windows NT console program which is generally run from a command prompt. The format is:

```
HOSTDISK  #<disk number>  [/d]
```

Where:

- **<disk number>** is the physical disk number from 0 (zero) to the total number of disks known to NT. Disk numbers can be determined by running the Windows NT **Disk Administrator**, found in the Adminstrative Tools group of NT. It displays each physical disk, along with the drive letters it is assigned to (if any).
- ***/d*** is an optional parameter which displays some disk and adaptor information in detail.

For example, if the disk administrator shows that disk #2 is your F: drive, issue

```
HOSTDISK #2
```

Which will display:

```
Hostdisk – determines host information for a given EMC drive
Test Physical drive #2
FDRSOS MVS label: E#01E7
```

If you use the detail switch:

```
HOSTDISK #2 /D
```

You would see

```
Hostdisk – determines host information for a given EMC drive
Test Physical drive #2
Details enabled
Drive details for drive #2:
   Removable      : No
    Wide 16-bit   : Yes
    Wide 32-bit   : No
    Vendor ID     : EMC
    Product ID    : SYMMETRIX
    Revision level : 5265
FDRSOS MVS label: E#01E7
```

## APPENDIX A. (CONTINUED...)

### GENDATA.EXE

GENDATA is a bound program (DOS and OS/2) which allows you to quickly create a very large file (up to 4GB) and verify its integrity. It can be used generate test data to be backed up by FDRSOS, and to verify its contents after a FDRSOS restore. It's form is:

GENDATA <file name> <size in bytes> [/f<fill value>] [/r]

Where:

- **<file name>** is the fully qualified name of the file to be created or verified
- **<size in bytes>** is the size of the file to be created or verified (for a verify, specify the same value used to create the file).
- **/f<fill value>** is an optional parameter that allows you to be able to specify a single data byte to repeatedly place in the generated file. It can have values from 0 to 255. If you do not specify the /f option, GENDATA will create the file with 4-byte numbers indicating the long word position. GENDATA will run faster with the /f option than without it. For a verify, specify the same value used to create the file.
- **/r** causes GENDATA to verify the expected contents of a file. This option is generally used after a restore to verify that the data was replaced correctly.

For example, to create a 20,000,000 byte file name C:\TEST.DAT, specify:

GENDATA C:\TEST.DAT 20000000

To create a 1,000,000,000 byte file containing all zeros quickly specify:

GENDATA C:\TEST2.DAT 1000000000 /f0

To verify that the last file was restored correctly:

GENDATA C:\TEST2.DAT 1000000000 /f0 /r

## B.FDRSOS - NetWare

### Introduction

FDRSOS includes a facility where Novell NetWare attached EMC drives (or any type of drive) can be mounted/dismounted throughout an FDRSOS backup or restore. While you can always manually mount/dismount a volume from the system console, this facility allows the process to be automated from a workstation and/or from the server. This facility also allows a workstation to load or unload an NLM and run an NCF server batch file on NetWare v4.x servers.

This facility is useful when implementing FDRSOS during:

- A backup when you wish to assure that the drive(s) are not being accessed and you wish to guarantee the integrity of the data
- A restore when you wish to assure that no users are going to be affected during the restore process
- A restore when you are restoring an older version of an entire drive (or drives) or when you are moving the backed up contents of a drive to a different drive and wish the operating system to be notified without having to shut down the operating system

FDRSOS provides an NLM which allows a mount or dismount without user intervention (regardless of open files): NWMOUNT.NLM and a workstation program NWSERVER which allows you to run server functions from a workstation (available for Windows NT and OS/2 at this time and other operating systems in the future).

### Installation

The diskette or CD-ROM provided with FDRSOS has only a few files in directories appropriate to the operating system on which they will execute. Copy these programs as needed. Here are the programs (the names shown are relative to the root directory of the diskette or the SOS\ directory of the CD):

| File Name | Description |
|---|---|
| GENDATA.EXE | A test data generation utility |
| NOVELL\NLM\NWMOUNT.NLM | utility to mount/dismount server volumes without operator prompts |
| NOVELL\NLM\VOLLIB.NLM | a Novell-supplied library required by NWMOUNT |
| NOVELL\NT\NWSERVER.EXE | Windows NT program to execute NLMs and HCF files |
| NOVELL\OS232\NWSERVER.EXE | OS/2 program to execute NLMs and HCF files |

## APPENDIX B. (CONTINUED...)

### NWMOUNT.EXE

NWMOUNT is a Netware program which allows you to mount or dismount a Netware volume without any user prompts. It can be used to automate the process of unmounting and remounting volumes during SOS backups and restores.

***WARNING: This is a very dangerous command and must be used with caution. The standard Netware command DISMOUNT will prompt you if there are open files on the volume and give you the option of bypassing the dismount. NWMOUNT will dismount the volume despite any open files.***

NWMOUNT.NLM and VOLLIB.NLM must be installed in the server's SYS:SYSTEM directory or in some directory in the SEARCH path. NWMOUNT can be executed from the Netware console or remotely using the NWSERVER utility as described later.

The command format is:

LOAD NWMOUNT <volume> [/Mount] [/Dismount]

Where:

- **<volume>** is the name of the Netware server volume to be mounted or dismounted. Do not include the trailing colon.
- **/Mount** requests that the specified volume be mounted. Be sure that the SOS backup or restore has completed before performing this function
- **/Dismount** requests that the specified volume be dismounted. As stated above, this can be dangerous if not done carefully.

For example, if you wish to dismount the VOL1 volume from the server console, specify:

LOAD NWMOUNT VOL1 /Dismount

If you wish to remount the VOL1 volume, specify:

LOAD NWMOUNT VOL1 /Mount

**Note: if you have dismounted the SYS volume, you cannot run NWMOUNT from the console to remount it since the NWMOUNT.NLM file is not available (it is on the SYS volume). However, you can use NWSERVER (described later) to remount the SYS volume from a workstation.**

## APPENDIX B. (CONTINUED...)

### B.2. NWMOUNT.EXE (continued)

NWMOUNT creates a separate window in which to display messages. If there is an error, the window remains displayed until a key is pressed, to give you an opportunity to read the messages. All significant messages are also written to the system console (for console logging purposes).

There are other advanced switches including:

- **/List** allows you to list all the volumes on the server (whether mounted or not, whether valid or not) with a significant amount of detail.
  Do not specify <volume> with this option.
- **/One** allows you to list volume details for just one volume (specified by <volume>)
- **/F<file name>** writes all output to the specified file name instead of the screen. The file name should be specified in Netware format immediately after the /F option (for example: /FSYS:NWOUNT.OUT). This option can be combined with any other option.
- **/R<new volume name>** allows you to rename <volume> to <new volume name>. Since this option does not interact with directory services, and since there are problems renaming a volume which has a duplicate volume name, we recommend using this option only after consultation with Innovation Technical Support.

## APPENDIX B. (CONTINUED...)

### B.3. NWSERVER.EXE

NWSERVER.EXE is a Windows NT or OS/2 program which allows you to perform a number of Netware server related functions from a workstation. Its form is:

NWSERVER <server> <function> ...parameters... [/t]

Where:

- **<server>** Is the NetWare server name that you wish to access. This must always be the first parameter and is required.
- **<function>** > is one of the function types below. Subsequent parameters are specific to the function type.
- **[/t]** is an optional parameter that you can specify after the last function parameter to enable tracing.

The functions supported are: ·

- **NCF** Specify this function if you wish to run an NCF batch file on the server. The single required parameter is the fully qualified path name of the NCF file that you wish to run. For example:

  NWSERVER SERVER1 NCF SYS:SYSTEM\TEST.NCF

- **NLM** Specify this function if you wish to load an NLM on the server. The single required parameter is the fully qualified path name of the NLM that you wish to load. Note that you can not pass parameters to the NLM. For example:

  NWSERVER SERVER1 NLM SYS:SYSTEM\USNDS.NLM

- **UNLOAD** Specify this function if you wish to unload a pre-loaded NLM on the server. The single required parameter is the name of the NLM that you wish to unload (this parameter is not usually fully qualified). For example:

  NWSERVER SERVER1 UNLOAD USNDS.NLM

- **MOUNT** Specify this function if you wish to mount a specified volume. The single required parameter is the volume name (without the trailing colon). This function is particularly useful for automating the mounting of the SYS volume when you used NWMOUNT to dismount it. For example:

  NWSERVER SERVER1 MOUNT SYS

- **DISMOUNT** Specify this function if you wish to dismount a specified volume. The single required parameter is the volume name (without the trailing colon). Note that this function is not terribly useful, as if there are ANY open files on the volume, the console will display a list of these files and wait for user intervention to acknowledge the breaking of the connection to these files. For example:

  NWSERVER SERVER1 DISMOUNT VOL1

Note that all NWSERVER functions require that you be logged in with console operator rights.

## APPENDIX B. (CONTINUED...)

### B.2. NWMOUNT.EXE (continued)

NWSERVER returns a zero program return code if the function could be initiated. For MOUNT, DISMOUNT and UNLOAD, this indicates whether the actual function was successful. For NCF, and NLM this indicates whether the function was started, but not the final return code.

## APPENDIX B. (CONTINUED...)

**Automation**

To automate the process of dismounting, a volume for backup most users will create two batch files on their workstation (one for mounting and one for dismounting), and one NCF batch file on their server (for dismounting). For example, through host automation which can run an application on a workstation (FDR/UPSTREAM is such a facility) you wish to run a job which will dismount a volume before beginning an EMC backup. Create a batch file on the workstation similar to the following:

**Sample batch file to dismount a volume**

```
Echo This batch file DISMOUNTs VOL1 on SERVER1
Echo It should only be called from the host.
NWSERVER SERVER1 NCF SYS:SYSTEM\VOL1DOWN.NCF
```

Note that your host facility should check the program return code from NWSERVER to see if the NCF file was correctly executed.

You would then write an NCF file on the server which would unmount the volume:

**Sample VOL1DOWN.NCF**

```
NWMOUNT VOL1 /Dismount
```

When the SOS backup has completed, run a batch file to remount the dismounted volume:

**Sample batch file to remount a volume**

```
Echo This batch file remounts VOL1 on SERVER1
NWSERVER SERVER1 MOUNT VOL1
```

Again, you will want to check the program return code to verify that the mount was successful; if not you will want to take manual action to examine the problem.

## APPENDIX B. (CONTINUED...)

### GENDATA.EXE

GENDATA is a bound program (DOS and OS/2) which allows you to quickly create a very large file (up to 4GB) and verify its integrity. It can be used generate test data to be backed up by FDRSOS, and to verify its contents after a FDRSOS restore. It's form is:

GENDATA <file name> <size in bytes> [/f<fill value>] [/r]

Where:

- **<file name>** is the fully qualified name of the file to be created or verified.
- **<size in bytes>** is the size of the file to be created or verified (for a verify, specify the same value used to create the file).
- **/f[fill value]** iis an optional parameter that allows you to be able to specify a single data byte to repeatedly place in the generated file. It can have values from 0 to 255. If you do not specify the /f option, GENDATA will create the file with 4-byte numbers indicating the long word position. GENDATA will run faster with the /f option than without it. For a verify, specify the same value used to create the file.
- **/r** causes GENDATA to verify the expected contents of a file. This option is generally used after a restore to verify that the data was replaced correctly.

For example, to create a 20,000,000 byte file name C:\TEST.DAT, specify:

GENDATA C:\TEST.DAT 20000000

To create a 1,000,000,000 byte file containing all zeros quickly specify:

GENDATA C:\TEST2.DAT 1000000000 /f0

To verify that the last file was restored correctly:

GENDATA C:\TEST2.DAT 1000000000 /f0 /r

THIS PAGE WAS INTENTIONALLY LEFT BLANK

## C. FDRSOS - OS/2

**Introduction**

FDRSOS includes a facility where an OS/2 attached EMC drive (one or more) can be locked throughout an FDRSOS backup or restore. This facility also assures that if the operating system definition of the drive is changed because you have restored a different drive to it, the operating system information about the drive is refreshed.

This facility is useful when implementing FDRSOS during:

- A backup when you wish to assure that the drive(s) are not being accessed and you wish to guarantee the integrity of the data
- A restore when you wish to assure that no users are going to be affected during the restore process.
- A restore when you are restoring an older version of an entire drive (or drives) or when you are moving the backed up contents of a drive to a different drive and wish the operating system to be notified without having to shut down the operating system.

The drive can remain locked until:

- A user presses a key.
- A predefined timeout occurs.
- Another program is run that tells the first program to unlock the drive.
- Note that when restoring a drive to a different location, the drives must be of the same size.

There are two programs in this facility: UNMOUNT and REMOUNT.

**Note: During a restore, UNMOUNT must be running and the drive must be locked to assure that the operating system definitions for the drive are reset. If the program is abnormally closed, the drive will remain locked and the system must be rebooted to make it accessable.**

**Installation**

The diskette or CD-ROM provided with FDRSOS has only a few files on it. Create a directory on your OS/2 system and copy the OS/2 files into it. Here are the programs (the names shown are relative to the root directory of the diskette or the SOS\ directory of the CD:

| File Name | Description |
|---|---|
| GENDATA.EXE | test data generation utility. |
| OS2\REMOUNT.EXE | utility to unlock a previously locked set of volumes. |
| OS2\UNMOUNT.EXE | utility to lock and dismount one or more volumes. |
| OS2\UNMOUNT.EXE | internal program called by UNMOUNT. |

## APPENDIX C. (CONTINUED...)

### UNMOUNT.EXE

UNMOUNT.EXE is a 32-bit OS/2 full-screen, windowable program that can lock one or more OS/2 drives and force a refresh of the OS/2 drive information when it is unlocked. UNMOUNT is a necessity when you are restoring an OS/2 volume so that OS/2 will recognize the structure of the restored data. UNMOUNT can also be used during backups to insure that no one can update the volume during the backup.

Unless you specify the /p option, UNMOUNT must be left running during the backup or restore to assume the the drive remains locked. During testing you should verify that the drive is locked during SOS operations by attempting to access the drive from OS/2 (such as a DIR command) and you should get a system pop-up message indicating the drive is locked.

UNMOUNT will fail if ANY programs are accessing the drive when it is executed. You must insure that all users and programs have closed all open files on the drive before running UNMOUNT to lock it. If this or any other error occurs, the program will write an error to the screen and set a non-zero return code.

Run this program from a OS/2 full-screen or window. The command format is:

UNMOUNT <drive letter:> ... [/n<name>] [/k] [/w<time>] [/p]

Where:

- **<drive letter:>** is a list of one or more drives to be locked, separated by spaces. For example C: or E: F: G:. You must specify at least one drive letter.
- **/n<name>** is an optional parameter which allows you to specify the semaphore name used when running REMOUNT. You need to specify this parameter only if you will be running REMOUNT and you wish to run more than one copy of UNMOUNT.EXE concurrently. For example, /nREMOUNT2.
- **/k** is an optional parameter that locks the drive until the user presses a key.
- **/w<time>** is an optional parameter that locks the drive for a given number of seconds. For example, for 5 minutes, specify **/w300**.
- **/p** is an optional parameter that causes UNMOUNT to terminate when the drive is locked. The drive will remain locked until the REMOUNT program (see below) is run. The /p option is particularly useful since it returns an immediately return code indicating if the drives were successfully locked.
- **/f** is an optional parameter which instructs UNMOUNT to open the drives specified, flush all buffers to disk and return without locking the drive. This may be useful during backups when you cannot afford to lock the drive but wish to have the most current information possible backed up.

## APPENDIX C. (CONTINUED...)

### UNMOUNT.EXE (CONTINUED)

UNMOUNT returns a code of 0 if all drives were successfully locked and 1 if any errors occurred.

Unless you specify the /f, /k or /w operands, the drive will remain locked until the REMOUNT program is executed.

If you wish to lock drives D, E, and F until REMOUNT is run, specify:

```
UNMOUNT D: E: F:
```

If you wish to lock drive D until you press a key, specify:

```
UNMOUNT D: /k
```

If you wish to lock D and E but unlock them with REMOUNT at separate times (see below for REMOUNT examples), specify:

```
UNMOUNT D: /nDDRIVE
UNMOUNT E: /nEDRIVE
```

In a batch file, the code to unmount the C drive, check the return code and return might look like:

```
UNMOUNT C: /p
if ERRORLEVEL 1 goto ERROR
Echo Successful lock – can begin SOS backup
goto SOS
:ERROR
Echo Unsuccessful lock – perform the SOS backup later
goto SOSLATER
```

### REMOUNT.EXE

REMOUNT.EXE is a 32-bit OS/2 program which will unlock an OS/2 drive and refresh the operating system's view of the contents of the drive. If you have executed UNMOUNT without the /f, /k and /w operands, you must execute REMOUNT to terminate UNMOUNT and activate the drive. Unless you used the /p operand on the UNMOUNT statement, you must run REMOUNT in a separate MS-DOS session from UNMOUNT.

The command format is:

```
REMOUNT [/n<name>]
```

Where:

- **/n<name>** is an optional parameter. If the UNMOUNT program was executed with the /n option, the same value must be specified on REMOUNT to unlock those drives. If omitted, REMOUNT will unlock all drives locked by an UNMOUNT with no /n parameter.

REMOUNT returns a code of 0 if all drives were successfully unlocked and 1 if any errors occurred.

In the first UNMOUNT example above, where drives D, E and F were locked, run this to unlock those drives:

```
REMOUNT
```

In the second UNMOUNT example above, where drives D and E were locked separately, unlock those drives with:

```
REMOUNT /nDDRIVE
REMOUNT /nEDRIVE
```

## APPENDIX C. (CONTINUED...)

### C.4. GENDATA.EXE

GENDATA is a bound program (DOS and OS/2) which allows you to quickly create a very large file (up to 4GB) and verify its integrity. It can be used generate test data to be backed up by FDRSOS, and to verify its contents after a FDRSOS restore. It's form is:

GENDATA <file name> <size in bytes> [/f<fill value>] [/r]

Where:

- **<file name>** is the fully qualified name of the file to be created or verified
- **<size in bytes>** is the size of the file to be created or verified (for a verify, specify the same value used to create the file).
- **/f<fill value>** is an optional parameter that allows you to be able to specify a single data byte to repeatedly place in the generated file. It can have values from 0 to 255. If you do not specify the /f option, GENDATA will create the file with 4-byte numbers indicating the long word position. GENDATA will run faster with the /f option than without it. For a verify, specify the same value used to create the file.
- **/r** causes GENDATA to verify the expected contents of a file. This option is generally used after a restore to verify that the data was replaced correctly.

For example, to create a 20,000,000 byte file name C:\TEST.DAT, specify:

GENDATA C:\TEST.DAT 20000000

To create a 1,000,000,000 byte file containing all zeros quickly specify:

GENDATA C:\TEST2.DAT 1000000000 /f0

To verify that the last file was restored correctly:

GENDATA C:\TEST2.DAT 1000000000 /f0 /r

## D. FDRSOS - AIX

To use FDRSOS to back up or restore AIX attached EMC drives there are recommended procedures which help to guarantee that the data is complete and correct and the the operating system is properly notified. Recommended procedures are provided for:

- Backup.
- Restoring disks to their original locations.
- Restoring one or more physical disks to different locations and scratching the original disks.
- Restoring a single volume volume group as a copy on the same system.
- Restoring a multiple volume volume group as a copy on the same system.

Note that coordination for the sequencing of host and workstation/server automated procedures is provided by FDR/UPSTREAM. FDR/UPSTREAM's USTBATCH host facility allows you to integrate workstation/server batch jobs into your host scheduling facilities. And FDR/UPSTREAM's host job execution facility allows host batch jobs to be executed from workstation/server automation. This allows you to coordinate the workstation/server procedures below with the execution of FDRSOS on the host. See chapter 8 (USTBATCH) in the FDR/UPSTREAM MVS manual and chapter 23 (Advanced FDR/UPSTREAM) in the FDR/UPSTREAM Workstation/Server manual for more information.

### Recommended Backup Procedure

- Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode. You can use smit **umountfs** or the **umount** command to unmount file systems.
- If the file systems can not be dismounted, you should insure that all cached I/O has been flushed out to the disk. Also, you should insure that no one is writing to the file system during the backup.
- If the physical disk(s) are in a volume group, deactivate the volume group. You can use smit **varyoffvg** or the **varyoffvg** command to deactivate a volume group. Again, if you can not deactivate the volume group, flush all I/Os to disk and verify that no users are writing to the file system.
- Run the SOS backups.
- Activate the volume group if the physical disk(s) are in a volume group. You can use smit **varyonvg** or the **varyonvg** command to activate a volume group.
- Remount the file systems. You can use smit **mountfs** , smit **mountg** (if the file systems have been assigned to a group), or the **mount** command to mount a file system.
- Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

## APPENDIX D. (CONTINUED...)

**Restoring one or more physical disks to original locations**

- Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode that are to be used for the restore. You can use smit **umountfs** or the **umount** command to unmount file systems.
- If the physical disk(s) are in a volume group, deactivate the volume group. You can use smit **varyoffvg** or the **varyoffvg** command to deactivate a volume group.
- Run the SOS restore. If you are restoring a multiple volume volume group, you MUST restore all of the volumes in the volume group.
- Activate the volume group if the physical disk(s) are in a volume group. You can use smit **varyonvg** or the **varyonvg** command to activate a volume group.
- Remount the file systems. You can use smit **mountfs** , smit **mountg** (if the file systems have been assigned to a group), or the mount command to mount a file system.
- Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

## APPENDIX D. (CONTINUED...)

**Restoring one or more physical disks to different locations and scratching the original disks.**

- Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode. You can use smit **umountfs** or the **umount** command to unmount file systems.
- If the physical disk(s) are in a volume group, deactivate the volume group. You can use smit **varyoffvg** or the **varyoffvg** command to deactivate a volume group.
- If the physical disk(s) are in a volume group, export the volume group. You can use smit **exportvg** or the **exportvg** command to export a volume group.
- Clear the physical volume id's (PVID's) from both the original and target physical disk(s) BEFORE running the restore. This is VERY important. AIX will not tolerate duplicate physical volume id's. To clear a PVID, use the following form of the chdev command:

  chdev -l hdisk# -a pv=clear

  where hdisk# represents the name of the physical volume such as hdisk5.
- Run the SOS restore. If you are restoring a multiple volume volume group, you MUST restore all of the volumes in the volume group.
- If the physical volume(s) are in a volume group, you must assign the physical volume id's of the restored physical volume(s). Note that physical volume id's will be the physical volume id's of the original volumes. Use smit chgdsk for EACH target volume. Make sure you change ***ASSIGN physical volume identifier*** to **yes**. AIX will use the physical volume identifier found on the restored volume.
- If the physical volume(s) are in a volume group, import the volume group. You can use smit **importvg** or the **importvg** command to import a volume group.
- Remount the file systems. You can use smit **mountfs** , smit **mountg** (if the file systems have been assigned to a group), or the **mount** command to mount a file system.
- Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

## APPENDIX D. (CONTINUED...)

**Restoring a single volume volume group as a copy on the same system.**

- Ensure that target volume has been assigned a physical volume id BEFORE running the restore. Use smit **chgdsk** to display the physical volume id of the target volume. If it already has one, then cancel out of this smit screen. If not, change ***ASSIGN physical volume identifier*** to yes and perform the change.
- Run the SOS restore using the VOLRESET=NO option. SOS will preserve the physical volume id. To allow you mount the file systems on the copy at the same time as the originals, SOS will change the mount points of all file systems during the restore by appending _sos to the original mount point. For example, if the original volume group had a file system with a mount point of /myfilesys, SOS will change it to /myfilesys_sos. This change will be picked up by AIX when the volume group is imported.

  Import the volume group. You can use smit **importvg** or the **importvg** command to import a volume group. Note that you can not use the name of the original volume group. You must assign a new name.
- You can now mount the filesystems on the new volume group. Remember that the mount points will be have _sos appended to the original mount point.

**Restoring a multiple volume volume group as a copy on the same system.**

- Clear the physical volume id's (PVID's) from target physical disk(s) BEFORE running the restore. To clear a PVID, use the following form of the chdev command:

  chdev -l hdisk# -a pv=clear

  where hdisk# represents the name of the physical volume such as hdisk5.
- Run the SOS restore using the VOLRESET=NO option. You MUST restore all of the volumes in the volume group. SOS will assign new physical volume id's to all of the volumes in the volume group. To allow you to mount the file systems on the copy at the same time as the originals, SOS will change the mount points of all file systems during the restore by appending _sos to the original mount point.

  For example, if the original volume group had a file system with a mount point of /myfilesys, SOS will change it to /myfilesys_sos. This change will be picked up by AIX when the volume group is imported.
- You must assign the physical volume id's of the restored physical volume(s). Note that physical volume id's will be the physical volume id's set by SOS. Use smit **chgdsk** for EACH target volume. Make sure you change ***ASSIGN physical volume identifier*** to yes . AIX will use the physical volume identifier found on the restored volumes.
- Import the volume group. You can use smit **importvg** or the **importvg** command to import a volume group. Note that you can not use the name of the original volume group. You must assign a new name.
- You can now mount the filesystems on the new volume group. Remember that the mount points will have _sos appended to the original mount point.

## E. FDRSOS - HP-UX Procedures

To use FDRSOS to back up or restore HP-UX attached EMC drives, there are recommended procedures which help to guarantee that the data is complete, correct and the operating system is properly notified. Recommended procedures are provided for:

- Backup.
- Restoring disks to their original locations.
- Restoring one of more physical disks to different locations or as a copy on the same system.

Note that coordination for the sequencing of host and workstation/server automated procedures is provided by FDR/UPSTREAM. FDR/UPSTREAM's USTBATCH host facility allows you to integrate workstation/server batch jobs into your host scheduling facilities. And FDR/UPSTREAM's host job execution facility allows host batch jobs to be executed from workstation/server automation. This allows you to coordinate the workstation/server procedures below with the execution of FDRSOS on the host. See chapter 8 (USTBATCH) in the FDR/UPSTREAM MVS manual and chapter 23 (Advanced FDR/UPSTREAM) in the FDR/UPSTREAM Workstation/Server manual for more information.

### Recommended Backup Procedure

- Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volumes(s) on the physical disk(s) in RAW mode. You can use SAM utilities or **umount** command to unmount the file systems. If the file systems cannot be dismounted, you should insure that all cached I/O has been flushed out to the disk. Also, you should insure that no one is writing to the file system during the backup.
- If the physical disk(s) are in a volume group, deactivate the volume group. You can use SAM utilities or the **vgchange** command to deactivate a volume group. Again, if you cannot deactivate the volume group, flush all I/Os to disk and verify that no users are writing to the file system.
- Run the SOS backups.
- Activate the volume group if the physical disk(s) are in a volume group. You can use SAM utilities or the **vgchange** command to activate a volume group.
- Perform 'file check' on the unmounted file systems to insure integrity. For example:

  ```
  fsck -F vxfs -o full ......
  fsck .............
  ```

- Remount the file systems. You can use SAM utilities or the **mount** command to mount a file system.
- Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

## APPENDIX E. (CONTINUED...)

### Restoring one or more physical disks to original locations

- Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode that are to be used for the restore. You can use SAM utilities or the **umount** command to unmount file systems.
- If the physical disk(s) are in a volume group, deactivate the volume group. You can use SAM utilities or the **vgchange** command to deactivate the volume group.
- Run the SOS restore. If you are restoring a multiple volume 'volume group', you MUST restore all of the volumes in the volume group.
- Activate the volume group if the physical disk(s) are in a volume group. You can use SAM utilities or the **vgchange** command to activate a volume group.
- Perform 'file check' on the unmounted file systems to insure integrity. For example:

  fsck -F vxfs -o full .....
  fsck .....

- Remount the file systems. You can use SAM utilities or the mount command to mount a file system.
- Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

## APPENDIX E. (CONTINUED...)

### Restoring one or more physical disks to different locations

- Unmount ALL active file systems in the volume group and/or shutdown any applications using the physical disk(s) or logical volume(s) on the physical disk(s) in RAW mode that are to be used for the restore. You can use SAM utilities or the **umount** command to unmount file systems.
- If the physical disk(s) are in a volume group, deactivate the volume group. You can use SAM utilities or the **vgchange** command to deactivate the volume group.
- If the physical disk(s) are in a volume group, export the volume group. You can use SAM utilities or the **vgexport** command to remove the volume group. The use of the **vgexport mapfile** parameter is recommended to retain the logical volume names for the **vgimport** command. Otherwise the system will created default logical volume names of "lvol1", "lvol", etc. For example:

  vgexport –v –m <mapfile> <vg>

  where mapfile is the name of the output map file and vg is the volume group name.
- Run the SOS restore. If you are restoring a multiple volume 'volume group', you MUST restore all of the volumes in the volume group.
- Recreate the target volume group directory and target group file using the following commands

  cd /
  mkdir dev/<vg>
  mknod /dev/<vg> /group c 64 0xNN0000

  where vg is volume group name and NN is unique 'minor number' relating to volume group number.

•Import volume group using SAM or the following command:
vgimport –v –m <mapfile> /dev/<vg> /dev/dsk/<cxtydz1> /dev/dsk/<cxtydz2> ..
where mapfile is the input map file name, vg is the volume group, and cxtydz1, etc. represents disk volumes in the volume group

- Activate the volume group if the physical disk(s) are in a volume group. You can use SAM utilities or the **vgchange** command to activate a volume group.
- Perform 'file check' on the unmounted file systems to insure integrity. For example:

  fsck -F vxfs -o full .....
  fsck .....

- Remount the file systems. You can use SAM utilities or the mount command to mount a file system.
- Restart the applications (if any) that were using the physical disk(s) or logical volume(s) in RAW mode.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

# F. FDRSOS - Solaris

**Warning**

***WARNING: ALL BACKUPS OF SUN VOLUME MANAGER DISKS SHOULD BE DONE WITH THE FILE SYSTEMS UNMOUNTED!***

For backups, this is required to "harden" the data and insure that all data is written to the disk, as described in Section 220.11 in the FDRSOS manual. If a backup is done with a file system mounted then a restore of that backup may result in downlevel data or other errors.

**Restoring to an alternate volume**

If you wish to restore backups of SUN Volume Manager disks to spare disk volumes (as described in Section 220.14 of the FDRSOS manual), use this procedure:

1. If the spare volumes are currently known to the SUN volume manager, usually because you have previously used them as the target of a FDRSOS restore, you must unmount the file systems on those volumes and deport the volumes before doing the restore.
2. Do the FDRSOS restore, to the spare disk volumes, with the VOLRESET=NO parameter. FDRSOS will change the internal volume name and volume group ID to make them unique. It will also make the volume group name unique by overlaying the first character of the volume group name with an underscore; for example, if the group name was "testgroup", VOLRESET=NO will change it to "_estgroup".

   If the underscore is not acceptable or does not result in a unique group name, you can specify the FDRSOS parameter "SUNGROUPID=xxx" on the MOUNT statement for the output volume. "xxx" is 1 to 3 characters; these characters will be made lower case (even if entered in upper case) and used to overlay the beginning of the group name. For example, SUNGROUPID=TMP will change "testgroup" to "tmptgroup". You must use the same SUNGROUPID value while restoring all volumes in the group.

   If you specify PRINT=STATUS on the RESTORE statement, FDRSOS will display the group and volume names before and after the restore; the after display will show the modifications described above.

   If the spare volumes were not previously managed by the SUN volume manager, you must refresh the volume manager so that it will recognize the new volumes. From the root user, issue:

   vxdctl enable
3. Import the volume group using the new group name set by FDRSOS (step 2 above). You can use the GUI interface of the volume manager (vxva), the menu interface (vxdiskadm), or the the vxdg command. For the latter, issue from the root user:

   vxdg import _estgroup
4. Start the volumes in the group you just imported. You can either start specific volumes or all the volumes. You can use the GUI or menu interfaces of the volume manager, or the vxvol command. For the latter, issue from the root user:

   vxvol -g _estgroup volume_name (for a specific volume)
   vxvol -g _estgroup startall (for all volumes)
5. Mount the filesystem(s) on the started volume(s). You can use either the GUI interface of the volume manager, or the **mount** command. To use the **mount** command, you should refer to /etc/vfstab for the options used to mount the original file system. You will need to create a new mount point and use the new device name and new mount point to mount the file system. Volume manager device names for a given volume will be:

   /Dev/vx/dsk/diskgroup/volumename
   /dev/vx/rdsk/diskgroup/volumename

   To mount a SUN (Veritas) filesystem on a volume named "vol01" which is in the disk group restored as _estgroup at mount point /test, issue this command:

   mount -F vxfs /dev/vx/dsk/_estgroup/vol01 /test

   At this point the files on the restored volumes should be accessible.

## APPENDIX F. (CONTINUED...)

### WARNING ON USE OF RESERVE=YES

We have found that the use of RESERVE=YES on DUMP statements may cause problems if the Open System using the volumes you are backing up is sensitive to a long term RESERVE on the volume. If the volume may still be in use on the Open System during the FDRSOS backup, you may need to omit RESERVE=YES (or specify RESERVE=NO). This is most likely to be a problem if the Open System volume involved is a system volume to the Open System (e.g., the boot volume).

In particular we have found that Sun Solaris systems will fail if their system volumes are reserved for long periods of time.

However, this may also be true for other Open System or for application-type volumes (such as data bases).

Certainly it will be true that doing a long-term RESERVE on a system or application volume will probably cause that system or application to hang until the backup is complete.

The default on a DUMP is RESERVE=NO.

## APPENDIX F. (CONTINUED...)

### F.1 INITIALIZING LOCALBACKUP ON SUN SPARC

In order to use FDRSOS local backup volumes with FDR/UPSTREAM/SOS on a SUN SOLARIS system running on a SUN SPARC platform, a special procedure must be followed. This does not apply to SOLARIS running on an Intel-type platform.

1. the disk is brand new, never before used, you should erase the disk completely with the FDRSOS ERASE TYPE=FULL function (Section 210.05).
2. from the SUN system, you must format the disk volume and specify a volume name of "FDRSOS". To do so:
   - Use the "format" command to format the disk. "format" will list the disks available to the system; select the disk to be used for local backups.
   - Use the "volname" subcommand to assign a volume name of "FDRSOS" to the volume.
   - Use the "partition" subcommand to delete all partitions on the volume **except the "backup" partition** which must be partition 2. For every partition whose "tag" is other than "unassigned", you must modify that partition to mark it "unassigned" and change its starting and ending cylinders to 0 (zero).
   - When done you can use the "verify" subcommand to verify that the volume is correctly initialized. The output will look something like this:

```
format> verify

Primary label contents:

Volume name = <  FDRSOS>
ascii name  = <EMC-SYMMETRIX-5264 cyl 1103 alt 2 hd 15 sec 64>
pcyl        = 1105
ncyl        = 1103
acyl        =    2
nhead       =   15
nsect       =   64
Part      Tag    Flag     Cylinders         Size            Blocks
  0 unassigned    wm       0               0          (0/0/0)          0
  1 unassigned    wm       0               0          (0/0/0)          0
  2      backup   wu       0 - 1102      517.03MB    (1103/0/0) 1058880
  3 unassigned    wm       0               0          (0/0/0)          0
  4 unassigned    wm       0               0          (0/0/0)          0
  5 unassigned    wm       0               0          (0/0/0)          0
  6 unassigned    wm       0               0          (0/0/0)          0
  7 unassigned    wm       0               0          (0/0/0)          0
```

3. use the FDRSOS LOCALBACKUP function to make the formatted disk as a FDRSOS local backup (see Section 210.10).

***Warning: if you do not follow this procedure, in this order, the local backup disk will be usable from non-SUN systems but not from the SUN system. Disks initialized with this procedure will be usable on the SUN and most other Open Systems.***

THIS PAGE WAS INTENTIONALLY LEFT BLANK

## READER'S COMMENT FORM
## FDRSOS USERS MANUAL V5.3
## INNOVATION DATA PROCESSING

If you have comments on this manual, including:

- errors in the text or typographical errors
- clarity
- suggestions for improvement in the manual
- suggestions for improvement in the product
- any other comments

Please complete this form and fax it to Innovation at 973-890-7147 (in Europe you may fax it to your local Innovation office as shown on the front page of the manual).

You may also e-mail your comments to Innovation at support@fdrinnovation.com (be sure to identify the manual name in the message).

Your name: _____

Company name: _____

Mailing address: _____

_____

E-mail address: _____

Comments:_____

_____

_____

_____

_____

THIS PAGE WAS INTENTIONALLY LEFT BLANK