

**VELOCITY**  
**S O F T W A R E**

## CMS Shared File System Usage and Administration

Velocity Software Inc.  
196-D Castro Street  
Mountain View CA 94041  
650-964-8867

Velocity Software GmbH  
Max-Joseph-Str. 5  
D-68167 Mannheim  
Germany  
+49 (0)621 373844

Richard Smrcina  
Velocity Software, Inc  
VM Workshop 2014

Introduction to SFS  
Disk Allocation  
Filepool Configuration  
Accessing the pool  
Authorization  
Space management  
Backup and Recovery  
Byte File System  
Management Tools

Makes pools of disk space available

Access controls at file level

One writer/Multiple readers

Hierarchical directory structure

Each user is allocated space in the pool

Provides storage for Byte File System (bfs)

## Sample directory entry

'A' disk for pool definition files and optionally backup

Control minidisk maps the space in the filepool

Log disks help maintain the integrity of the filepool and provide a rollback mechanism

Catalog keeps track of files and directories and access control information for them

Data disk(s) provide storage for end user data

```
IDENTITY VMSYSVPS VELOCITY 96M 96M BG
INCLUDE IBMDFLT
BUILD ON VSIVM5 USING SUBCONFIG SFSVPS-1
IPL CMS
IUCV ALLOW
IUCV *IDENT RESANY GLOBAL
MACH XC
OPTION MAXCONN 200 APPLMON ACCT QUICKDSP SVMSTAT
SHARE ABS 3%
LINK MAINT 0193 0193 RR
*
SUBCONFIG SFSVPS-1
MDISK 0191 3390 3331 0030 VM5W01 MR
*Control
MDISK 0301 3390 3361 0020 VM5W01 WR
*Logging
MDISK 0302 3390 3381 0010 VM5W01 WR
MDISK 0303 3390 3391 0010 VM5W01 WR
*Catalog
MDISK 0304 3390 3401 0010 VM5W01 WR
*DATA
MDISK 0305 3390 3411 0450 VM5W01 MR
```

## Mapping minidisks into the filepool

### POOLDEF file

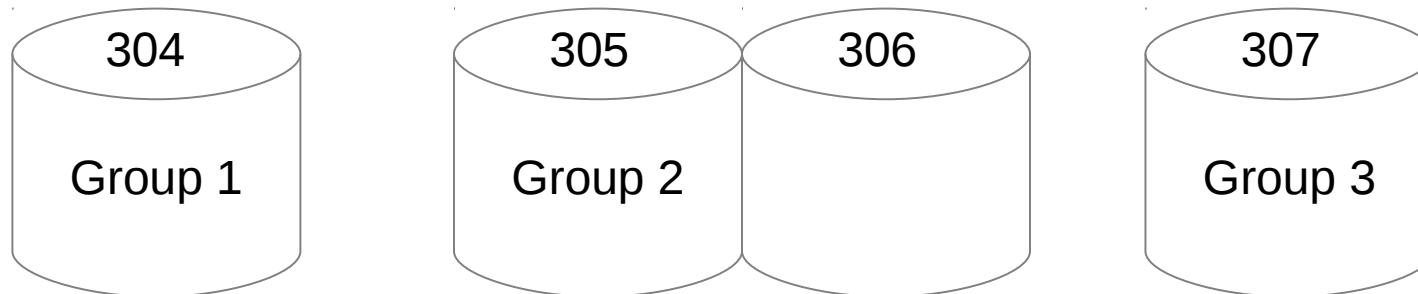
```
MAXUSERS=500
MAXDISKS=500
DDNAME=CONTROL VDEV=301
DDNAME=LOG1 VDEV=302
DDNAME=LOG2 VDEV=303
DDNAME=BACKUP DISK FN=CONTROL FT=BACKUP FM=A
DDNAME=MDK00001 VDEV=304 GROUP=1
DDNAME=MDK00002 VDEV=305 GROUP=2
```

```
MDISK 0191 3390 3331 0030 VM5W01 MR
*Control
MDISK 0301 3390 3361 0020 VM5W01 WR
*Logging
MDISK 0302 3390 3381 0010 VM5W01 WR
MDISK 0303 3390 3391 0010 VM5W01 WR
*Catalog
MDISK 0304 3390 3401 0010 VM5W01 WR
*DATA
MDISK 0305 3390 3411 0450 VM5W01 MR
```

## Grouping minidisks

### POOLDEF file

```
MAXUSERS=500
MAXDISKS=500
DDNAME=CONTROL  VDEV=301
DDNAME=LOG1     VDEV=302
DDNAME=LOG2     VDEV=303
DDNAME=BACKUP   DISK  FN=CONTROL  FT=BACKUP  FM=A
DDNAME=MDK00001 VDEV=304  GROUP=1
DDNAME=MDK00002 VDEV=305  GROUP=2
DDNAME=MDK00003 VDEV=306  GROUP=2
DDNAME=MDK00004 VDEV=307  GROUP=3
```



# Filepool Configuration

## Two files used to define/configure a filepool

### DMSPARMS file

```
ADMIN MAINT FTPSERVE ZVPS ZPRO  
BACKUP  
SAVESEGID CMSFILES  
FILEPOOLID VMSYSVPS  
USERS 200
```

### POOLDEF file

```
MAXUSERS=500  
MAXDISKS=500  
DDNAME=CONTROL VDEV=301  
DDNAME=LOG1 VDEV=302  
DDNAME=LOG2 VDEV=303  
DDNAME=BACKUP DISK FN=CONTROL FT=BACKUP FM=A  
DDNAME=MDK00001 VDEV=304 GROUP=1  
DDNAME=MDK00002 VDEV=305 GROUP=2
```

## DMSPARMS parameters

ADMIN – defines filepool administrators

- Also BFS super users

BACKUP – automatic backup of control data

- Kicks in when logs are 80% full

SAVESEGID – name of saved segment that contains filepool executable code

FILEPOOLID – name of THIS filepool

- Also the filename of the POOLDEF file

USERS – Estimate of the number of users



## POOLDEF parameters

MAXUSERS - number of filepool users authorized to create objects

- Files, directories, aliases
- Used to calculate size of catalog and control areas

MAXDISKS - number of minidisks ever to be used by this filepool server

## Generate the filepool

Log on to filepool server

Format the A-disk

Create POOLDEF, DMSPARMS and PROFILE EXEC

Access the 193 disk

Issue the command:

```
FILESERV GENERATE VMSYSVPS P00LDEF
```

## FILESERV GENERATE output

```
DMSWFV1117I FILESERV processing begun at 19:12:41 on 12 Jul 2011
DMSWFV1121I VMSYSVPS DMSPARMS A1 will be used for FILESERV processing
DMS4PD3400I Initializing begins for DDNAME = CONTROL
DMS4PD3400I Initializing ends for DDNAME = CONTROL
DMS4PD3400I Initializing begins for DDNAME = MDK00001
DMS4PD3400I Initializing ends for DDNAME = MDK00001
DMS4PD3400I Initializing begins for DDNAME = MDK00002
DMS4PD3400I Initializing ends for DDNAME = MDK00002
DMS4PD3400I Initializing begins for DDNAME = LOG1
DMS4PD3400I Initializing ends for DDNAME = LOG1
DMS4PD3400I Initializing begins for DDNAME = LOG2
DMS4PD3400I Initializing ends for DDNAME = LOG2
DMS5FD3032I File pool server has terminated
DMSWFV1120I File VMSYSVPS P00LDEF A1 created or replaced
DMSWFV1117I FILESERV processing ended at 19:13:02 on 12 Jul 2011
```

## Backup the control area

FILESERV BACKUP

## Create PROFILE EXEC

```
/* VMSYSVPS PROFILE EXEC */  
'CP SPOOL CON ZVPS START'  
'CP SET PF12 RETRIEVE'  
'ACCESS 193 B'  
'CP SET RUN ON'  
'CP SET EMSG ON'  
'FILESERV START'
```

## Execute PROFILE to start filepool server

```
DMSACP723I B (193) R/0  
DMSWFV1117I FILESERV processing begun at 20:00:53 on 12 Jul 2011  
DMSWFV1121I VMSYSVPS DMSPARMS A1 will be used for FILESERV processing  
DMSWFV1121I VMSYSVPS POOLDEF A1 will be used for FILESERV processing  
DMS5BB3045I Ready for operator communications
```

Users are enrolled in the filepool

- Lets them connect to the pool
- Allows access to directories and files

```
ENROLL USER ZVPS VMSYSVPS
```

Users can be given space in the filepool

```
ENROLL USER ZVPS VMSYSVPS (BLOCKS 20000
```

Or in a specific group

```
ENROLL USER ZVPS VMSYSVPS (BLOCKS 20000 GROUP 3
```

Hint: 100 cylinders is approximately 18,000 4K blocks

# Accessing the pool

## Graphic administration of the pool

**zPRO SFS Management - Shared File Subsystem Summary**

Display All records Copy CSV PDF Print Search:

Showing 1 to 23 of 23 entries First Previous 1 Next Last

Filepool Name	Status	Owning Userid/Node	Admin	Groups/ Minidisks	Catalog Blks Alloc	Catalog Blks Used	Catalog Blks Free	Pct Used	File Space Blks Alloc	File Space Blks Used	File Space Blks Free	Pct Used	Minidisk Log Blks	Pct Used	Minidisk Control Blks
SFSDEMO	Global	SFSDEMO	Yes	2/2	2,689	77	2,612	3%	8,983	2	8,981	0%	2,507	1%	6,551
SFSVM4	Global	SFSVM4	Yes	2/4	17,974	6,792	11,182	38%	521,431	457578	63,853	87%	3,586	59%	36,436
SFSZVPS4	Global	SFSZVPS4	Yes	2/4	8,983	2,127	6,856	24%	251,711	173064	78,647	68%	1,788	36%	14,570
COMSERV	Global	VSIVM5	Unknown												
ES4VM4	Global	ES4SERVE	Unavailable												
VSITIME	Global	VSIVM2	Unavailable												
LINUXVM	Global	LINUXVM	Notenrolled												
SFSVM5	Global	VSIVM5	Notenrolled												
SFSVM6	Global	VSIVM6	Notenrolled												
VMSYS	Local	VMSERVS	Notenrolled												
VSIPRB	Global	VSIVM1	Notenrolled												
ESAWEB	Global	ESAWEB	Notadmin												
SFSZVPS1	Global	VSIVM1	Notadmin												
VMSYSU	Local	VMSERVU	Notadmin												
VSIFTP	Global	VSIVM1	Notadmin												
VSIPRV	Global	VSIVM2	Notadmin												
BFSSVR	Global	VSIVM1	Local												
SFSVM1	Global	VSIVM1	Local												
SFSVM2	Global	VSIVM2	Local												
SFSZVPS	Global	VSIVM2	Local												
VSIDEV	Global	VSIVM2	Local												
VSIIDE	Global	VSIVM2	Local												
VSIWEB	Global	VSIVM1	Local												

**Select Option for filepool SFSZVPS4 or Cancel** ✕

Filepool User Detail

Filepool Administrators

Filepool Minidisk Detail

Enroll a User    Userid:

Storage group:     Initial Quota(in 4K blocks):

Enroll an Admin    Admin:

Cancel

## Graphic administration of the pool

**Select Option for filepool SFSZVPS4 or Cancel** ✕

---

---

---

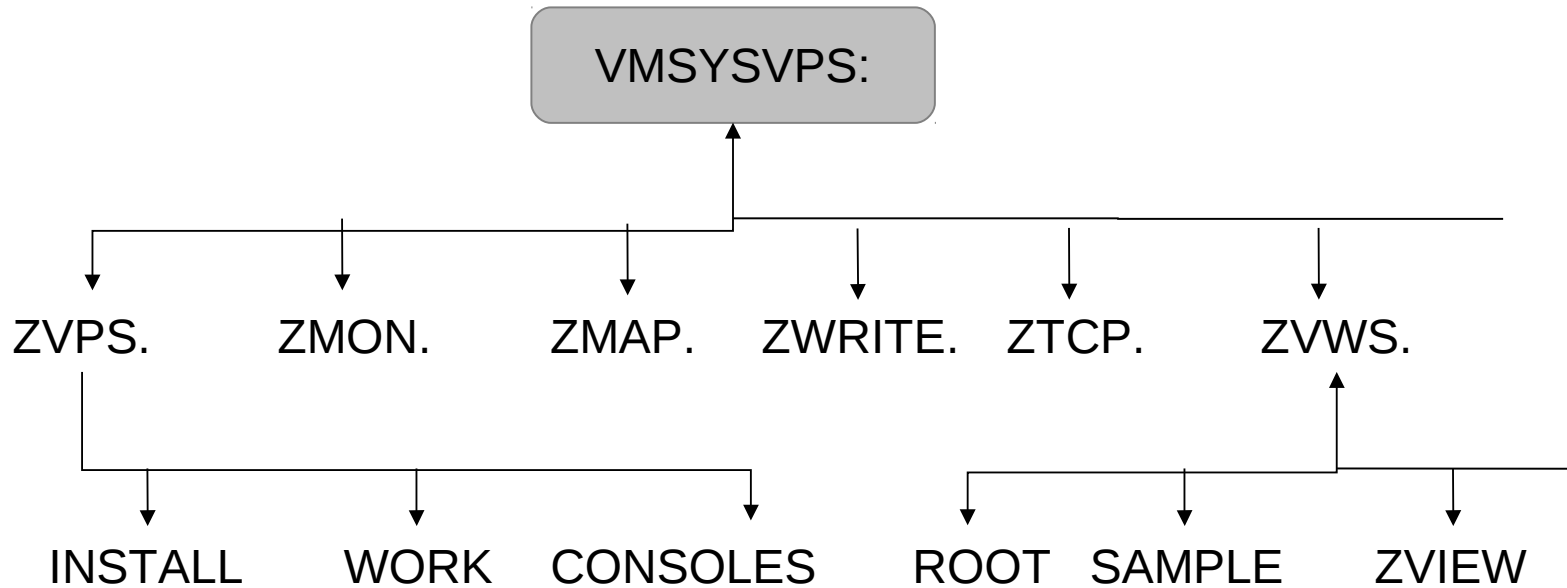
     Userid:

Storage group:       Initial Quota(in 4K blocks):

---

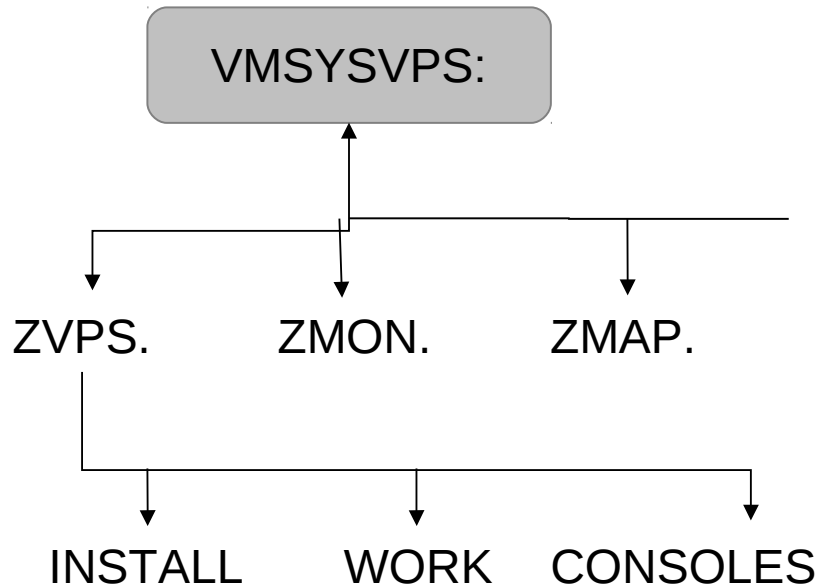
     Admin:

## Conceptual view





# Accessing the pool



Use CMS ACCESS command

First component is filepool name

Then top level directory and any subdirectories

ACC VMSYSVPS:ZVPS. B

ACC VMSYSVPS:ZVPS.CONSOLES D

## Set the default filepool

```
SET FILEPOOL VMSYSVPS
```

## Set top level directory as A-disk

```
IPL CMS PARM AUTOOCR FILEPOOL VMSYSVPS:
```

## Different pools can be accessed

```
ACC VMSYSVPS:ZVPS.CONSOLES B
```

```
ACC VMSYSU:RKSDEV. Q
```

## VMLINK command

```
VMLINK .DIR VMSYSVPS:ZVPS.CONSOLES
```

## Which directories are accessed?

### q accessed

Mode	Stat	Files	Vdev	Label/Directory
A	R/W	125	191	RKS191
S	R/0	699	190	MNT190
X	R/0	362	DIR	SFSZVPS4:ZVPS.CONFIG
Y/S	R/0	1097	19E	MNT19E
Z	R/0	692	DIR	VMSYSVPS:ZMON.CODE

### q search

RKS191	191	A	R/W	
MNT190	190	S	R/0	
-	DIR	X	R/0	SFSZVPS4:ZVPS.CONFIG
MNT19E	19E	Y/S	R/0	
-	DIR	Z	R/0	VMSYSVPS:ZMON.CODE

## Some commands allow directory ID specification

```
ERASE RACF DATA SFSVM6:RKSDEV.
```

```
FILELIST * * SFSVM6:RKSDEV.
```

```
RENAME RACF DATA SFSVM5:RKSDEV. RACF DATAOLD SFSVM5:RKSDEV.
```

See individual command syntax

## Display directory hierarchy with LISTDIR

```
LISTDIR
```

```
Fm Directory Name
```

```
A SFSVM5:RKSDEV.
```

```
LISTDIR VMSYSVPS:ZVPS.
```

```
Fm Directory Name
```

- VMSYSVPS:ZVPS.
- VMSYSVPS:ZVPS.CONFIG
- VMSYSVPS:ZVPS.CONSOLES
- VMSYSVPS:ZVPS.INSTALL
- VMSYSVPS:ZVPS.WORK

## Full screen display of directory hierarchy with DIRLIST

```
RKSDEV  DIRLIST  A0  V 319  Trunc=319  Size=5  Line=1  Col=1  Alt=0
```

```
Cmd  Fm  Directory Name
Z  VMSYSVPS:ZVPS.
X  VMSYSVPS:ZVPS.CONFIG
-  VMSYSVPS:ZVPS.CONSOLES
-  VMSYSVPS:ZVPS.INSTALL
-  VMSYSVPS:ZVPS.WORK
```

```
1= Help      2= Refresh  3= Quit    4= Sort(fm)  5= Sort(dir)  6= Auth
7= Backward  8= Forward  9=         10=          11= Filelist  12= Cursor
```

```
====>
```

```
X E D I T  1 File
```

Directory attributes are specified at creation

Two different attributes

- Directory Control (DIRCONTROL)

- File Control (FILECONTROL)

**DIRCONTROL** directories work like minidisks

- One writer, Multiple readers

- Authorization given at directory level

- Suited primarily for read activity

**FILECONTROL** directories much more flexible

- Multiple writers and readers

- Authorization given at file level

Access is granted to a directory

Four types of directory access

- READ
- NEWREAD
- WRITE
- NEWWRITE



## READ

- Allows access to a directory
- See the files and sub-directories

## NEWREAD

- Implies READ
- Allows a user to see new files or sub-directories

## WRITE

- Implies READ
- Allows a user to create new files

## NEWWRITE

- Implies WRITE
- Allows a user to write to new files

To give a user full access to a directory

```
GRANT AUTH VMSYSVPS:ZVPS. TO RKSDEV (NEWREAD NEWWRITE
```

Give a user read access to a directory

```
GRANT AUTH VMSYSVPS:ZVPS.CONSOLES TO ZWEB01 (NEWREAD
```

Access can also be granted to a file or files in a directory

Two types of file access

- READ, WRITE

READ

- See the contents of a file

WRITE

- Change the contents of a file
- Rename and erase a file
- Requires write access to the directory

## Grant read access to a file

```
GRANT AUTH PROFILE EXEC VMSYSVPS:ZVPS. TO RKSDEV (READ
```

## Grant all access to a file

```
GRANT AUTH INSTALL PROD4200 VMSYSVPS:ZVPS. TO RKSDEV (WRITE
```

## Grant access to all files in a directory

```
GRANT AUTH * * VMSYSVPS:ZVPS.CONSOLES TO ZWEB01 (READ
```

Enrolled users also have access to PUBLIC areas, directories or files granted to PUBLIC

```
GRANT AUTH VMSYSVPS:ZMON.CODE TO PUBLIC (NEWREAD
```

All users can be allowed to connect

```
ENROLL PUBLIC VMSYSVPS:
```

GRANT AUTHORITY command is used to allow users to access files and directories

- Can be issued by ADMIN or OWNER

GRANT is reversed with REVOKE AUTHORITY

ADMIN authority

- Users listed in ADMIN statement in POOLDEF
- Users listed in ENROLL ADMIN

ENROLL ADMIN ZVPS

## Storage group messages

DMS4KC3202W Storage group 2 is short on storage

DMS4DL3201E Storage group 2 is full

## DMS3202W is issued based on the GROUPTHRESH parameter

In the DMSPARMS file

Default is 90%





## Check minidisk and storage group space

### STORAGE GROUP MINIDISK TOTALS

Storage Group No.	4K Blocks In-Use	4K Blocks Free
1	618 - 35%	1172
2	71265 - 84%	13229

---

### LOG INFORMATION

First Log Virtual Address           0302  
Second Log Virtual Address        0303

1788   Number of Log Minidisk 4K Blocks  
41%    Percent(%) of Log Space Used  
95%    LUW Rollback/Suspend Threshold (% Log Space)  
80%    Backup Threshold (% Log Space)  
05/25/14   Date of Last Control Backup  
17:41:13   Time of Last Control Backup

---

### CONTROL MINIDISK INFORMATION

Control Minidisk Virtual Address   0301  
14570   Number of Control Minidisk 512 Blocks

---

## Check minidisk and storage group space

**Minidisk detail for filepool SFSZVPS4** ✕

STORAGE GROUP MINIDISK INFORMATION

Storage Group No.	Minidisk Number	4K Blocks In-Use	4K Blocks Free	Virtual Address
1	1	2124 - 24%	6859	0304
2	2	102856 - 95%	5022	0305
2	3	53914 - 100%	21	0306
2	4	16269 - 18%	73629	0307

=====

LOG INFORMATION

First Log Virtual Address           0302  
Second Log Virtual Address        0303

1788 Number of Log Minidisk 4K Blocks  
34% Percent(%) of Log Space Used  
95% LUW Rollback/Suspend Threshold (% Log Space)  
80% Backup Threshold (% Log Space)

06/12/14 Date of Last Control Backup  
07:39:03 Time of Last Control Backup

=====

CONTROL MINIDISK INFORMATION

Control Minidisk Virtual Address 0301  
14570 Number of Control Minidisk 512 Blocks

=====

Cancel

## Filespace messages

DMSCPY1141W User filespace threshold exceeded for file pool SFSVM5

DMSCPY1254E An attempt to commit will exceed the number of 4K blocks allowed for the user in file pool SFSVM5

## Check filespace allocation

### q limits for rksdev

Userid	Storage Group	4K Block Limit	4K Blocks Committed	Threshold
RKSDEV	2	70000	69923-99%	90%

### q limits all

Userid	Storage Group	4K Block Limit	4K Blocks Committed	Threshold
BARTON	2	0	0-00%	90%
DSADEV	2	0	0-00%	90%
OPERATOR	2	20000	1231-06%	90%
ZADMIN	2	180	3-01%	90%
ZMAP	2	1000000	1246-00%	90%
ZMON	2	3600	1562-43%	90%
ZSERVE	2	180	3-01%	90%
ZTCP	2	5400	302-05%	90%
ZVPS	2	200000	62547-31%	90%
ZVWS	2	5580	2531-45%	90%
ZWEBLOG	2	1800	1185-65%	90%
ZWEB01	2	180	3-01%	90%
ZWEB02	2	180	3-01%	90%
ZWEB03	2	180	3-01%	90%
ZWEB04	2	180	3-01%	90%
ZWEB05	2	180	3-01%	90%
ZWRITE	2	1000000	686-00%	90%

## Check filespace allocation

**User detail for filepool SFSZVPS4** ✕

Show 25 entries

Showing 1 to 25 of 62 entries Search:

<input type="checkbox"/> Sel	<input type="checkbox"/> Userid	<input type="checkbox"/> Storage Group	<input type="checkbox"/> 4K Blocks Limit	<input type="checkbox"/> 4K Blocks Used	<input type="checkbox"/> Percent Used	<input type="checkbox"/> Warning Threshold	<input type="checkbox"/> BFS	<input type="button" value="Select All"/>
<input type="checkbox"/>	ZWEB05	2	180	180	100%	90%		<input type="button" value="Reset"/>
<input type="checkbox"/>	ZVWS	2	5,580	4,616	82%	90%		<input type="button" value="Delete"/>
<input type="checkbox"/>	ZVPS	2	260,000	138,594	53%	90%		
<input type="checkbox"/>	ZPRO	2	6,300	3,221	51%	90%		
<input type="checkbox"/>	ZWEB01	2	360	177	49%	90%		
<input type="checkbox"/>	ZWEB11	2	180	75	41%	90%		
<input type="checkbox"/>	ZMON	2	5,000	1,808	36%	90%		
<input type="checkbox"/>	ZWEBLOG	2	60,000	16,675	27%	90%		
<input type="checkbox"/>	ZWEB03	2	180	48	26%	90%		
<input type="checkbox"/>	ZMAP	2	5,000	1,248	24%	90%		
<input type="checkbox"/>	ZALERT	2	1,000	220	22%	90%		
<input type="checkbox"/>	ZWEB08	2	180	29	16%	90%		
<input type="checkbox"/>	OPERATOR	2	20,000	3,113	15%	90%		
<input type="checkbox"/>	ZWRITE	2	5,000	687	13%	90%		
<input type="checkbox"/>	ZWEB10	2	180	11	6%	90%		
<input type="checkbox"/>	ZWEB12	2	180	12	6%	90%		
<input type="checkbox"/>	ZPROLOG	2	30,000	1,204	4%	90%		
<input type="checkbox"/>	ZWEB02	2	180	7	3%	90%		
<input type="checkbox"/>	ZWEB04	2	180	7	3%	90%		
<input type="checkbox"/>	ZWEB07	2	180	7	3%	90%		
<input type="checkbox"/>	ZWEB09	2	180	7	3%	90%		
<input type="checkbox"/>	ZADMIN	2	180	4	2%	90%		
<input type="checkbox"/>	ZPROXY	2	900	24	2%	90%		
<input type="checkbox"/>	ZWEB06	2	180	5	2%	90%		
<input type="checkbox"/>	DEMOZPRO	2	5,000	7	0%	90%		



## Controlling filespace allocation

```
ENROLL USER ZWEB05 (BLOCKS 180)
```

## Increase/reduce filespace allocation

```
MODIFY USER +100 FOR ZWEB05
```

```
MODIFY USER -50 FOR ZWEB01
```

## Delete filespace

```
DELETE ZWEB01
```

**Functions for ZWEB05 in filepool SFSZVPS4** ✕

Increase/Decrease User Quota      4K Blocks:

Current allocation is 180 of 180 blocks(100%)  
Enter a new max value in the next field and click to change

Grant/Revoke Authorization

Cancel

## Add space to a storage group

- Add a minidisk to the server
- Create a file describing the new minidisk
- Expand storage group

## Add minidisk to the server

```
MDISK 0307 3390 3251 0500 VM5W02 WR
```

## Create a file describing the new minidisk

- On an admin ID
- Eg: NEW307 POOLDEF A

DDNAME=MDK00004 VDEV=307 GROUP=2

```
MAXUSERS=500
MAXDISKS=500
DDNAME=CONTROL          VDEV=301
DDNAME=LOG1             VDEV=302
DDNAME=LOG2             VDEV=303
DDNAME=BACKUP   DISK   FN=CONTROL   FT=BACKUP   FM=C
DDNAME=MDK00001        VDEV=304        GROUP=1   BLOCKS=1790
DDNAME=MDK00002        VDEV=305        GROUP=2   BLOCKS=80906
DDNAME=MDK00003        VDEV=306        GROUP=2   BLOCKS=3588
```



## Expand storage group

### VMLINK MAINT 193

DMSVML2060I MAINT 193 linked as 0120 file mode X

### FILEPOOL MINIDISK VMSYSVPS NEW307 POOLDEF A

DMSWFP3485I FILEPOOL processing begun at 07:53:36 on 15 Jul 2011.  
DMSJMD3426I The following minidisk(s) will be formatted and reserved  
DMSJMD3426I for VMSYSVPS on RKS2LV  
DMSJMD3426I MDK00004 0307 00002  
DMSJMD3427R FORMAT will erase all files on the above minidisk(s).  
DMSJMD3427R Do you wish to continue? Enter 1 (Yes) or 0 (No)  
1  
DMSJMD3533I Linking to minidisk MDK00003 at 0307 as FFFF.  
DASD FFFF DETACHED  
DMSJMD3423I The minidisk with virtual device address FFFF has been  
formatted  
DMSJMD3922I 1 minidisk(s) added to the file pool  
DMSJMD3428I New minidisk(s) will not be available for use until a  
DMSJMD3428I confirmation message is sent to your virtual reader  
DMSWFP3486I FILEPOOL processing ended at 07:53:55 on 15 Jul 2011.

## Backups

Control data

User data

## Control data

User allocations

Authorizations

Storage group 1, control disk and POOLDEF file

Control backups automatic when logs fill

From an admin

FILEPOOL CONTROL BACKUP



## User data

Files and directories that a user creates

## Two type of user backup

FILEPOOL BACKUP

FILEPOOL UNLOAD

Both admin commands

Can be issued concurrent with filepool operations

## FILEPOOL BACKUP

Used to back up a storage group

FILEDEF must be issued to define the backup file

```
FILEDEF BACKUP DISK VMSYSVPS BKUP2 D  
FILEPOOL BACKUP 2
```

## Restore from BACKUP

Restore a storage group

FILEDEF must be issued to define the backup file

```
FILEDEF RESTORE DISK VMSYSVPS BKUP2 D  
FILEPOOL RESTORE 2
```

## Restore from BACKUP

Restore individual files

Create a file containing the list of files to restore

- CONTROL FILELOAD A
- *filename filetype dirname*

eg:

```
INSTALL LOG ZVPS.  
ZMON 20140531 ZVPS.CONSOLES
```

FILEDEF must be issued to define the backup file

```
FILEDEF RESTORE DISK VMSYSVPS BKUP2 D  
FILEPOOL FILELOAD 2
```

## FILEPOOL UNLOAD

Used to back up a storage group or filespace

Storage group, file space or individual files can be recovered

FILEDEF must be issued to define the backup file

```
FILEDEF UNLOAD DISK VMSYSVPS UNLD2 D
```

```
FILEPOOL UNLOAD GROUP 2
```

- *or* -

```
FILEPOOL UNLOAD FILESPACE ZVPS
```

## Restore from UNLOAD

Restore storage group

FILEDEF must be issued to define the backup file

```
FILEDEF RELOAD DISK VMSYSVPS UNLD2 D  
FILEPOOL RELOAD GROUP 2
```



## Restore from UNLOAD

Restore file space

FILEDEF must be issued to define the backup file

Optional GROUP can be used to move a filespace to a different group

- Use ALL option on UNLOAD
- Backs up aliases and authorizations by other users

```
FILEDEF RELOAD DISK VMSYSVPS UNLD2 D
```

```
FILEPOOL RELOAD FILESPACE USER ZVPS [GROUP n]
```

## Restore from UNLOAD

Restore individual file

Create a file containing the list of files to restore

- CONTROL RELOAD A
- *filename filetype dirname*

eg:

```
INSTALL LOG ZVPS.  
ZMON 20140531 ZVPS.CONSOLES
```

FILEDEF must be issued to define the backup file

```
FILEDEF RELOAD DISK VMSYSVPS UNLD2 D  
FILEPOOL RELOAD FILES
```

## Additional information

All commands require access to MAINT 193

Backups/Restores can use tape

Read usage notes!

Open Extensions implements POSIX standards under CMS

POSIX 1003.1 (POSIX.1) - System Interfaces

POSIX 1003.1a (POSIX.1a) - Extensions to POSIX.1

POSIX 1003.1c (POSIX.1c) - Threads

POSIX 1003.2 (POSIX.2) - Shell and Utilities

Provides POSIX compliant file system

Shell and Utilities

Traditionally used to port UNIX features

Now used for z/OS packages on z/VM

- LDAP, SSL, MPROUTE, etc.

## Provides POSIX compliant file system

- Stored as a filespace in SFS

- Byte-stream view and format

- Long file/directory names

- Many possible directory levels

- Unix style permissions

- Unix command set

## Shell and Utilities

Provides a command line interface

Runs OE programs

Environment variables and aliases

Shell script programming

Manipulate BFS files

Ported applications

Development utilities

## Creating a BFS filesystem

ENROLL USER RKS0 (BLOCKS 20000 BFS)

## Specifying a BFS path in CMS command

././VMBFS:filepool:filesystem/

Required part  
of the path  
identifier

File pool that  
contains the  
data

BFS filesystem  
to reference

## Integrating BFS actions in CMS

OPENVM command allows manipulation of BFS objects from CMS

CREATE DIRECTORY - Create a directory

ERASE - Delete a file or directory

GETBFS - Extract BFS object into CMS

LISTFILE - List files in a directory

MOUNT/UNMOUNT - Mount/unmount a filespace

OWNER - Change owner of file or directory

PERMIT - Change permissions of file or directory

PUTBFS - Insert CMS file into BFS

SET DIRECTORY - Change current working directory

SHELL - Start OPENVM shell



## Starting the shell

```
openvm mount ../../vmbfs:vmcfs:root/ /  
Ready; T=0.01/0.02 08:05:03  
openvm shell
```

```
IBM  
Licensed Material - Property of IBM  
5654-A17 (C) Copyright IBM Corp. 1995  
(C) Copyright Mortice Kern Systems, Inc., 1985, 1993.  
(C) Copyright Software Development Group, University of Waterloo, 1989.
```

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

#

## Starting the shell

```
#
pwd
/
#
ls -l
total 0
drwxr-xr-x  1 maint630 system      0 May 29  2013 bin
drwxr-xr-x  1 maint630 system      0 May 29  2013 dev
Erwxrwxrwx  1 maint630 system    21 May 29  2013 etc -> ../../VMBFS:VMSYSU:ETC
drwxr-xr-x  1 maint630 system      0 May 29  2013 home
lrwxrwxrwx  1 maint630 system      8 May 29  2013 lib -> /usr/lib
drwxr-xr-x  1 maint630 system      0 May 29  2013 opt
Erwxrwxrwx  1 maint630 system    21 May 29  2013 tmp -> ../../VMBFS:VMSYSU:TMP
lrwxrwxrwx  1 maint630 system      5 May 29  2013 u -> /home
drwxr-xr-x  1 bin          bin      0 May 29  2013 usr
Erwxrwxrwx  1 maint630 system    21 May 29  2013 var -> ../../VMBFS:VMSYSU:VAR
```

## Similar command set

```
#  
ps -ef  
  UID      PID      PPID  STIME TTY      TIME COMMAND  
maint630  3868        1 17:42:23 tty      0:00  
maint630  4098        1 16:46:59 tty      0:00  
maint630  4360      3868 17:42:24 tty      0:00 sh -L  
maint630  4625      4360 20:40:03 tty      0:00 ps -ef
```

## Other commands

alias, awk, bg, c89, cat, cd, chgrp, chmod, chown, cp, cut, date, echo, find, grep, head, join, ln, ls, make, mv, pax, sleep, tail, tar, type

## Referencing a CMS file within the shell

### **cat //cor.doc.a**

Please reference the Packing list, Media map and Service install document located in file one of the electronic document envelope that was received with your PTF(s):

- The Packing list is identified by COR VMELEDOC.
- The Media map is identified by CORnn MEDIAMAP.
- The Service install document is identified by either COR DOCUMENT (for VMSES) or SERVICE DOCUMENT (for Non-VMSES).

### **ls -l /usr/include/c\***

```
Erw-r--r--  1 bin      bin      27 May 29  2013 ceedcct.h -> //CEEEDCCT.H.*
Erw-r--r--  1 bin      bin      23 May 29  2013 cics.h  -> //CICS.H.*
Erw-r--r--  1 bin      bin      26 May 29  2013 collate.h -> //COLLATE.H.*
Erw-r--r--  1 bin      bin      26 May 29  2013 complex.h -> //COMPLEX.H.*
Erw-r--r--  1 bin      bin      23 May 29  2013 cpio.h  -> //CPIO.H.*
Erw-r--r--  1 bin      bin      22 May 29  2013 csp.h   -> //CSP.H.*
Erw-r--r--  1 bin      bin      24 May 29  2013 ctest.h -> //CTEST.H.*
Erw-r--r--  1 bin      bin      24 May 29  2013 ctype.h -> //CTYPE.H.*
```

## Viewing the contents of a file

```
#  
cd home  
#  
ls -l  
total 0  
drwxrwxrwx  1 zvps  staff      0 Jul  4  2012 zvps  
#  
ls -l zvps  
total 8  
-rw-r--r--  1 zvps  staff    41 Jul  4  2012 hello.cgi  
#  
cat hello.cgi  
/* HELLO CGI */  
  
'output Hello, World.'  
#
```

## Invoking XEDIT for a BFS file

### Outside the shell

```
xedit ../../vmbfs:vmsys:root/home/zvps/hello.cgi (nametype bfs
```

```
...me/zvps/hello.cgi  V 80  Trunc=80  Size=3  Line=0  Col=1  Alt=0
=====
|...+....1....+....2....+....3....+....4....+....5....+....6....+....7....
===== * * * Top of File * * *
===== /* HELLO CGI */
=====
===== 'output Hello, World.'
===== * * * End of File * * *

=====>
X E D I T  1 File
```

## Invoking XEDIT for a BFS file Within the shell

Create a shell script in /usr/bin

```
cat xedit
#!/bin/sh
cms xedit $1 '(nametype bfs' $2
```

Make it executable

```
chmod +x xedit
```

Run it

```
xedit ../vmbfs:vmsys:root/home/zvps/hello.cgi
```

## Mount NFS exports in BFS

```
openvm mount ../nfs:192.168.5.209/home/rksdev/ /home/opensuse  
(userid rksdev password xxxx
```



## A number of tools ported

VM OpenEdition Ported Packages

<http://vm.marist.edu/~neale/vmoe.html>

gzip, gnumake, syslog, ldapd, apache

“Porting Unix Applications to OpenEdition for VM/ESA” - SG24-5458

Read about the process

## VM Downloads page

<http://www.vm.ibm.com/download/packages/>

Name	Types	Date	Abstract
SFSKTOOL	v-365K	2011-02-28	Various tools related to SFS management. V1.0
SFSULIST	v-134K	2011-02-28	<u>SfsUserList</u> lets you handle SFS users in a FILELIST like way V1.13
SETDATE	v-9K	2009-10-27	SETDATE EXEC - Set the date of last update on CMS or SFS files.
NONAMES	v-5K	2008-08-06	Avoid NAMEFIND <u>lookup</u> for SFS administrative commands
ERALIAS	v-8K	2002-10-25	An exec to remove <u>ERASEd ALIASes</u> from the SFS catalog.
DIREXEC	v-65K	2000-05-01	Execute or <u>execload</u> execs from non-accessed SFS directories.
XSFS	v-76K	2000-05-01	XEDIT files that reside in <u>unaccessed</u> SFS directories.
OBFUSCAT	v-8K	1996-12-16	Prevent direct links by randomly changing file and HTML link names in SFS.
FPUBLIC	v-16K	1996-08-26	Identify SFS Files and Directories with PUBLIC Access
PIPEFILE	v-47K	1996-08-26	Modern Pipelines file I/O using DMSREAD/DMSWRITE; SFS
SFSTAR	v-32K	1996-08-26	Archive (zip up) SFS directory tree into file/pipeline stream
WORF	v-30K	1996-08-26	Simple Shared File System (SFS) external security manager (ESM)
WORFAUTH	v-11K	1996-08-26	WORF SFS ESM rule database updater

CMS File Pool Planning, Administration, and Operation

CMS Commands and Utilities Reference

CMS User's Guide

OpenExtensions Shell Commands

OpenExtensions User's Guide

OpenEdition for VM/ESA Implementation and  
Administration Guide (SC24-4747)

Porting UNIX Applications to OpenEdition for VM/ESA  
(SG24-5458)

## Questions?

Velocity Software Inc.  
196-D Castro Street  
Mountain View CA 94041  
650-964-8867

Velocity Software GmbH  
Max-Joseph-Str. 5  
D-68167 Mannheim  
Germany  
+49 (0)621 373844

Rich Smrcina  
Velocity Software, Inc.  
[rich@velocitysoftware.com](mailto:rich@velocitysoftware.com)