

# **z/VM Single System Image Installation and Configuration Workshop**

---

## **The Directory Maintenance Feature (DirMaint)**



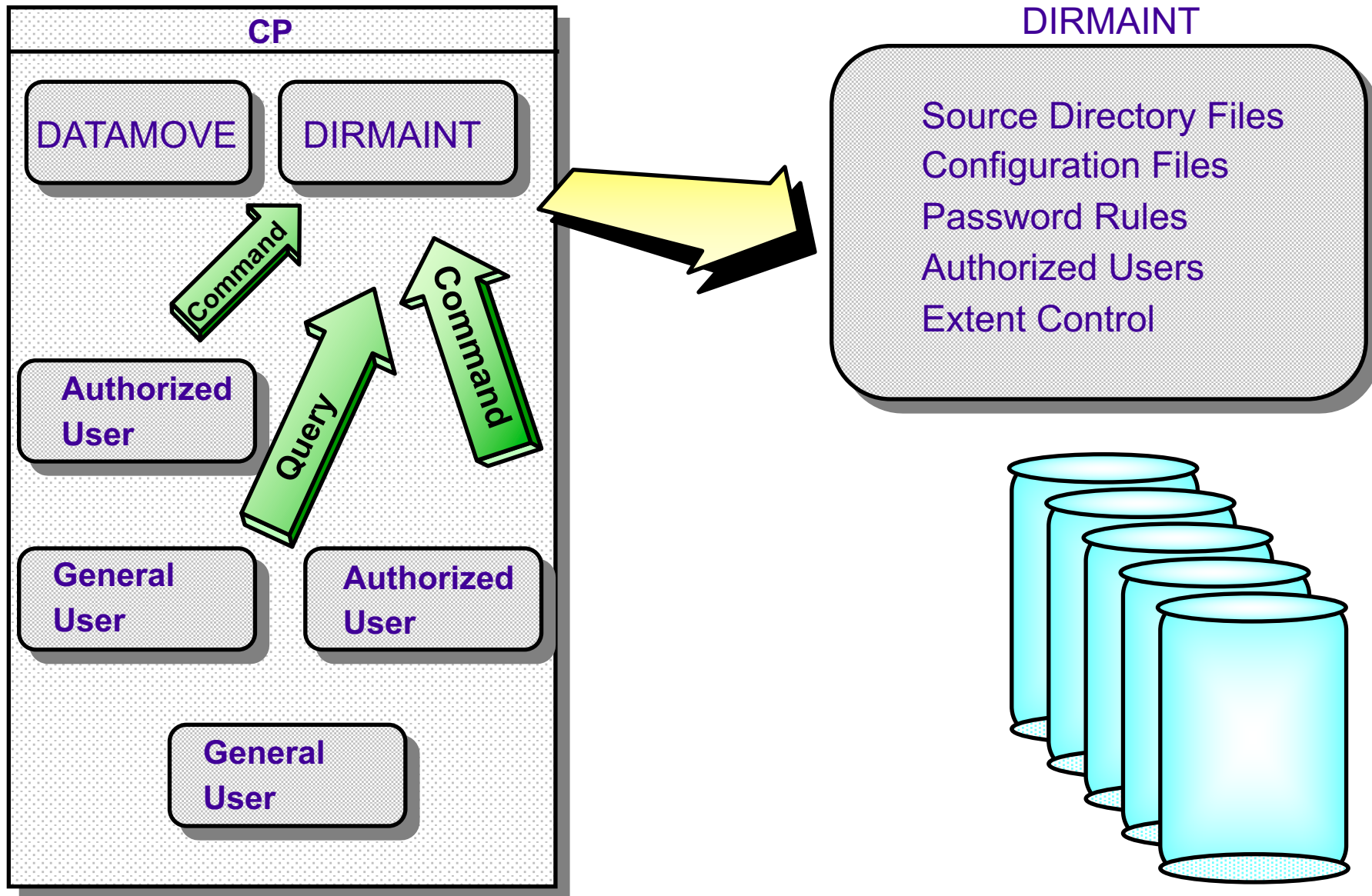
# Overview



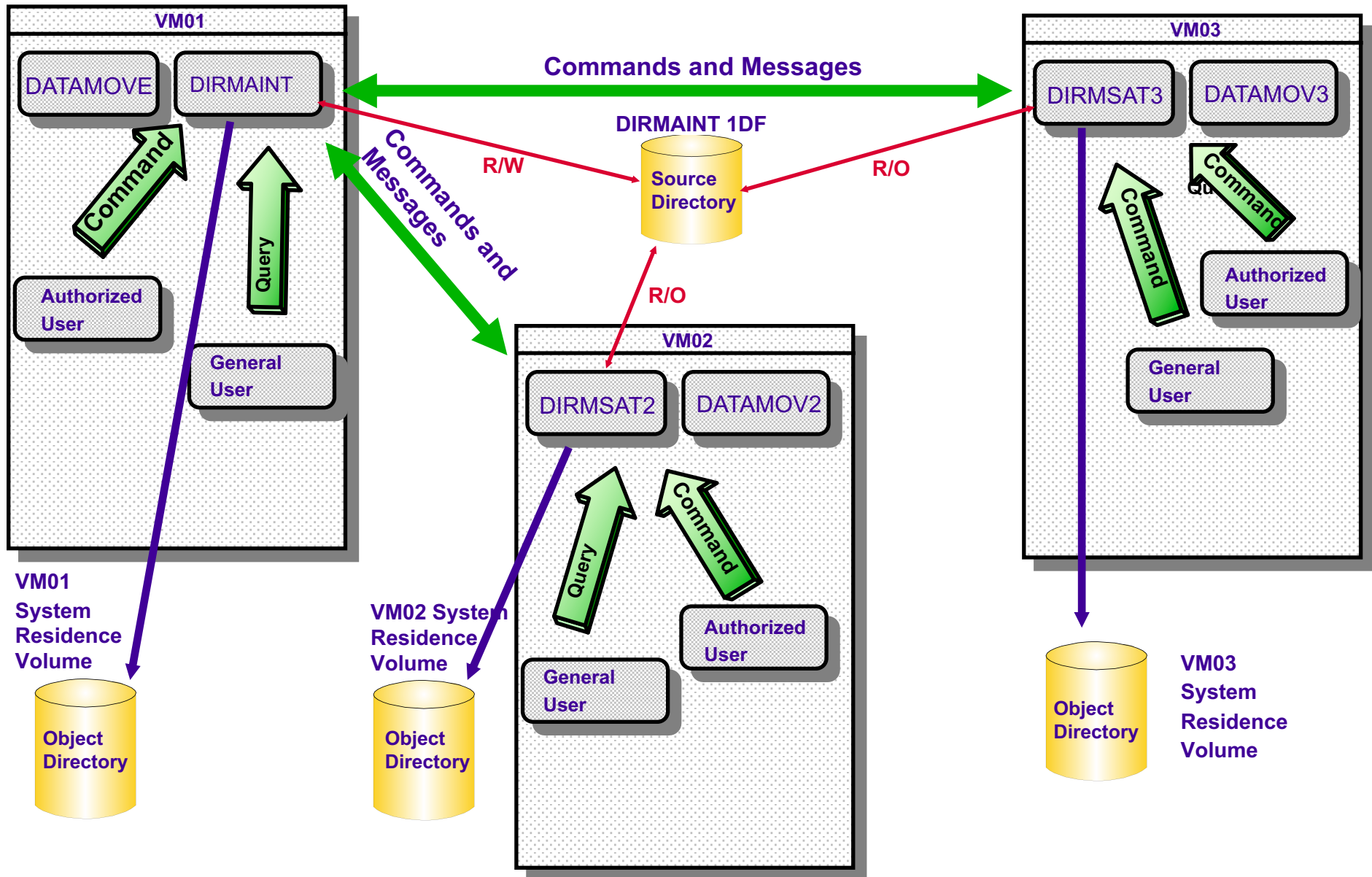
Directory Maintenance (DirMaint) is a program that helps you manage your VM user directory. DirMaint is a priced optional feature of z/VM.

- Provides a command interface to initiate updates and queries to the directory
- Error checking ensures that only valid changes are made
- Automated facilities can allocate, deallocate, and copy minidisks
- All transactions can be password controlled and recorded for auditing
- Server virtual machines provide DirMaint services
  - **DIRMAINT**
  - **DATAMOVE**
  - **DIRMSAT**
- Coordinates directory updates in an SSI cluster
- Can work in conjunction with external security managers, such as RACF.

# How DirMaint Works



# How DirMaint Works in a Cluster



# DIRMAINT Server



The DIRMAINT server owns and manages the directory:

- Receives commands (transactions) from authorized users
- Verifies that the transactions are valid
- Makes the appropriate updates to the source directory
- Uses DIRECTXA to place directory changes online
- Optionally:
  - monitors user passwords
  - controls allocation of DASD
  - allocates work to one or more DATAMOVE servers and monitors their progress

# DATAMOVE Server



The DATAMOVE server is responsible for manipulating minidisks on behalf of the DIRMAINT server. DATAMOVE can:

- Format newly allocated DASD space for the user
- Format a new extent to receive files from an existing disk, copy files from an existing disk to the new extent, and optionally format the old extent to delete remaining files
- Format an old extent being deallocated to prevent exposure of any residual data to the next user.
- Clone (copy) disks, such as from a master image to a new server.

There can be multiple copies of the DATAMOVE server.

# DIRMSAT Server



In an SSI cluster, the DIRMSAT server is responsible for synchronizing the object directory on systems other than the system on which the primary DIRMAINT server runs. The DIRMSAT server:

- Receives notifications from the DIRMAINT server whenever the source directory has been updated and put online by the DIRMAINT server. It places the update online on its own system.
- Routes DIRMAINT commands from users on its own system to the DIRMAINT machine and routes command output files from DIRMAINT back to command users.

There should be one copy of the DIRMSAT server running on each member of an SSI cluster that is NOT running the DIRMAINT server.

# Installing DirMaint



DirMaint is pre-installed on the VM system in a disabled state. In order to use DirMaint you must:

- License the product
- Enable DirMaint in the SYSTEM CONFIG file
- Enable DirMaint to VMSES/E (the VM installation/service component)
- Configure the DIRMAINT, DIRMSAT and DATAMOVE servers
- Move DirMaint code into production
- Automate server startup at IPL

DirMaint product code does not reside on disks owned by the DIRMAINT server. An installation userid called **7VMDIR10** owns these disks. Procedures for installing and customizing DirMaint are located in the *Memo to Users, DirMaint Program Directory* and the *DirMaint Tailoring and Administration Guide*.



# Configuration Files



The behavior of the DirMaint servers is controlled through configuration files. For a basic installation and configuration, the following files are updated:

- **CONFIGxx DATADVH** - Configuration file for the servers
- **CONFIGSS DATADVH** - Defines satellite servers in an SSI cluster
- **EXTENT CONTROL** - defines any volume that is being used for minidisk allocation and provides a template of how the space should be used

# CONFIG DATADVH



**CONFIG DATADVH** contains a large number of customization options. Most of these options have acceptable defaults. A few options must be updated in order to run DirMaint. **CONFIG DATADVH** is used to set:

- Directory update options
- Restart and recovery characteristics
- Security and auditing characteristics
- Password control characteristics

The **CONFIG DATADVH** file itself is NEVER modified. You create an override files to change the options.

- Called CONFIG\* DATADVH where \* is 0-99, A-ZZ
- Contents are searched in reverse alphabetical order. For example, CONFIG99 is searched before CONFIGA.

# Sample CONFIG DATADVH (Partial)



```
RUNMODE= TESTING | OPERATIONAL
SRCUPDATE= NOP | DISABLED
ONLINE= OFFLINE | SCHED | IMMED
UPDATE_IN_PLACE= YES | NO
WRK_UNIT_ONLINE= NO | YES
WRK_UNIT_CLEANUP= ERASE | RENAME
/ DIRECTXA_OPTIONS= | <MIXED <NOMIXMSG>> <&SYSRES +VMRES>
DIRECTXA_OPTIONS=
SORT_DIRECTORY= NO | YES
SORT_BY_DEVICE_ADDRESS= NO | YES
SORT_COMMENTS_WITH_STATEMENTS= YES | NO
BACKUP_REBUILD= CLUSTER DVHLINK
DVHWAIT_BATCH_INTERVAL= 00:01 | mm:ss | hh:mm:s
DVHWAIT_CLUSTER_INTERVAL= 00:15 | mm:ss | hh:mm:s
DVHWAIT_IDLE_INTERVAL= 05:00 | mm:ss | hh:mm:s
SHUTDOWN_LOGOFF_THRESHOLD= 3 /* Choose 2, 3, or 4. */
SHUTDOWN_RESET_THRESHOLD= 3 /* The s_r_t must be >= 1.*/
SHUTDOWN_REIPL_COMMAND= CP IPL CMS PARM AUTOOCR
```

# Basic CONFIG DATADVH Configuration



There are MANY configuration options available in the CONFIG DATADVH file. Only a few changes are necessary for basic DirMaint configuration:

- **RUNMODE=OPERATIONAL**

By default, DirMaint runs in 'TESTING' mode. No changes are made to the source directory while DirMaint is in testing mode.

- **ONLINE=IMMED**

By default, changes made to the source directory are not placed online with the DIRECTXA command. These changes should be placed online 'IMMED'ately.

- **DATAMOVE\_MACHINE=DATAMOVE \* \***

This statement identifies the userid of the DATAMOVE server. It is commented out in the default configuration file.

- **DISK\_CLEANUP=YES**

If you want DATAMOVE to format disks after they have been de-allocated, set this option to YES.

# CONFIGSS DATADVH



**CONFIGSS DATADVH** is used in an SSI cluster environment to define a satellite and datamove server for each SSI member system. This file is created automatically during the z/VM installation process when you choose an SSI installation. Here is the file created during the lab installation:

```
SATELLITE_SERVER= DIRMSAT VM01
SATELLITE_SERVER= DIRMSAT2 VM02
DATAMOVE_MACHINE= DATAMOVE VM01      *
DATAMOVE_MACHINE= DATAMOV2 VM02      *
```

**CONFIGSS DATADVH** is a override file for **CONFIG DATADVH**.

# EXTENT CONTROL



To automatically allocate minidisks without specifying the exact location on a volume, DirMaint needs to know what volumes are available and how they should be used. The EXTENT CONTROL file:

- Defines volumes used for minidisk allocation
- Defines how the space should be used
- Contains system and default device values

EXTENT CONTROL is stored on the DIRMAINT 1DF disk.

# Default Extent Control File



```
:REGIONS.  
  *RegionId      VolSer      RegStart      RegEnd      Dev-Type  
:END.  
:GROUPS.  
  *GroupName RegionList  
:END.  
:SSI_VOLUMES.  
*VolumeFamily      Member      VolSer  
:END.  
:EXCLUDE.  
  * UserId Address  
:END.  
:AUTOBLOCK.  
  * IBM supplied defaults are contained in the AUTOBLK DATADVH file.  
  * The following are customer overrides and supplements.  
  *  
  *DASDType BlockSize Blocks/Unit Alloc_Unit Architecture  
:END.  
:DEFAULTS.  
  * IBM supplied defaults are contained in the DEFAULTS DATADVH file.  
  * The following are customer overrides and supplements.  
  *  
  *DASDType Max-Size  
:END.
```

# EXTENT CONTROL - REGIONS



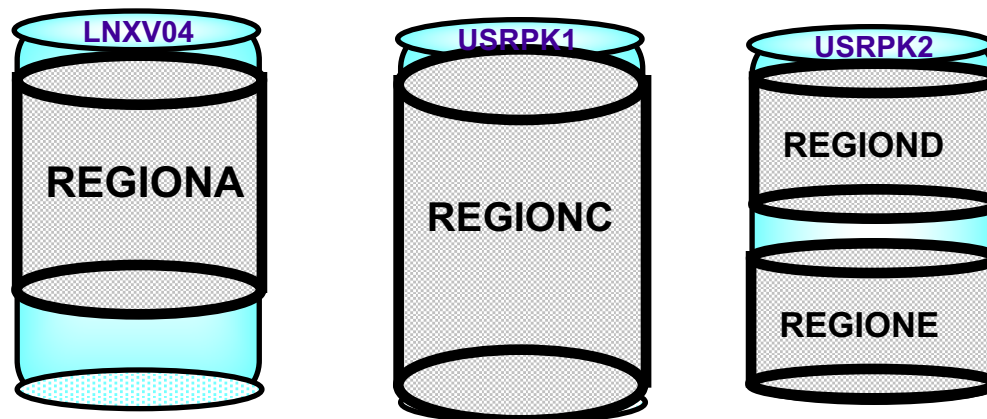
A **Region** is the basic unit of DASD segmentation used by DirMaint. It defines a single, contiguous area on a single DASD volume. Regions are defined as follows:

Regionid	Volser	RegStart	RegEnd	Dev-Type
----------	--------	----------	--------	----------

**Example:**

RegionA	LNKV04	50	2000	3390-03
RegionC	USRPK1	1	END	3390-03
RegionD	USRPK2	1	1000	3390-03
RegionE	USRPK2	1500	END	3390-03

Every volume that is used for automatic minidisk allocation must have a region or regions defined for it.





# EXTENT CONTROL - GROUPS



A **Group** is a collection of one or more regions. They are defined as follows:

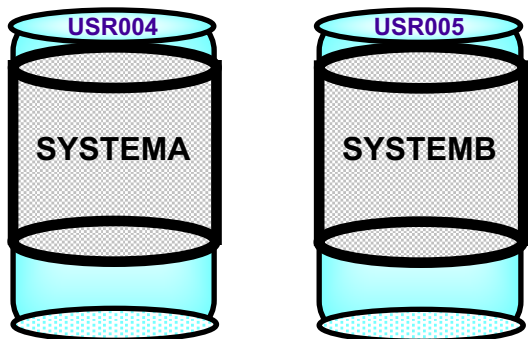
groupid      region1      region2      ....      regionn

**Example:**

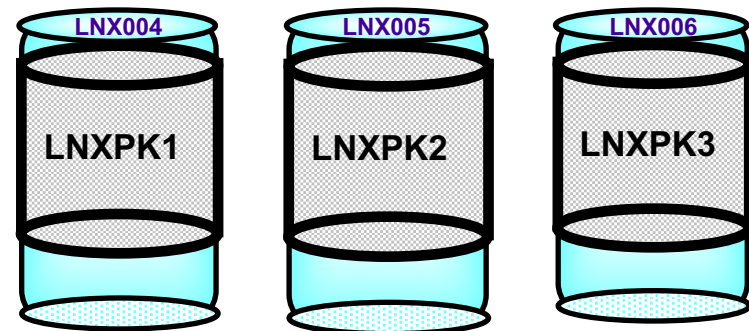
GrpSyS      SystemA      SystemB  
GrpLnx      Lnxpk1      Lnxpk2      Lnxpk3

Regions specified in a group entry must be defined in the **Regions** section.

**GRPSYS**



**GRPLNX**



# EXTENT CONTROL - EXCLUDE



## Exclude Section

Some userids contain minidisk extents that deliberately overlap other minidisks. For example, a virtual machine used for system backup may have a minidisk that spans the entire volume to be backed up, which overlaps all minidisks contained on the volume. In order to exclude these minidisk extents from extent checking during DirMaint DASD allocation operations, they must be listed in the Exclude section in the following manner:

**userid address or subconfig address**

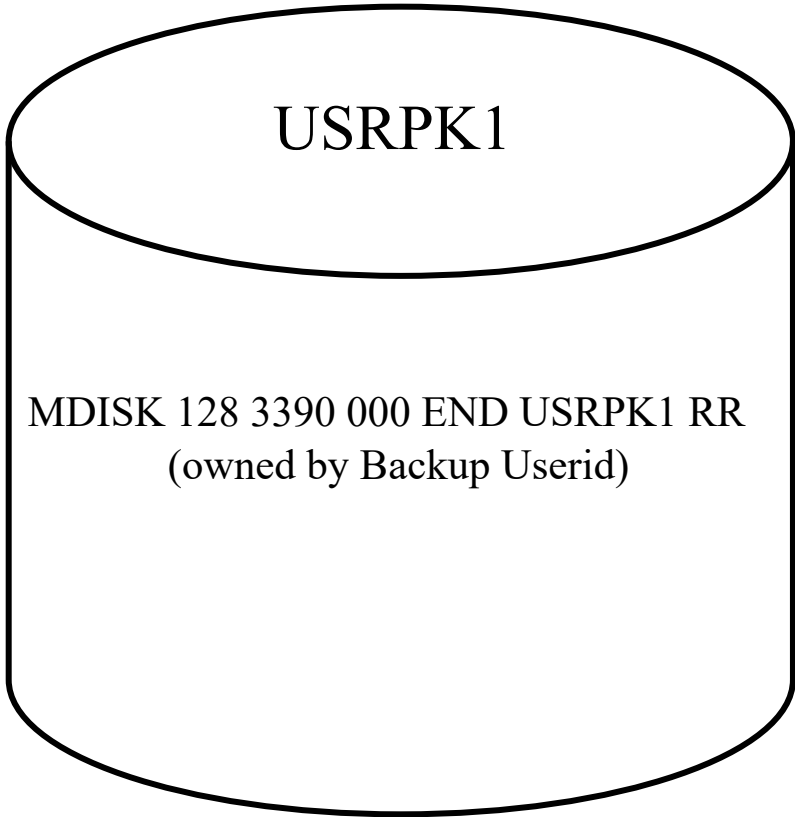
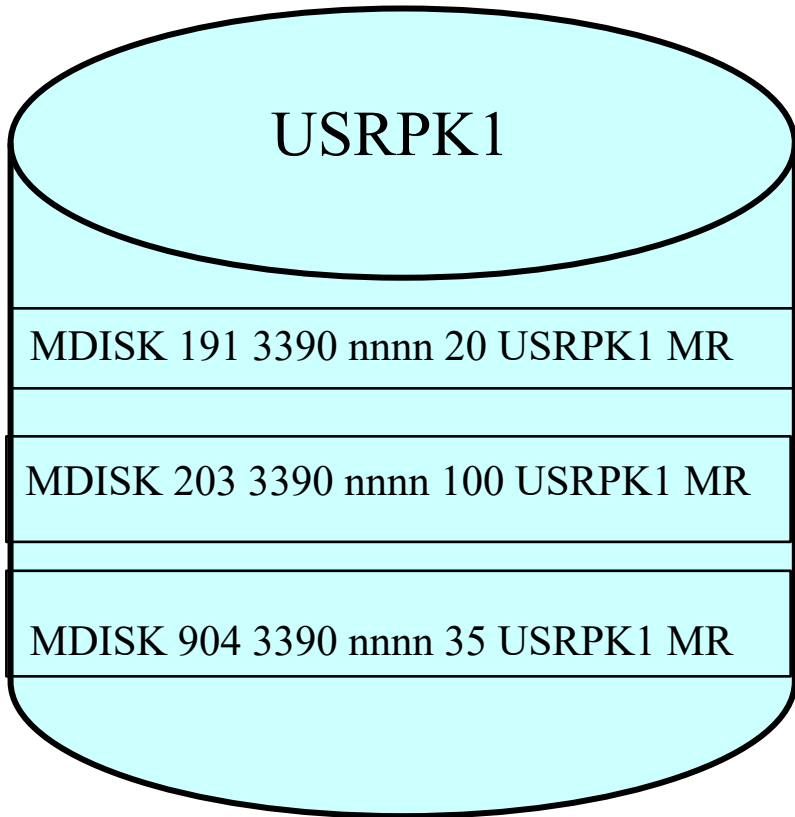
Examples:

**BKUP-\* 001\*** (excludes minidisks 0010-001F defined in BKUP- subconfigs)

**LIN\* 3044** (excludes the 3044 disk for all userids beginning with 'LIN')

# EXTENT CONTROL - EXCLUDE

*(continued)*



To perform system backup, a user has a minidisk defined at virtual device address 128 that spans the entire volume. If this minidisk is not listed in the exclude section of EXTENT CONTROL, DirMaint will not allocate unused cylinders on USRPK1 for new minidisks.

# EXTENT CONTROL - DEFAULTS and AUTOBLOCK



## Defaults Section

In the Regions section, each region has a DASD type/model associated with it. The defaults section allows you to define a DASD type/model and the number of maximum allocatable cylinders or blocks for that type/model. The type/model must match exactly the entry in the **regions** section.

**ttttmmmm maxcyl**

Example:

**3390-3      3339**

Most dasd types are predefined in the DEFAULTS DATADVH file. There is no need to update this section in EXTENT CONTROL unless you have special allocation requirements.

## Autoblock Section

Autoblock entries are used by DIRMAINT to calculate the number of cylinders to allocate with some allocation methods. Defaults are provided in file AUTOBLK DATADVH. You normally do not need to update this section.

# DirMaint Commands



The **DIRMAINT** command provides the interface between the user and the DIRMAINT virtual machine. DirMaint provides a command for every directory statement, plus additional commands for managing DirMaint itself.

- Commands mimic the format of the directory entries
- Online HELP is available for every command and message
- Command authorization is controlled by use of command sets
  - Similar to CP command privilege structure
- Commands are issued to DirMaint through the **DIRMaint** command
  - For the user issuing the command
  - For another user

# DirMaint Command Classes



Class	User and Function
A	Administration, non-DASD related
D	DASD Management
G	General users
H	Helpdesk
M	Password Monitor
O	System Operator
P	DASD management automated Programs, such as DFSMS/VM
S	Support Programmer
Z	Internal communication

# DIRMAINT Help



—Enter HELP ADVH MENU or DIRM ?

ADVH MENU

Menu Help Information

line 10 of 34

*DVHAMENG	CLAss	DLink	IOPriori	NOTAPE	QUery	SPOOL
*DVHUCENG	CLEANUP	DMDisk	IPL	OFFline	REPlace	SSI
*UDVH	CLONEDisk	DROPBy	IUCV	ONline	REView	STAG
?	CMDisk	DROPFor	Link	OPTion	RLDCode	STATus
:ADVH	CMS	DROPScif	LOADDEV	PAValias	RLDData	STDEvopt
:HELP	COMMAND	DSECuser	LOCK	POOL	RLDExtn	STorage
ACCount	CONsole	DUMP	LOGmsg	POSIXFSRo	RMDisk	SUBscribe
ACIgroup	CP	D8ONECMD	LOGONBY	POSIXGLIs	SATellite	SUPGLIST
ACNTAdd	CPU	ELink	MACHine	POSIXGRou	SCAN	SYSaffin
ACNTDel	CRYpto	ENable	MAIL	POSIXINFo	SCReen	Term
Add	DASD	EXECDrop	MAXSPool	POSIXIUPg	SECuser	TESTpw
AMDisk	DASDOPT	EXECLoad	MAXstorag	POSIXIWDi	SEND	TMDisk
APPCpass	DATAmove	EXTNchk	MAXstore	POSIXOPT	SETAcnt	UNDOSSI
AUTHBy	DATEForma	FILE	MDAUDit	PRIORity	SETCclass	UNLock
AUTHFor	DEDicate	FREEExt	MDisk	PRI0set	SETCPU	USEDext
AUTHLink	DEFAULTs	Get	MDPW	PRIVclass	SETMach	USER
AUTHScif	DEFINESTa	GETCONsol	MINIOPT	PURGE	SETOptn	USERMAP
AUTOlog	DIRECT	GLOBALOpt	MMDisk	PW	SETPRiori	USEROPtn
BACKUP	DIRECTORY	GLOBALv	NAMEsave	PW?	SETpw	VMRELOCat
BATCh	DIREEDIT	HELP	NEEDPASS	PWGen	SETSTAG	WORKUNIT
CHECK	DIRMAP	HISTory	NEWS	PWMON	SHARE	XAUTolog
CHKsum	DISABle	INClude	NICDEF	QLog	SHUTDOWN	XCONfig
CHNgid	DISTRib	INVen	NOPdata	Qry	SPEcial	XSTORE
CHVaddr						

\*\*\* End of File \*\*\*

PF1= Help      2= Top            3= Quit        4= Return     5= Clocate    6= ?  
PF7= Backward 8= Forward    9= PFkeys    10=           11=           12= Cursor

====> \_

Macro-read 1 File

# DIRMAINT Command Examples



–Change your CP logon password:

**DIRMAINT PW**

–Change another user's password:

**DIRMAINT FOR BOB PW**

–Add a LINK statement to a user's configuration:

**DIRMAINT FOR BOB LINK TCPMAINT 592 592 RR**

–Tell DIRMAINT to save a copy of a file to one of its disks:

**DIRMAINT FILE LINDB PROTODIR A = = A**



# DIRMAINT Command Examples (continued)



-Add a 10 cylinder minidisk (AMDISK) at virtual address 999 to user BOB's virtual machine:

On region REGIONB:

**DIRMAINT FOR BOB AMDISK 999 3390 AUTOR 10 REGIONB MR**

On a region in the USRGRP group:

**DIRMAINT FOR BOB AMDISK 999 X AUTOG 10 USRGRP MR**



USRGRP = REGIONA REGIONB REGIONC

# Create a Userid



Use the **DIRMAINT ADD** command to create a virtual machine directory entry. The user's configuration can be defined in one of two ways:

- 1 The directory statement entries for the user reside in a file on a disk accessible to the issuing userid, where the filename is the userid to be created and the filetype is DIRECT.

**DIRM ADD *userid or identity***

- 2 A prototype file is stored in the DIRMAINT server's A-disk. It can be used as a template for new userids. When a virtual machine is created using a prototype, all of its directory entry statements will be identical to the prototype's except the userid, password, and MDISK statement(s).

The prototype file must have a filename of the prototype name and a filetype of PROTODIR. Use the keyword LIKE on the DIRM ADD command to indicate that you want the new userid to be configured using a prototype.

**DIRM ADD *userid or identity* LIKE *prototype* PW *password***

# Create a Userid (1)



## JAMES DIRECT

**Example: DIRM ADD JAMES**

**Where the userid issuing the command has this file (called JAMES DIRECT) on it's A-disk.**

**Notice that the AMDISK command was used in place of an MDISK statement to allow DIRMAINT to select the location of the minidisk.**

```
USER JAMES SEC21PR 64M 128M G
IPL CMS PARM AUTOOCR
MACHINE ESA
CONSOLE 009 3215
SPOOL 00C 2540 READER *
SPOOL 00D 2540 PUNCH A
SPOOL 00E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
AMDISK 0191 3390 AUTOG 0050 GRPSYS MR
```

# Create a Userid (2)



## Example: DIRM ADD SHEILA LIKE GENUSER PW NEWPASS

### GENUSER PROTODIR

Where the DIRMAINT userid has this file (a prototype directory called GENUSER PROTODIR) on it's A-disk.

When the **DIRM ADD SHEILA LIKE GENUSER** command is issued, a new virtual machine directory entry is created using all of the statements in GENUSER PROTODIR except:

- the userid and password are changed to the parameters given in the command
- the AMDISK statement is replaced by DIRMAINT with the correct MDISK statement.

```
USER GENUSER GENUSER 64M 128M G
IPL CMS PARM AUTOOCR
MACHINE ESA
CONSOLE 00C 3215
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
AMDISK 0191 3390 AUTOG 0050 GRPUSR RR
```

# Create an Identity (1)



Identities and Subconfigs are separate entities in the directory, so each is handled separately in DirMaint. Use DIRM ADD for each. The Identity must be created first, followed by the Subconfigs.

## TUSER DIRECT

### DIRM ADD TUSER

```
IDENTITY TUSER APASSWD 64M 128M G
IPL CMS PARM AUTOOCR
MACHINE ESA
CONSOLE 00C 3215
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
```

# Create an Identity (2)



## DIRM ADD TUSER-1 BUILD ON VM01 IN TUSER

TUSER-1 DIRECT

SUBCONFIG TUSER-1  
AMDISK 191 3390 AUTOV 005 M01W01

## DIRM ADD TUSER-2 BUILD ON VM02 IN TUSER

TUSER-2 DIRECT

SUBCONFIG TUSER-2  
AMDISK 191 3390 AUTOV 010 M02W01

# Create an Identity (2)



Results in this in the user directory:

```
IDENTITY TUSER APASSWD 64M 128M G
BUILD ON VM01 USING SUBCONFIG TUSER-1
BUILD ON VM02 USING SUBCONFIG TUSER-2
IPL CMS PARM AUTOOCR
MACHINE ESA
CONSOLE 00C 3215
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
LINK MAINT 0190 0190 RR
LINK MAINT 019D 019D RR
LINK MAINT 019E 019E RR
SUBCONFIG TUSER-1
  MDISK 0191 3390 2733 5 M01W01
SUBCONFIG TUSER-2
  MDISK 0191 3390 2728 10 M02W01
```



Added by  
DIRMAINT

# Starting and Stopping DirMaint



Once DirMaint is installed and configured, the servers should be added to the AUTOLOG1 startup process. The **XAUTOLOG** command logs the servers on in disconnected mode and their profiles will execute, starting the DirMaint application. All DirMaint servers by default have passwords of "AUTOONLY". This means that they can **only** be autologged, not logged on to a terminal.

If a server needs to be stopped (for example, after maintenance is applied or when the system is shutting down), use:

- **DIRMAINT AT VM02 SATELLITE SHUTDOWN** (for satellite servers)
- **DIRM SHUTDOWN** (for the local DIRMAINT server)

This gives the DirMaint servers time to complete any currently running transactions, shutdown, and logoff.