Introduction to the CP User Directory

Tony Noto

November 4, 2026

Agenda

- Introduction
- Resources
- SSI vs non-SSI
- User Direct file
- Virtual machine definitions
- IPL of the OS
- SPOOL devices
- Minidisk allocation and usage
- User profiles
- BUILD/SUBCONFIG
- Utilities

Introduction

The CP user directory is used to define virtual machines, assign privileges and allocate resources. This presentation will discuss memory, storage, common profiles and Single System Image (SSI) concerns. In addition, it will cover basic utilities related to User Directory management and touch briefly on Directory Management applications.

Resources

The following reference material will be extremely helpful when managing user definitions and environments.

Reference material

- z/VM: CP Planning and Administration - SC24-6271
- z/VM: CP Commands and Utilities Reference - SC24-6268
- z/VM: CMS Command and Utilities - SC24-6260

Read **T**hose **F**abulous **M**anuals

SSI vs non-SSI

A Single System Image (SSI) cluster

- Consist of between one and eight LPARs enrolled as a member of the cluster and connected via CTCs. The LPARs can reside on multiple CECs as long as within the CTC range limitations.
- Allows for virtual machines to be defined across the cluster in a central location.
- Allows for virtual machines to be defined with different characteristics depending on which cluster member it is running.
- Allows for certain types of virtual machine to be moved between cluster members without interruption of operation.
- Can share “common” resources
- Can often times be manipulated inter-LPAR

A non-SSI system

- A single LPAR that may be one of many LPARS in a CEC but all resources and commands are discreet to the LPAR.

USER DIRECT file

Straight from the z/VM CP Planning and Administration manual …

"The z/VM user directory specifies the configuration and operating characteristics of virtual machines. A z/VM user directory exists in two forms: a source form that consists of one or more CMS files, and an object form, created from the source, on a CP-formatted disk. Multiple object directories can be accessed by the z/VM system, but only one object directory can be active (online) for a system at any given instant."

In a new system, the source directory is initially located on the PMAINT 2CC minidisk. The name and location of the file may be changed by the installation administrator. It may also be migrated to be owned by one of the few commercially available directory managers.

The object directory is located in CP formatted DASD space allocated as "DRCT". This space is typically allocated and formatted by a CP utility such as CPFMTXA.

USER DIRECT file

The source directory can exist in one of two formats:

- Monolithic
- Cluster

A monolithic format contains all of the directory statements.

A cluster (not to be confused with an SSI cluster) format consists of an index file, which in-turn consists of LOAD statements which point to definition files.

However defined, the source directory is processed in the following order

- DIRECTORY statement(s)
- Global definition entry
- PROFILE entries
- USER, IDENTITY, and subconfiguration entries

USER DIRECT file

DIRECTORY statement

Used to define the device that contains the allocated directory space. The device must be a CP-owned volume.

Syntax (non-SSI)

DIRectory *vdev devtype volid* {*altvdev* | EDIT[1]} {*nnnnnn-xxxx sysafnid*}

Syntax (SSI)

DIRectory SSI *vdev devtype volid1* {*volid2* … *volid8*}

(1) The EDIT keyword is left over for HPO compatibilty

Virtual Machine definitions

A virtual machine is defined within the source user directory using a collection of resource defining statements. The initial statement for a virtual machine is either IDENTITY or USER.

Defining a virtual machine as a “USER” indicates that there can only be one active occurrence of that virtual machine within an SSI cluster.

Defining a virtual machine as an “IDENTITY” indicates that there can be multiple active occurrences of the virtual machine within an SSI cluster. Each occurrence may have different characteristics as well as common traits.

Virtual Machine definitions

The USER and IDENTITY directory statements have a similar syntax.

Syntax:

User | IDENTity *userid*[1] *password*[2] {*stor* {*mstor* {*classes*} {*pri*[3]} {*symbols*}}}

(1) A *userid* is a 1 to 8 character name consisting of A-Z, 0-9, @, #, $, _, -
(2) A *password* is a 1 to 8 character string, or one of the following
    (1) NOLOG
    (2) NOPASS
    (3) AUTOONLY
    (4) LBYONLY
(3) The *pri* value is for VM/SP compatibility

Virtual Machine definitions

The following directory statements are used when the asterisk “*” place holder is used on the USER/IDENTITY statement:

- Class *classes*
- STORAGE *stor*
- MAXSTORAGE *mstor*

Virtual Machine definitions

stor - specifies the default memory size of the virtual machine at logon time. It is a number with a single character “storage unit” suffix. Valid suffixes are:

- K - kilobytes (will be rounded up to an M value)
- M - megabytes
- G - gigabytes
- T - terabytes
- P - petabytes
- E - exabytes

Virtual Machine definitions

mstor - specifies the maximum memory size the virtual machine can define. It is a number with a single character “storage unit” as above.

classes - specifies the CP privileges classes A through Z and 1 through 6. IBM-defined classes are A through G. Classes “F” and “H” are generally reserved for IBM use. The IBM-defined classes can be augmented and/or modified using the CP MODIFY COMMAND command.

symbols - specifies the override line-end, line-delete, character-delete, and escape characters

IPL of the OS

The Initial Program Load (IPL) is used to initiate the execution of an Operating System (OS) such as:

- z/VM Control Program (CP)
- Conversational Monitor System (CMS)
- z/VSE
- z/OS
- Some flavor of supported Linux on z

IPL of the OS

IPL syntax

Ipl *systemname* PARM *parm_string*

Ipl *vdev devparms*

- *systemname* specifies the name of an NSS (named saved system) such as CMS, ZCMS, GCS, etc.
- *vdev* specifies the virtual address of an IPLable device. This can actually be the virtual card reader.
- *parm_string* specifies parameters used by the IPL process such as:
  - AUTOCR – automatic carriage return (ENTER)
  - FILEPOOL – Specifies the filepool to use for accessing the “a-disk” instead of the virtual 191

SPOOL devices

Card Reader
Card Punch
Printer

Spool *vdev devtype class*

Console

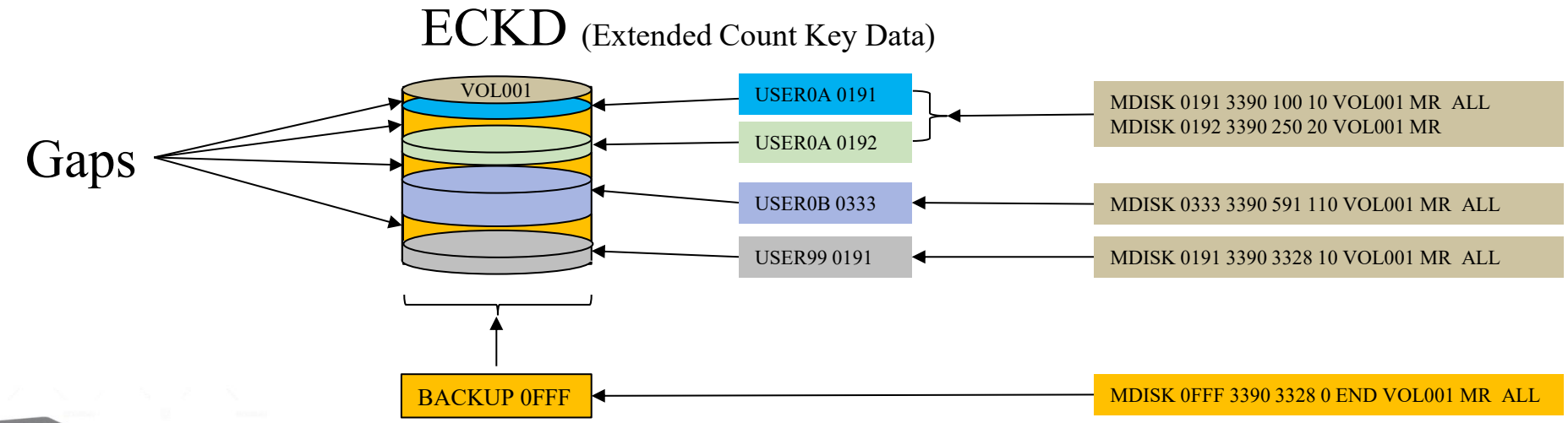
Console *vdev devtype* {*class*|T {*userid* {OBServer}}}

Examples:

```
CONSOLE 0009 3215
SPOOL 000C 2540 READER *
SPOOL 000D 2540 PUNCH A
SPOOL 000E 1403 A
```

Minidisk allocation

A minidisk is all, or part, of a “real” DASD (Direct Access Storage Device). A minidisk is allocated in units of cylinders for ECKD, and 512 byte blocks for FBA (Fixed Block Architecture) devices. The current geometry of a DASD 3390 device is 180 4k blocks per cylinder. The number of cylinders depends on the model number. For instance a 3390-9 (model 9) has 10017 cylinders.







VELOCITY SOFTWARE
PROVEN PERFORMANCE

Minidisk allocation

When defining a minidisk as an FBA device on a 3390 DASD device, it's important to remember that a cylinder contains approximately 1440 512-byte blocks, or 8 * 180 4k blocks.

Virtual machines that require SCSI disk storage do NOT use MDISK statements to define the storage. For SCSI storage, an EDEVICE is required. A SCSI device is defined using the CP SET EDEVICE command or by specifying an EDEVICE statement in the system configuration file.

MDISK statement

Mdisk *vdev devtype operands*

Where:

vdev	- Virtual device address (1-4 hexadecimal digits)
devtype	- Device type (For ECKD generally 3390, for FBA generally FB-512 or 9336)

Minidisk allocation

operands -

ECKD or FBA

- beginning location in cylinders or blocks
- length in cylinders or blocks (use END to indicate end-of-volume)
- Volume serial number
- Mode R, RR, W, WR, M, MR, **MW** (may also contain a mode suffix: V – use virtual reserve/release, S – allow stable access mode, E – allow exclusive and stable access modes, of D – defer definition until a LINK command is issued.
- Passwords
    - *Readpw* or ALL
    - *Writepw* or ALL
    - *Multipw* or ALL

Other

- V-DISK *blks*
- T-DISK *blks* | *cyls*
- DEVNO *rdev*

LINK statement

The LINK statement enables access to another virtual machine's minidisk directly or indirectly.

Link *userid* | * *vdev1 vdev2 mode{suffix}*

Where:

userid	- 1-8 character target virtual machine name. An asterisk (*) indicate self.
vdev1	- Target machine's virtual device number
vdev2	- Virtual device number for this virtual machine's reference
mode	- The mode similar to what would be specified on the MDISK statement

- Mode R, RR, W, WR, M, MR, **MW** (may also contain a mode suffix: V – use virtual reserve/release, S – allow stable access mode, E – allow exclusive and stable access modes, of D – defer definition until a LINK command is issued.)

User Profiles

User profiles allow installations to define a collection of directory statements that are common between multiple virtual machines. Often times these profiles include:

IPL statements
CONSOLE statements
SPOOL definitions
LINK statements
NOT MDISK statements

Virtual machine definitions can utilize a profile definition by specifying an INCLUDE statement as the first statement after the USER/IDENTITY statement.

Syntax:

INclude *profilename*

User Profiles

Sample profile.

The following profile is a copy of the IBMDFLT profile shipped, by IBM, with the base z/VM product

```
CONSOLE 0009 3215 T
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
```

BUILD/SUBCONFIG statements

BUILD statements associate a collection of directory statements defined within a SUBCONFIG to a virtual machine when active on a particular member of an SSI cluster.

- Only occurs for IDENTITY virtual machine definitions
- Always defined in BUILD/SUBCONFIG pairs
- SUBCONFIG statements can only be referenced by one BUILD statement
- Directory statements that appear outside of SUBCONFIG definition are “global”
- Directory statement that appear inside of a SUBCONFIG definition are “local” to the cluster member
- Not all directory statements are allowed within a SUBCONFIG definition

BUILD/SUBCONFIG statements

Sample SSI virtual machine definition

```
IDENTITY USER01 XXXXXXXX 32M 512M BCDEGZ
   INCLUDE IBMDFLT
   BUILD ON LPAR0A USING SUBCONFIG USER01A
   BUILD ON LPAR0B USING SUBCONFIG USER01B
   IUCV *MSG MSGLIMIT 1024
   NAMESAVE ESAMON
*
SUBCONFIG USER01A
   IPL CMS PARM AUTOCR
   MDISK 0191 3390 3173 50 VOL003 MR
   MDISK 0192 3390 1685 28 VOL003 MR
*
SUBCONFIG USER01B
   IPL CMS PARM AUTOCR FILEPOOL SFSPOOL:
   MDISK 0192 3390 3766 1 VOL002 MR
```

vs.

Sample non-SSI virtual machine definition

```
USER USER0D USERPASS 64M 1024M BCDEG
   INCLUDE USERPROF
   IUCV ANY MSGLIMIT 512
   IUCV ALLOW
   IUCV *SPL MSGLIMIT 512
   NAMESAVE TESTER WORKDCSS ZMON ZVWS
   OPTION DIAG88 DEVMAINT MAXCONN 128
   XCONFIG ACCESSLIST ALSIZE 126
   XCONFIG ADDRSPACE MAXNUMBER 16 TOTSIZE 500M SHARE
   LINK TCPMAINT 0592 0592 RR
*
   MDISK 0191 3390 4407 40 VOL001 MR
*
   MDISK 0192 3390 6107 100 VOL009 MR
```

DIRECTXA utility

The DIRECTXA utility is used to (re)create a user directory. It can also be used to check the syntax of the directory statements.

Syntax:

DIRECTXA *fn*|USER *ft*|DIRECT *fm*|* (*options*

Where:

fn - Filename of the user directory
ft - Filetype of the user directory
fm - *Filemode of the user directory*

Options:

EDIT - Check directory statements syntax. Do not update the online directory
Other - …

DISKMAP utility

The DISKMAP CP utility is used to evaluate and summarize the MDISK statements in the source user directory. The output from this utility shows gaps and overlaps between minidisk assignments on DASD devices.

Syntax:

DISKMAP *fn ft*|DIRECT *fm*|* (DOENDS

Where:

fn - Filename of the user directory
ft - Filetype of the user directory
fm - Filemode of the user directory

DOENDS indicates that MDISK statements that had “END” specified are included in the report.

Notes:

- The DIRMAP CMS utility works similarly and produces a similar report. The DIRMAP utility has several extended options that can be specified to refine the report.

- HELP command is available for DIRECTXA, DISKMAP, and DIRMAP utilities.


Questions