z86VM Operations Manual

Mantissa Corporation

1012 Edenton Street, Birmingham, AL 35242 USA T 205-968-3939 F 205-968-3932 W www.mantissa.com

UNITED STATES ▪ UNITED KINGDOM ▪ NORWAY ▪ ITALY ▪ JAPAN ▪ SINGAPORE ▪ KOREA

Intellectual Property of Mantissa Corporation

June 2, 2014 at 4:32 PM

Document Overview

z86VM, the cornerstone of Mantissa's Utility VirtualizationTM product line, allows you to virtualize and run x86- based operating systems (e.g. Linux, Windows®, etc.) on IBM zSeriesTM mainframe systems.The goal of this document is to provide base knowledge of the z86VM environment and the usage of commands and functions associated with this environment. This document will guide you through the installation and verification stages of configuring Z86VM.

Changes from Prior releases of z86VM

As this product is still in Beta release, commands and functions may change prior to General Availability of the product.

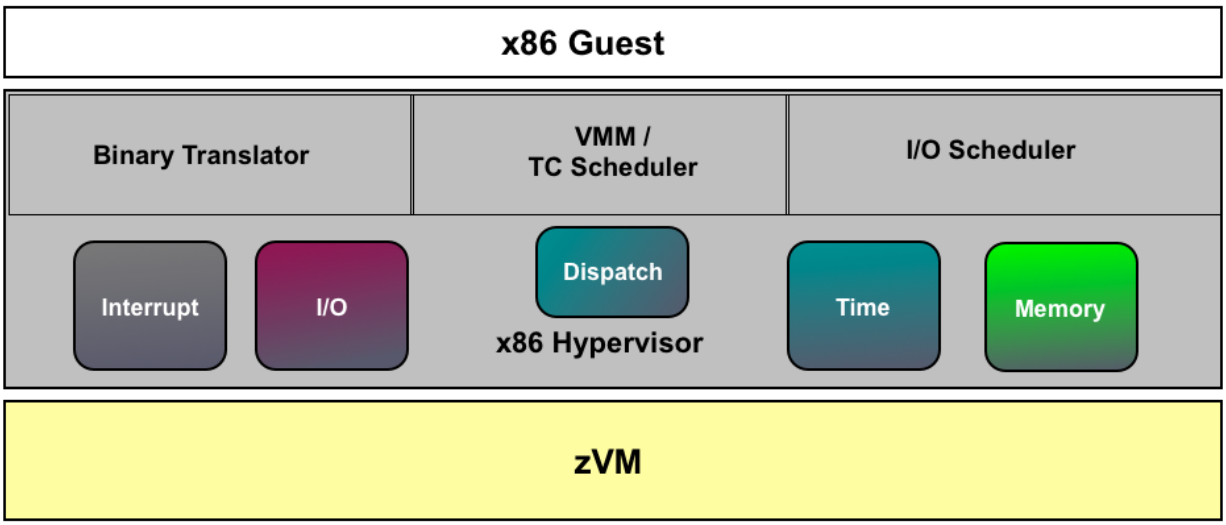
PROFILE Z86

With Beta 1 and Beta 2, the **PROFILE Z86** file contained variables used by the Rexx exec: **Z86 EXEC**. Those variables were in the form *parm.variable = 'value'*. For ease of use and to avoid potential integrity problems with the EXEC, we've changed those parameters to *variable = value*. No apostrophe's are required.

In addition, sub-profiles are now created for the zIO server and each z86 user to allow variations in the different types of users (e.g. client or server) and the size of the virtual machine (e.g. memory, disk capacity). This greatly simplifies deployment of different user types.

Should you have any questions or suggestions for this document, please let us know by contacting us at z86vm@mantissa.com.

Internal organization of z86VM



z/VM

While not part of z86VM, IBM's z/VM product provides the base virtual machine manager for hosting the x86 guests. Each x86 operating image operates as a separate z/VM guest and they share I/O processing and management servers as other z/VM guests.

Binary Translator

This component reads the x86 object code, determines what instruction needs to be executed and branches to a routine to handle that particular instruction. Some instructions have a natural affinity to the z architecture and become a quick translation. Other instructions that have to manage context across instruction operations, such as for i/o or process management will be calling other routines for in depth processing.

Dispatch

Dispatch component is necessary to manage threads and processes for the guest x86 virtual machines. It will maintain context for all users and manage their internal dispatch priorities.

I/O device drivers

With the x86 guests running on System z, a collection of devices must be available to those images as well. Hard Disk drivers will map x86 guest file system to locally attached mainframe disk drives. Screen device drivers, tape subsystems and network attached file systems are examples of other devices drivers that are provided with z86VM.

Interrupt handler

A large advantage of the z architecture is its in interrupt handling. It effectively manages work across multiple processes and can run at 100% processor utilization without fear of failure. The z86VM interrupt handler takes advantages of these same capabilities, but applies them to the x86 workloads. As a result, the processor utilization of x86 virtual machines can run far higher than a native x86 environment. These interrupts may be a result of paging, i/o or thread dispatch of workloads.

Time Management

The original 8088 processor of the x86 architecture had a “pit timer” that would add to a counter and through this recurrent “tick” it was copied into future x86 chips to provide the timer function. This was one of the most difficult pieces of the architecture to recreate. But each x86 operating system running as a virtual machine must depend on an accurate clock and timer processing to maintain the integrity of files, databases and transaction processing. More modern operating systems can handle a Network Time Protocol or external timer. For either of these class of operating systems, this timer function is a critical dependency.

Memory Mapping

x86 operating systems use memory pages to host the real and virtual storage associated with their systems. These pages are mapped to the System z paging infrastructure. Detailed algorithms and exploitation of z architecture have been undertaken to share as many pages (system kernel and shared files) as possible across x86 virtual guests to reduce the amount of real memory that is consumed by the overall z/VM environment.

I/O Scheduler

Another benefit of the z architecture is its asynchronous I/O processing. Again, z86VM takes advantage of that architecture by scheduling independent read and write operations to the storage and network system while dispatching other units of work. Once complete, an interrupt will be scheduled and work on that virtual guest will resume.

Virtual Memory Manager/TC Scheduler

Virtual machine manager (Main Loop), translated code (fragment) scheduler. As part of the z86VM environment, if code segments are repeatedly used, for example, the kernel operating system or a specific application, those code fragments will be cached for reuse to speed up run time performance. The principal is very similar to the JAVA Just in Time compiler. Rather than re-interpret every code string, if it recognizes the code string after several uses, it will compile it to speed up performance. In the case of z86VM, the code isn’t compiled, but rather, a z architecture binary is created from it’s translated x86 binary so that the code can be quickly executed. z86VM will also recognize common code seg-

ments across multiple guest virtual machines. Those code segments will be cached in a common memory to each virtual guest for rapid deployment.

Primary z/VM Virtual Machines

The goal has been and continues to be defining the virtual machines as G class guests so that they have no special privileges. To date, this remains the case. The IO server is the most critical as it has multi-user influence.

Z86MNT

This is the primary virtual machine for managing all other virtual machines within the z86VM functionality. The **Z86 EXEC** is the primary mechanism for operating the environment, including Installation, customization, start, stop and debug commands. Each of these commands is documented in a later section.
Z86MNT is the default name for this guest virtual machine, but it could be tailored based on other z86VM environments running within the same z/VM operating environment.

Z86IO

This is the virtual machine that is tasked with translation and operation of the x86 i/o system and it's mapping to the z architecture i/o system. Imagine that z86VM operates a SAN environment shared across all the x86 guest virtual machines. z86IO would act as the SAN controller and manage access for all the individual guest virtual machines.

z86user

Each x86 guest virtual machine will have it's own owning user id. This can be defined as part of the installation or using several of the Z86 EXEC command options. Each user owns their own files for security purposes. These id's can be cloned copies of each other or different resources, such as virtual memory and locally attached file system, can be defined for the various needs of different virtual machines.

The functional tasks of z86VM

There are certain major areas that must be defined for the functional operation of the z86VM environment. These areas are managed via the **Z86 EXEC** and the commands associated with that exec. Each section, below, will highlight the commands necessary and in a later section, the commands will be described in more detail.

z86VM Boot Drive

Each x86 user in the z86VM environment must have its own virtual machine defined and that must be a bootable x86 operating system. Multiple users may share the same operating system image, which will reduce systems administration time and patch management. However, some users may prefer Linux, some may prefer Windows and then there may be release variations for those operating systems.
The goal of a business deploying z86VM should be to reduce the number of variations of operating system images to reduce change, policy and security administration controls necessary for the many OS variants.

I/O Boot Drive

The ZIO Module must have its own boot drive. This is a shared virtual machine to process the I/O of all z86 virtual guest images. It will translate the requests of the PC based fixed block architecture virtual devices to the real hardware running z format devices.

X86hd Drives

x86hd Drives are one or more 3390 volumes which are formatted (FINIT) as x86 hard drive space. Each z86 guest is allocated a portion of disk space directed by the ' hdmemory' profile parameter.

User Definitions

Each z86VM user must be properly defined to z/VM as a guest virtual machine. Then, each user must have its boot operating system drive associated with it. They also need to have their x86 drives associated with it.

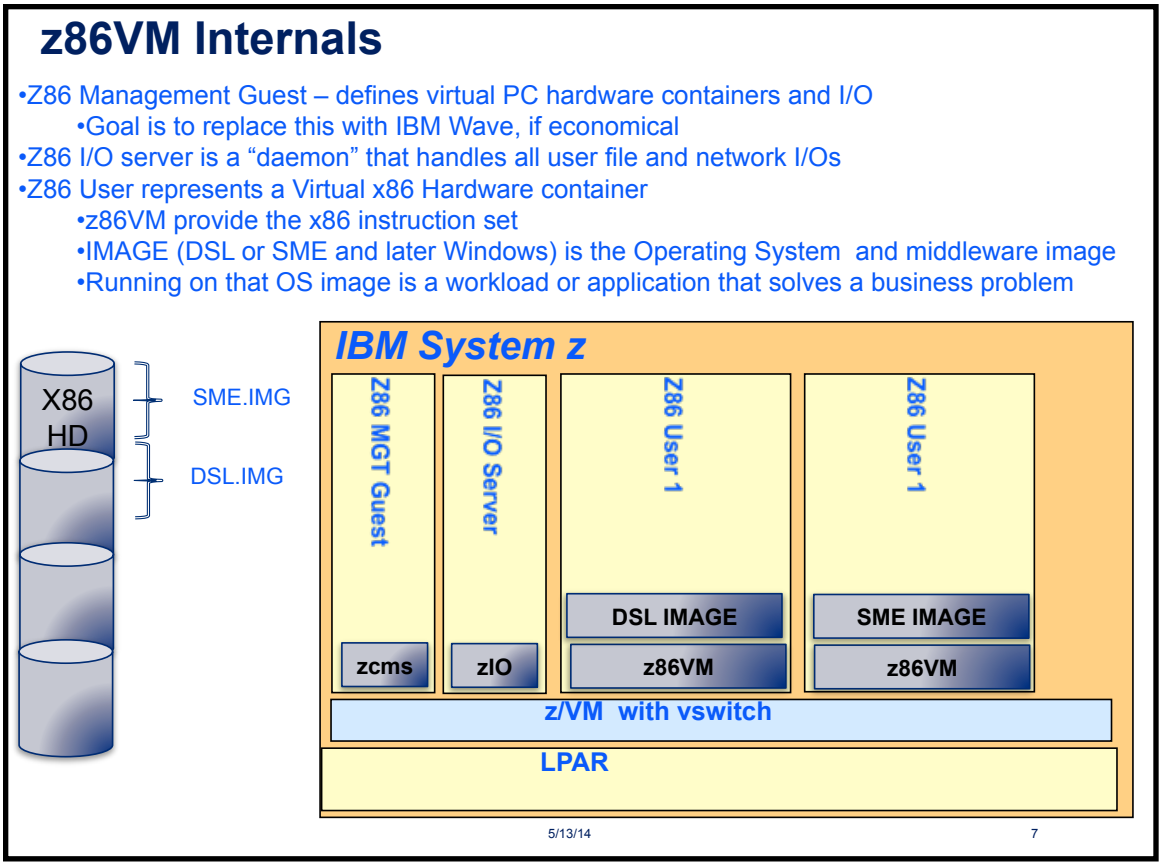
These users may also be automatically started and stopped via admin commands. This is especially helpful when the guests are acting as Daemons or server machines.

Two operations are required for z86 User definitions. First, the z/VM directory must be updated to include the user as a valid virtual machine guest. That will be described in a later section.

Second, those z86 users must be defined to z86VM.

Admin Defined Components of z86VM

The following diagrams are intended to provide a visual overview of the z86VM environment. In this example, we'll use the MGT server, the IO server and two z86 users. We'll also have two x86 defined disks to meet their combined storage needs.

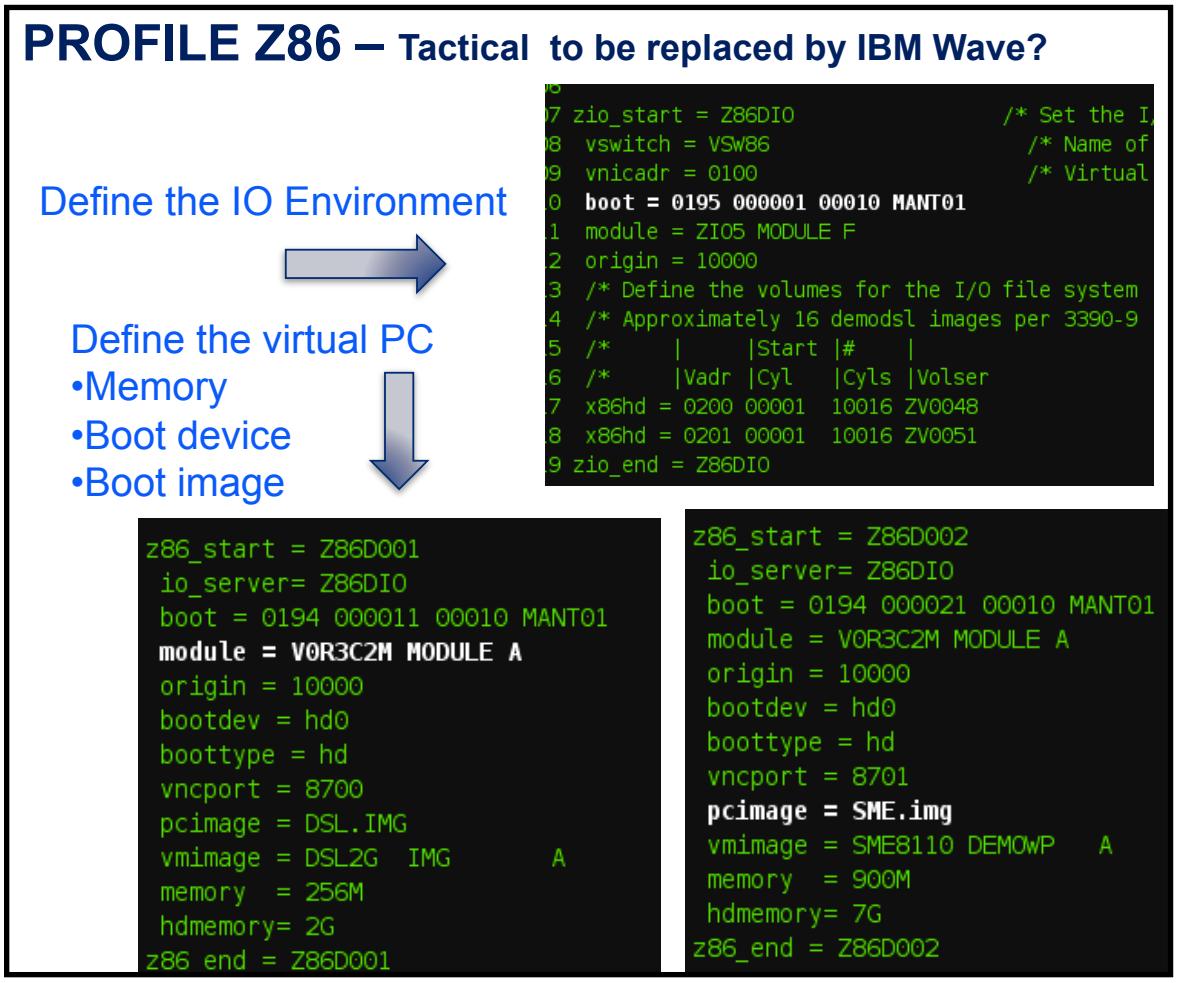


PROFILE Z86 A file

These components are defined within the PROFILE Z86 file. The PROFILE Z86 file[1] is the main administrative initialization file for the z86VM environment. It is comparable to /etc/init, config.sys or SYS1.PARMLIB for UNIX/LINUX, Windows and z/OS operating systems respectively.

[1] Beta 1 and Beta 2 users, please note, the format of the PROFILE Z86 file has changed. You'll need to modify your own profile exec

There are a number of control points that must be pre-established for successful operation of z86VM. The parameters and their potential values follow. In particular, the zIO server and the two z86 users have their specific definitions within these profiles. You might consider that these definitions are really attempting to define the hardware associated with a virtual PC device.



The note above saying that this might be replaced by IBM Wave is something that Mantissa and IBM are considering. Should the PROFILE Z86 become part of the official z86VM product at general availability, we expect an additional level of simplification to be undertaken in the profile definitions. For example, a generic profile could be created for an SME Server and another generic profile for DSL clients. Individual users could then associate themselves with the generic profile and list any exceptions (e.g. different memory or disk drive size).

Processing rules associated with PROFILE Z86

For processing purposes, all text may be in upper or lower case. However, when processed, it will be converted to upper case exclusively.

Unless specified otherwise, only the last instance of a duplicate parameter will be recognized for processing.
Blank lines in the file are allowed and ignored.
Comments are allowed across a single or multiple lines beginning with **/*** and ending with ***/**

Comments on a line that includes a parameter must follow the parameter. For example,
`z86id = z860001  /* this is the first id  */`
is a valid line.
However,
`/* This is the first id  */ z86id = z860001`
is invalid and will result in the line being ignored.

Required parameters are listed in bold.

Parameter	Function and values
	Base z/VM Set up
tcpipname	Name of the z/VM TCP/IP stack that is defined in the z/VM setup. Default: **`tcpipname = tcpip`**
link	This is a z/VM CP link command that can add multiple mini disks to the z86VM environment that may be necessary to host shared EXEC and MODULE files for users. This command can be repeated in the profile. Example: **`Link userid 191 A91 rr`**
access	This is a z/VM CMS access command that associates a linked disk for the purpose of using it to find z/VM related files that are shared across users. This command can be repeated in the profile. **`Example: access A91 E`**
	zIO server set up
zio_start	name of the IO server Example: **`zio_start=z86io`**
vswitch	This is the name of the virtual switch defined under z/VM. Information about setting up the VSWITCH can be found in the ***Appendix A***. Default: **`vswitch = vsw86`**

Parameter	Function and values
vnicadr	a device address number provided by the z/VM admin that created the VSWITCH. Default: v**nicadr = 0100.**
module	Name of the I/O server program module that handles I/O operations and is shipped by Mantissa. This is allowed to vary when a new release comes out. Example: **module = zio5 module a**

x86hd	This can be multiple devices that make up the file system to be used by the IO server. These are logically concatenated together to act like a single storage area network. Example: `x86hd = vadr startcyl #cyl volser`
	`x86hd = 0200 00001 10016 ZV0048`
	`x86hd = 0201 00001 10016 ZV0049`

zio_end	signifies the end of the IO server profile Example: **zio_end = Z86IO**
z86 User Definition	
z86_start	Specify the userid to be defined. Marks the beginning of a z86 user profile Example: **z86_start=DEMODSL**
io_server	name of zIO server to link the z86 user `io_server = z86io`
boot	Specify the z/VM mini disk to boot guests from. Default = 194 Device# startcyl #cyl volser e.g. **boot = 194 100 20 MANT01** **Default: boot = 194**
module	Name of the z86VM program module that handles x86 operations and is shipped by Mantissa. This is allowed to vary when a new release comes out. Example: **module = V0R1C3M MODULE A**
vncport	The initial TCP/IP port number to be used to connect z86VM users to an end user device via VNC service. Example: **vncport = 2000**
vnicadr	a device address number provided by the z/VM admin that created the VSWITCH. Default: **vnicadr = 0100**.
pcimage	x86 image file name. **Example: pcimage = smg.img**
vmimage	z/VM FileName FileType FileMode of image to be loaded to the I/O Server as pcimage. **Example: vmimage = sme8 img a**

Parameter	Function and values
memory	The amount of memory to be associated with a guest virtual machine. A server may require more memory than a Desktop client. Default: **`memory = 900m`**
hdmemory	This signifies the amount of space within the IO file system to be reserved on behalf of this user. Default: **`hdmemory = 6g`**
dbgmachine	Name of a guest virtual machine that is used to debug the z86VM environment. This userid is set during creation of the boot parameters for a z86 user machine. Default: **`dbgmachine = null`**
getimage	z/VM file_name file_type file_mode for an image to be loaded from the I/O Server. Example: **`getimage = new img b`** (Must have enough free minidisk space available to load a compete image)
z86_end	This marks the end of this particular users profile Example: **`z86_end = DEMODSL`**

Sample (minimum) PROFILE Z86 file

```
/* */
zio_start = Z86IO
 vnicadr =  0100
 vswitch =  VSW86
 x86hd =    0200
zio_end =   Z86IO

z86_start =  DEMODSL
 io_server=  Z86IO
 module =  V0R3C4M  MODULE   A
 vncport =  2001
 vnicadr =  0110     /* Add this record */
 pcimage =  dsl.img
 VMIMAGE = DSL0602  IMG      A
z86_end =  DEMODSL

z86_start =  SME8
 io_server=  Z86IO
 module =  V0R3C4M  MODULE   A
 vncport =  1999
 vnicadr =  0100     /* Add this record */
 pcimage =  sme.img
 vmimage =  SME8    IMG   A

z86_end =  SME8
```

Z86 EXEC Command

Functional overview

The Z86 EXEC provides the management capabilities for the z86VM environment. It can start, stop, define and update z86 users and the z/VM shared virtual machines for I/O.

Z86 EXEC operates against the z86 user to define it, create a boot image and its associated mini disks. For the I/O shared virtual machine, it will define, start, update and format the relevant storage devices for this image. For the management of the overall environment, the command will show performance, memory and debug administration capabilities.

		OPTIONS							
		INSTALL	FMTX86	DEFINE=	INITBOOT=	LOAD=	START=	STOP=	ABORT
FUNCTION	Initialize Boot Drive	☆			☆				
	Initialize x86 HD	☆	☆						
	Define x86 Server	☆		☆					
	Load Image	☆				☆			
	Start I/O Server	☆					☆		
	Start x86 Server						☆		
	Stop I/O Server							☆	☆
	Stop x86 Server							☆	☆

Command Syntax of Options

```
z86 parmfile | PROFILE Z86 A | (command options
```

User centric Cmds:

```
Install
FMTx86
Define=  ALL | user
Initboot=ALL | user
initio
Load=    ALL | user
Start= ALL |##|user
Stop=  ALL |##|user
Abort
```

Environment Cmds:

```
Getimage
```

Informational Cmds:

```
Test
WS
Perf
Help
Chart
Status
Log
More
Console
```

USER CENTRIC COMMANDS

Install

The logic is as follows:

1. Formats the boot disks for z86 users.
2. Loads the modules used for the base z86VM functionality and I/O.
3. Provides connection to the zVM predefined TCP/IP stack and identifies the port to be used for VNC connections to z86 users.
4. Creates the I/O file system for the overall z86VM environment and formats the disks
5. Logs on the z86 IO processing guest.
6. Adds any new users to the z86 environment and associates their disk drives to that environment
7. Copies the PC boot image to each individual guest.

FMTx86

CMS formats the defined 'x86hd' I/O disks for the system. Then signs on the I/O Server and performs an FINIT command to format as an x86 hard drive.

Define

DEFINE =ALL | z86userid

Define users and images to the I/O file system; On behalf of the request, the following commands get executed for each user:

ADDUSER AAAAAAAA

Add a user AAAAAAAA to the file system.

ADDFILE USERID TYPE FILEID SECTORSIZE DISKSIZE

Add a file to the file system.

UserID is the Z86 user that the new file belongs.

TYPE is File type: HD/FD/CD as specified in the PROFILE Z86.

FILEID is name of file *(up to 200 CHARS)*. This is the pcimage defined in PROFILE Z86.

SECTORSIZE of 512 for HD/FD; 2048 for CDROM. Can be overridden in the PROFILE Z86.

DISKSIZE is size of disk These definitions are from the 'z86 Reference Card' specified in KBytes, MBytes, GBytes. This size is defined by hdmem in the PROFILE Z86.

Examples:

ADDFILE USER2 HD DISK.IMG 512 200M Defines a 200mb Hard Disk.

ADDFILE USER2 FD FLOPPY.IMG 512 144K Defines a 144kb Floppy Disk.

A**DDFILE USER2 CD CDROM.IMG 2048 100M** Defines a 100mb CDROM

ADDBOOT AAAAAAAA XXXXX

Adds boot information for User AAAAAAAA

XXXXX is the boot information. These options are specified in the PROFILE Z86.

Example:

ADDBOOT USER4 MEM=2G FDA=M32.IMG BOOT=FDA TYPE=16

INITBOOT ALL | server id

For all I/O and x86 Servers or individual servers.

Profile definitions are used to build the boot disk, which by default is the 194 for x86 servers and 195 for I/O servers. The 'module' defined in the Profile is copied to the boot disk. Boot information for an I/O server includes 'module' and 'origin'. Additional boot information for an x86 server includes: 'io_server', 'tcpip', 'vncport' and 'vnicadr'.

InitIO

INITIO - Auto starts all I/O servers and then finit's the I/O file system volumes

Load

LOAD - (Performs a ZVCOPY to pc image) Copies the master x86 operating system image, 'vmimage', from z/VM CMS to the specified x86 user id's PC image space as 'pcimage'. The z86 user ids and the PC image are listed in the PROFILE Z86 file. If different operating sys-

tems will be loaded to a variety of different z86 users, then separate PROFILE Z86 files can be utilized to isolate operations to specific userids. **Important Usage Note:** The z86 user must be logged off when processing this command. The zIO server must be logged on or disconnected.

Start

START - ALL | ## | user XAUTOLOG the z86vm machines;
= the first ## defined in the Profile z86.
Works for both zio and z86 users

Stop

STOP - ALL | ## | user FORCE a z/VM logoff of the z86vm machine(s);
= the first ## defined in the Profile z86.
Note: STOP will ask for a "YES" confirmation to complete the STOP as follows:
When an x86 server request is made: "Are you sure you want to force xxx before you perform a shutdown? The image may be damaged."
When an I/O server request is made: "Are you sure you want to force I/O Server xxx? If a YES reply is not entered, the STOP operation is skipped.

Abort

Abort - FORCE a z/VM logoff of all x86 and I/O servers;
ABORT will ask only once for a "YES confirmation: Abort may destroy data and require an install. Answer YES to ABORT or NO to skip".

ENVIRONMENTAL COMMANDS

Getimage = z86user

Runs zvcopy CMS: copy from PC image, 'pcimage' to zVM file, 'getimage'. This is one way to create a backup of the PC operational environment. **Important Usage Note:** The z86 user should be shutdown and logged off when processing this command. The zIO server must be logged on or disconnected.

INFORMATIONAL COMMANDS

Test

TEST - List values from the parameter file, whose default is PROFILE Z86 A;

WS

WS provides the working set information associated with zVM guests being used with z86VM. At present, this is being used for the development and performance improvement of z86VM. It is not expected to be a customer option when the product becomes generally available.

Perf

PERF - This command provides performance data in the system log. It can be processed on its own. It is also included within the WS option.

'WS' and 'PERF' are primarily for z86VM development benefit to check z/VM utilization. Paging was a big consideration before we got the z114 and IFL's. We could add any commands we need.

parm.z86.0 contains the number of z86id entries from the PROFILE Z86.
Example, if PROFILE Z86 had
z86id = DEMODSL
z86id = SME8
parm.z86.0 = 2 and
parm.z86.1 = 'DEMODSL' and parm.z86.2 = 'SME8'

'WS' would issue '`cp ind user DEMODSL`' and '`cp ind user SME8'`.

'WS' Results

```
I/O volumes configured: 1
```

```
z86VM machines configured: 1
TCP/IP Stack: TCPIP
TCP/IP Port: 8760
-------------------------
DEMODSL PAGES: RES=00012070 WS=00011793 LOCKEDREAL=00000000 RESVD=00000000
DEMODSL CPU 00: CTIME=00:00 VTIME=000:24 TTIME=000:25 IO=000574
```

'PERF' Results

```
AVGPROC-056% 02
XSTORE-000000/SEC MIGRATE-0000/SEC
MDC READS-000001/SEC WRITES-000001/SEC HIT RATIO-083%
PAGING-1/SEC STEAL-000%
Q0-00001(00000)                          DORMANT-00034
Q1-00000(00000)           E1-00000(00000)
Q2-00001(00000) EXPAN-001 E2-00000(00000)
Q3-00002(00000) EXPAN-001 E3-00000(00000)
PROC 0000-037% IFL      PROC 0001-075% IFL
LIMITED-00000

                EXTENT     EXTENT  TOTAL  PAGES   HIGH    %
VOLID  RDEV      START        END  PAGES IN USE   PAGE USED
------ ---- ---------- ---------- ------ ------ ------ ----
M01P10 1332          0      10016  1761K      0      0   0%
M01P09 1331          0      10016  1761K      0      0   0%
M01P08 1330          0      10016  1761K      0      0   0%
M01P07 1231          0      10016  1761K      0      0   0%
M01P06 1230          0      10016  1761K      0      0   0%
M01P05 1131          0      10016  1761K      0      0   0%
M01P04 1130          0      10016  1761K      0      0   0%
M01P03 1031          0      10016  1761K      0      0   0%
M01P02 1030          0      10016  1761K     12     12   1%
M01P01 1002          1      10016  1761K      0      0   0%
                                   ------ ------        ----
SUMMARY                            17608K     12          1%
USABLE                             17608K     12          1%

DEMODSL PAGES: RES=00020073 WS=00020063 LOCKEDREAL=00000000 RESVD=00000000
DEMODSL CPU 00: CTIME=00:01 VTIME=000:51 TTIME=000:52 IO=000574
```

Help

This lists all of the options available within the Z86 EXEC.

Chart

This will provide a z/VM equivalent of the functional chart demonstrated above in the Z86 EXEC functional overview

Status

This will query the sign on status of all zio and z86 users.
Possible outcomes are:

- user disconnected
- user signed on
- no information provided for logged off users.

Z86 EXEC options:

CONSOLE

Create a console log of command execution.

LOG

Create a log file of command execution.

MORE

Create more detail information in the console and/or log file of command execution.

Changing the Z86 EXEC for local usage

Because this is a REXX exec, it is easily modifiable for local usage. At this time, we recommend that customers do not modify this exec or minimize changes to it. Mantissa still has a list of changes that it expects to make prior to general availability of the product. We also anticipate changes will be made within the PROFILE Z86, but that should be an easier migration.

Within Mantissa, each developer/tester, has their own z86 environment with different z86 Management servers, zIO servers and z86 user names. These must be unique names across the z/VM environment. These changes can be identified within the PROFILE Z86 that each user has.

However, in a customer environment, there may only be two z86 environments, one for production and one for development and/or test. At this point, Mantissa is not recommending Z86 for a production environment. But when running as such, the development and production environments, while running within the same z/VM image, should have separate management and IO servers defined.

Use Cases for Z86 EXEC

Making a backup of a z86 user's current PC image

use **`z86 (getimage=z86user`**

Adding a user

- create the user definition within your PROFILE Z86 file
- use **`z86 (define=z86user`**

Change the operating system of a user

- make sure the user is logged off.
- Modify the user profile with PROFILE Z86 to change the vmimage to the system to be loaded
- **`z86 (load=z86user`**

Important z86VM Program Modules and their functions

ZIOx MODULE (Beta 3: ZIO5 MODULE)

This provides the I/O functionality that translates X86 FBA requests into z specific storage.

ZVCOPY MODULE

This enables copying between a z/VM CMS guest virtual machine and a z86 guest uservirtual machine. See the ZVCOPY command above for the appropriate syntax and options associated with the command.

V0RxCyM MODULE (Beta 3: V0R3C4M MODULE)

This provides the main functionality for the operation of the z86VM environment within a z86 guest user. Each user has a copy of this module loaded for translating x86 binaries to the z architecture, handling interrupts, managing process modules and mapping the x86 memory to the z architecture's memory.

Z86VMB3 VMARC

This is a z/VM backup file that includes all the other z86VM files. In PC or Linux terms, this is like a z specific ZIP or TAR file for aggregating a lot of other files. It is expanded during the INSTALL process.

ZVCOPY Command

ZVCOPY is used to copy a z86VM file between a CMS user and the x86 filesystem.
Important Usage Note: The z86 user MUST BE LOGGED OFF.
The zIO Server must be logged on or Disconnected.
The syntax is as follows:

`ZVCOPY z86vmIO USERID TARGET PCFILE FN FT FM`

z86vmIO = VM ID OF z86vm to I/O machine XXX.YYY.
TARGET = PC or CMS.
PC implies you are copying from CMS to the z86VM file system.
CMS implies you are copying from the z86VM file system to CMS.
PCFILE = PCFILE name to Create/Read. From the pcimage in PROFILE Z86
FN/FT/FM = CMS file name. This can be any file name.

The ZVCOPY command, when copying an operating system image, could take several minutes to complete. Consider this the full FTP of an operating system and it will make sense to most administrators. We tell you this so that you don't abandon the operation prematurely.

Using the z86VM Console

When a z86 user is booted up as a z/VM guest, the z/VM environment is capable of processing a number of z86VM commands, as well as CP commands. The following is a list and description of the available z86VM commands.
These commands are also listed on a reference card available at www.mantissa.com

MAP	MAPS STORAGE
CHECK STORAGE	CHECKS GETMAIN STORAGE CHAINS
Q TASK	DISPLAY ACTIVE TASKS
Q ECB	DISPLAY WAITING ECBS
Q X86	TEST X86 STORAGE FOR AVAILABILITY
REPLID	DISPLAY OUTSTANDING OPERATOR MESSAGES
/	PASS COMMAND TO MMU
STOP	STOP ALL TASKS
Q CONN	DISPLAY CONNECTIONS TO THIS VM MACHINE
DEBUG -1	DISPLAY ORIGINAL ABEND ON RECURSIVE ABEND
DEBUG	DISPLAY LAST ABEND
KILL	KILL X86 ENVIRONMENT
SHOW COMAREA	DISPLAY COMMON AREA
SHOW TICKS	DISPLAY TICK COUNT
SHOW WAIT	DISPLAY NBR OF TIMES IN WAIT
SHOW INT	SHOW X86 INTERRUPT STATUS
SHOW KEYTAB	DISPLAY ADDRESS OF KEYBOARD TABLE BUFFER
SHOW KEYS	DISPLAY PENDING KEYS IN KEYBOARD BUFFER
SHOW LASTKEY	DISPLAY LAST 100 KEYS SENT TO X86
SHOW X86ADR	DISPLAY REAL ADDRESS FOR X86 ADDRESS SPACE
SHOW ZCPUSERA	DISPLAY ADDRESS OF ZCPUSER AREA
X86	PASS COMMANDS TO X86 DEBUGGER
CLEAR TIMEOUT	CLEAR TIMEOUT VALUE
CLEAR TICKS	CLEAR X86 TICK COUNTER
TRACE DISK OFF/ON	TURN OFF/ON DISK I/O TRACING
TRACE FP OFF/ON	TURN OFF/ON FLOATING POINT TRACING
TRACE PAGEFAULT OFF/ON	TURN OFF/ON PAGE FAULT MESSAGES FOR X86
TRACE INST OFF/ON	TURN OFF/ON X86 INSTRUCTION TRACE TO 00F
TRACE INT OFF/ON	TURN OFF/ON X86 INTERRUPT TRACE
TRACE KEY	START X86 KEYTRACE
TRACE MOUSE OFF/ON	TURN OFF/ON MOUSE EVENT TRACING
TRACE PORTS OFF/ON	TURN OFF/ON GENERAL PORT TRACING
TRACE VIDEO OFF/ON	TURN OFF/ON VIDEO PORT TRACING
TRACE PCI OFF/ON	TURN OFF/ON PCI PORT TRACING
TRACE CACHE OFF/ON	TURN OFF/ON X86 INSTRUCTION CACHING

VIDEO SAVE OFF/ON	TURN OFF/ON VIDEO SAVE MODE
VIDEO TILE OFF/ON	TURN OFF/ON VIDEO TILEBOX MODE
INT OFF/ON	TURN OFF/ON X86 INTERRUPTS
SET DEBUG OFF/ON	TURN OFF/ON X86 DEBUG MODE
SET IRQ1 ON	TURN ON IRQ1
SET IRQ6 ON	TURN ON IRQ6
SET IRQ14 ON	TURN ON IRQ14
DUMP TRACE	DUMP TRACE FILE TO 00E
DUMP PCI	DUMP PCI CONFIG TABLES
DUMP VIDEO	DUMP ENTIRE TEXT VIDEO CONTENTS
LOG ???	WRITE ??? CONTENTS TO TRACE FILE
VNC RESET	IF VNC WON'T RECONNECT, ISSUE THIS AND TRY AGAIN

Installation of z86VM Product

Beta 3 Distribution

The Beta 3 distribution of z86VM demonstrates its capability to execute both an x86 SME Server image and a DSL 2012 client image on an IBM z Series processor. The distribution is based on Linux kernel level 2.6 for SME and 2.4.3 for DSL.

Introduction
The z86VM Beta 3 installation procedure is executed from the installation z/VM guest, Z86MNT. The Z86 EXEC is used to install the z86VM Beta 3 files. After installation, a 3270 Emulator can be used to log on the z86VM I/O server, Z86IO, and the z86VM machines, DEMODSL or SME8. The I/O server, z86IO, should always be logged on first before logging on the z86 machine, DEMODSL or SME8.

Files delivered via the Mantissa FTP site, CD or thumb drive

- z86vmb3.pdf — This document.
- readme.txt — A document with additional install and test information.
- z86vmb3.direct.txt — A copy of z/VM directory entries for z86VM guests (Z86MNT, Z86IO, DEMODSL and SME8)
- z86vmb3.vmarc — z86VM files to be installed on z/VM user id Z86MNT.
- tightvnc-2.0.2-setup.exe — TightVNC Self-installing package for Windows.

z86VM Requirements

The following facilities are required to install and run this distribution:

- An IBM System z machine (must be a z114 or z196 processor or newer) running z/VM Version 6.2 or later
- An OSA adapter to configure a Layer 2 VSWITCH
- A minimum of 2 - 3390-9 DASD or equivalent (20,000 cylinders)
- Available TCP/IP Ports
- Access to VMARC
- z/VM SMSG and IUCV accessibility
- 3270 Emulator

Installation Steps

There are several steps necessary for total installation and successful operation of z86VM and a guest operating system on top of that. This section provides the steps to do that from scratch. Sample definitions and Profiles are included to speed a system administrator toward completion of the tasks.

1. Download "Beta 3" contents from Mantissa
2. Setup z/VM VSWITCH
3. Define z86 user ids as listed in z86vmb3.direct.txt
4. Install the Z86MNT user id
5. Install the z86 servers via Z86 EXEC
6. Start the z86VM guest images via the Z86 EXEC

1. Download "Beta 3" contents from Mantissa

** New note: Experience has told us that you need to initiate the following FTP from a Windows or Mac system and then locally update it to your zVM system. There seems to be an issue with the FTP Client on zVM that is inhibiting a successful download. Once downloaded to the desktop system, from that system, open a connection to your zVM and PUT the files there.

```
ftp  ftp.mantissa.com <http://ftp.mantissa.com>
```

user: (Insert userid received in email)

pw: (insert case sensitive password received in email)

```
ascii
get  z86vmb3.direct.txt
bin
get  tightvnc-2.0.2-setup.exe     <== if you need a VNC Client
get  z86vmb3.vmarc
quit
```

2. Setup z/VM VSWITCH

z86VM requires a z/VM Layer 2 VSWITCH. Refer to ***Appendix A*** for an example of selecting an OSA adapter and configuring a VSWITCH. The name of the VSWITCH, VSW86 in the example, should be referenced in the directory definition in the following step.

3. Define z86 user ids

The downloaded file, z86vmb3.direct.txt contains the directory entries for the required z86 user ids. Contact your z/VM System Administrator to modify and install the directory entries into the z/VM USER DIRECT. The sample directory entries must be configured with:

- your VSWITCH name, defined above;
- a unique macid;
- minidisk <start_cyl> and <volser> information.

** Updated 06/05/14
We've gotten some requests to provide some sample guest sizes. In the Mantissa development environment, these are some of the set up we use: :

Each x86 ID is defined with 6 Gig memory and a 10 cyl boot drive. (The 194 drive)

The I/O ID is defined with 9 Gig memory and a 10 cyl boot drive. (The 195 drive) The x86 hard drive space (UDM) is defined on the I/O ID. Beta 3 defines it as 10k Cyls.

The maint ID is defined with 1 Gig memory and a 10k Cyl A-disk.

**Updated 06/05/14
We found a problem where different operating system images might have network conflicts because they required a fixed MAC address. This requires changes to the definition of the IO server and in the PROFILE Z86 for each user that requires a fixed MAC address.

In the USER DIRECT for the I/O machine define and associate a virtual NIC address with a fixed MACID.
Example:

```
NICDEF 0100 TYPE QDIO LAN SYSTEM VSW86 MACID FFF100
NICDEF 0110 TYPE QDIO LAN SYSTEM VSW86 MACID FFF110
NICDEF 0120 TYPE QDIO LAN SYSTEM VSW86 MACID FFF120
```

In the PROFILE Z86 assign the virtual NIC address to the x86 machine using vnicadr:
Example:

```
z86_start =  SME8
```

```
io_server=  Z86IO
module =  V0R3C4M  MODULE   A
vnicadr =  0100
vncport =  1999
pcimage =  sme.img
vmimage =  SME8    IMG   A
z86_end =  SME8
```

When vnicadr is not specified, the default is nicadr = 0100

The NIC Address and Port are included on the Stand-Alone Program Loader(SALIPL) IPLPARMS for the boot drive of the z86VM userID.

Note:
Under z/VM, the first 3 bytes of the MAC address(MACPREFIX) are unique for each VM image in your network. By default these bytes are set by z/VM to be x'02-00-00'. The x'02' is fixed but the remaining two bytes may be changed to create a unique MAC address domain for your VM image. The MACPREFIX, is modifiable in the System Config file. The MACID defined in the NICDEF above, is appended to the MACPREFIX to create a unique MACID for the network. Example: '020000FFF100' (see z/VM: CP Planning and Administration for more information).

**Added Jun6 15, 2014

If you have only 3390 model 3, add 2 additional 3390 and configure as follows:

Update USER DIRECT

```
USER Z86IO   Z86IO   9G  20G  G
 INCLUDE Z86PROF
 OPTION MAXCONN 2000
 NICDEF 0100 TYPE QDIO LAN SYSTEM VSW86 MACID FFFF02
 NICDEF 0110 TYPE QDIO LAN SYSTEM VSW86 MACID FFF202
 MDISK  0195 3390 <start_cyl> 00010 <volid> MR Z86IO Z86IO Z86IO
 MDISK  0200 3390 <start_cyl> 3339   <volid> MR Z86IO Z86IO Z86IO
 MDISK  0201 3390 <start_cyl> 3339   <volid> MR Z86IO Z86IO Z86IO
 MDISK  0202 3390 <start_cyl> 3339   <volid> MR Z86IO Z86IO Z86IO
```

Update PROFILE Z86

```
zio_start = Z86IO
vswitch =  VSW86
x86hd =    0200
x86hd =    0201
x86hd =    0202
module = ZIO5 MODULE A
zio_end =   Z86IO
```

Then follow install instructions.

4. Install the Z86MNT user id and z86VM product

Log on to the Z86MNT User Id :

1. Format the A-disk
   Enter: **FORMAT 191 A**
   Enter: **1**
   Enter: **MNT191**

2. Upload the z86vmb3.vmarc file to the Z86MNT 191 minidisk as Z86vmb3 VMARC A in FIXED BINARY LRECL 80.

3. Perform a LISTFILE to verify the format.
   Enter: **LISTFILE * * a (d**

```
FILENAME FILETYPE FM FORMAT LRECL
Z86VMB3  VMARC    A1 F      80
Ready;
```

4. If Z86VMB3 VMARC A is not FORMAT=Fixed and LRECL=80 then execute the following to reblock:
   Enter: **PIPE < Z86VMB3 VMARC A | FBLOCK 80 00 | > Z86VMB3 VMARC A F 80**

5. Unpack the Z86VM VMARC file:
   Enter: **vmarc unpk z86vmb3 vmarc a**

```
DSL      IMG      A1. Bytes in= 294153360, bytes out= 471859200 (   160%).
PROFILE  EXEC     A1. Bytes in=       800, bytes out=      3120 (   390%).
PROFILE  Z86      A1. Bytes in=     26880, bytes out=    618320 (  2300%).
ZIO5     MODULE   A1. Bytes in=     79440, bytes out=    257211 (   323%).
ZVCOPY   MODULE   A1. Bytes in=     43840, bytes out=    118984 (   271%).
V0RxCyM  MODULE   A1. Bytes in=    500000, bytes out=   2042241 (   408%).
```

```
Z86      EXEC     A2. Bytes in=     21520, bytes out=    168000 (   780%).
Z86VMB3  DIRECT   A1. Bytes in=     13120, bytes out=    169760 (  1293%).
Ready;
```

6. Perform a LISTFILE to verify the unpacked files:
   Enter: **`LISTFILE * * a (d`**

```
FILENAME FILETYPE FM FORMAT LRECL
Z86VMB3  VMARC    A1 F         80
DSL      IMG      A1 F        512
PROFILE  EXEC     A1 F         80
PROFILE  Z86      A1 F         80
ZIO5     MODULE   A1 V      65535
ZVCOPY   MODULE   A1 V      65535
V0RxCyM  MODULE   A1 V      65535
Z86      EXEC     A2 F         80
Z86VMB3  DIRECT   A1 F         80
Ready;
```

7. IPL ZCMS to execute the PROFILE EXEC:
   Enter: **`IPL ZCMS`**

```
DMSACC724I 19E replaces Y (19E)
DMSACP723I Y (19E) R/O
z/CMS V6.2.0 2011-10-07 10:29
DMSWSP100W Shared Y-STAT not available
Ready;
```

8. XEDIT PROFILE Z86 A, and make the following changes as your z/VM TCP/IP environment requires:

```
tcpipName =  TCPIP   /* Name of your z/VM tcp/ip stack */

vncport =  2001      /* Unique tcp/ip port for DEMODSL */

vncport =  1999      /* Unique tcp/ip port for SME8 */
```

5. Install the z86 servers via the Z86 EXEC

Install the z86 IO users and z86 users via Z86 EXEC

The installation of the servers will take some time since all the dasd will be formatted and the x86 disk space will be initialized. This step should only be performed once when a good completion is achieved. Installing users is as simple as:

From Z86MNT, Enter: **`Z86(INSTALL LOG MORE`**

Appendix B contains a sample Install Log.

This completes the z86VM Beta 3 install.

6. Starting and Stopping z86VM guest virtual machines

Starting z86 servers

To start nnn z86 Servers, from the Z86MNT User Id :

Enter: **Z86 (START=nnn LOG**

To start all installed z86 Servers, from the Z86MNT User Id :

Enter: **Z86 (START=ALL LOG**

To start a single z86 Server, from the Z86MNT User Id :

Enter: **Z86 (START=z86user LOG**

An example of the console follow

```
AUTO LOGON  ***       Z860001  USERS = 38
yyyy-mm-dd hh:mm:ss Z860001 server started; VNC Port: 2001.
.
.
.
AUTO LOGON  ***       Z8600nnn  USERS = 39
yyyy-mm-dd hh:mm:ss Z860nnn server started; VNC Port: 2nnn.
  yyyy-mm-dd hh:mm:ss Z86 START command ending...
```

Stopping z86 Servers

Syntax of the command is the same as the start command. To stop all z86 Servers, from the Z86MNT User Id :

Enter: **Z86 (STOP=ALL LOG**

An example of the console follow

```
yyyy-mm-dd hh:mm:ss Z860001 server stopped.
.
.
.
yyyy-mm-dd hh:mm:ss Z860512 server stopped.
yyyy-mm-dd hh:mm:ss Z86 STOP command ending...
```

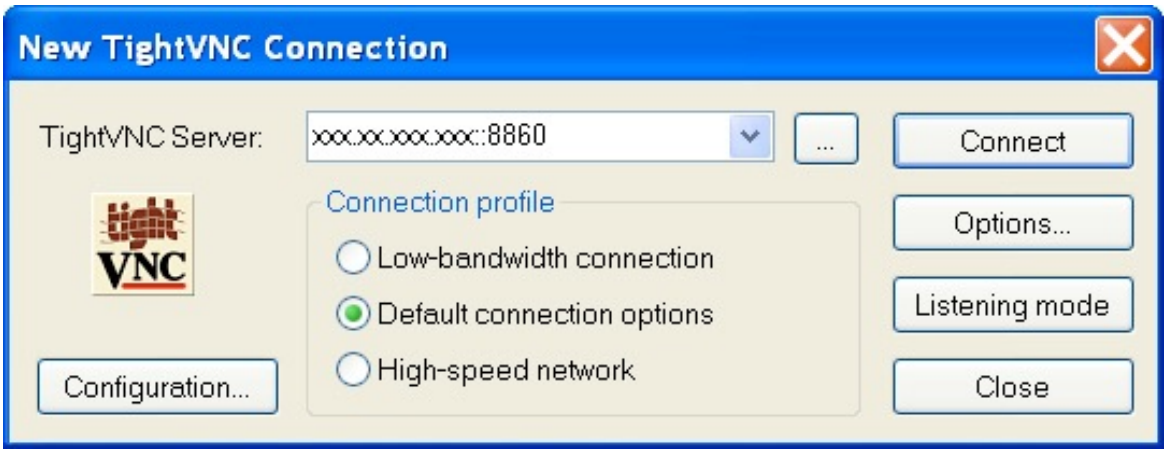
7. Running the DEMODSL

Enter: z86 (start=demodsl This isn't necessary if you just issued z86 (start=all

Start a VNC Client with your installation's z/VM TCP/IP IP address and the port number recorded in the PROFILE Z86 file (8, in the Installation Process) for DEMODSL.

Start a VNC Client with your installation's z/VM TCP/IP IP address and the port number recorded in the PROFILE Z86 file for DEMODSL (8, in the Installation Process).

The TightVNC Client for Windows is provided in the event a VNC Client is not available at the install site. The use of TightVnc may cause periodic disconnects of the client to the vncserver in z86VM. If this happens - simply reconnect to the server. 'Chicken of the VNC' as well as 'JollysFast' VNC (for OSX) work without this issue.



Linux will boot and a DSL desktop will appear:

Record the IP Host address displayed on the right side of the panel (172.16.200.110 in this example).

Start a second VNC Client with this address and port 5900 (172.16.200.110:5900 in this example).

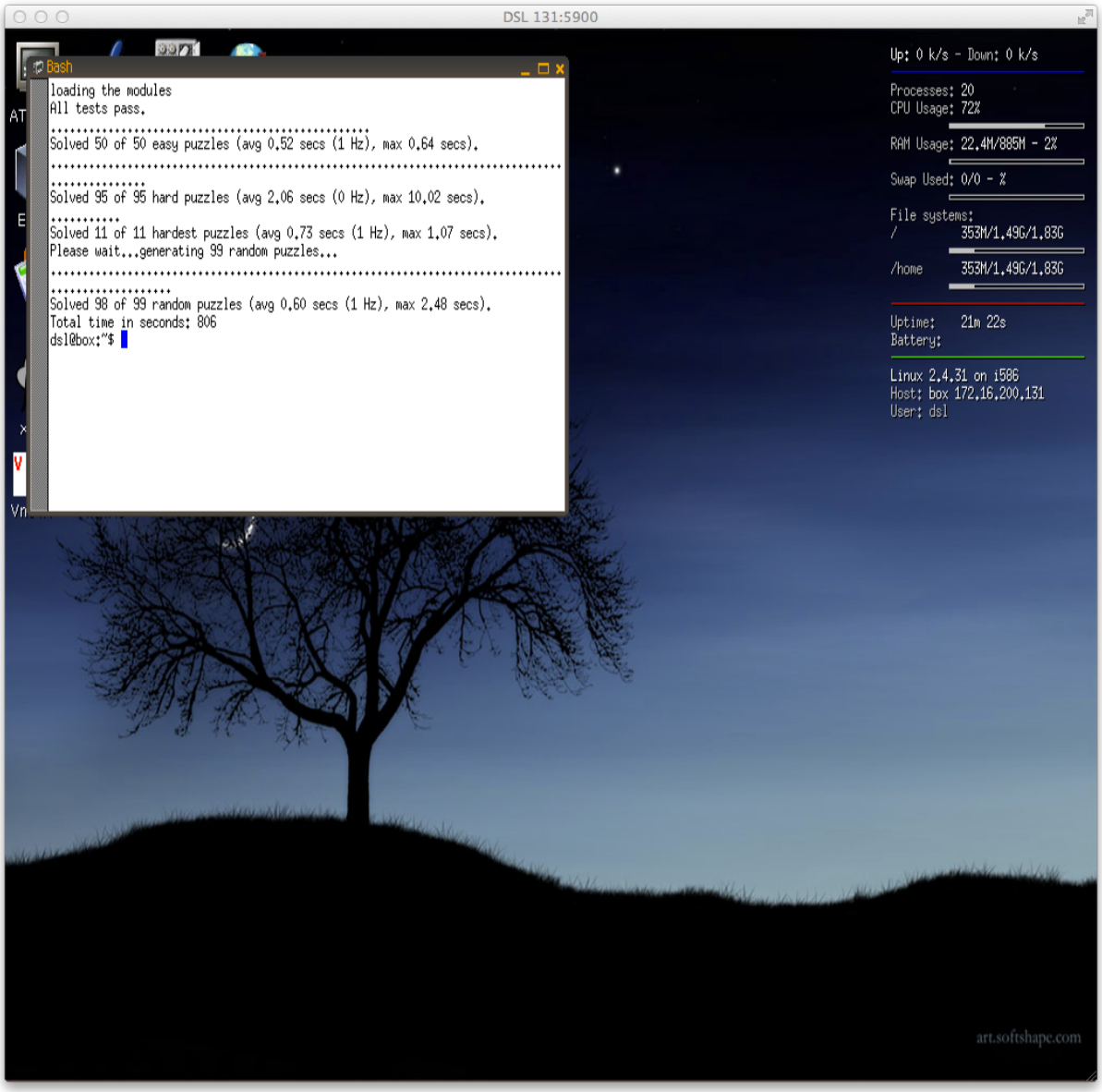
A second desktop will appear. Close the first VNC Client.

Running The Performance Script

From the DSL desktop, Select DSL-XShells-Light to get a terminal panel.

Enter: ‘./go.sh’ to run the sudoku performance test.

The sudoko test program will run to completion. Record the test script duration (i.e, “Total Time in seconds:”). This is a performance metric that should be returned to Mantissa.

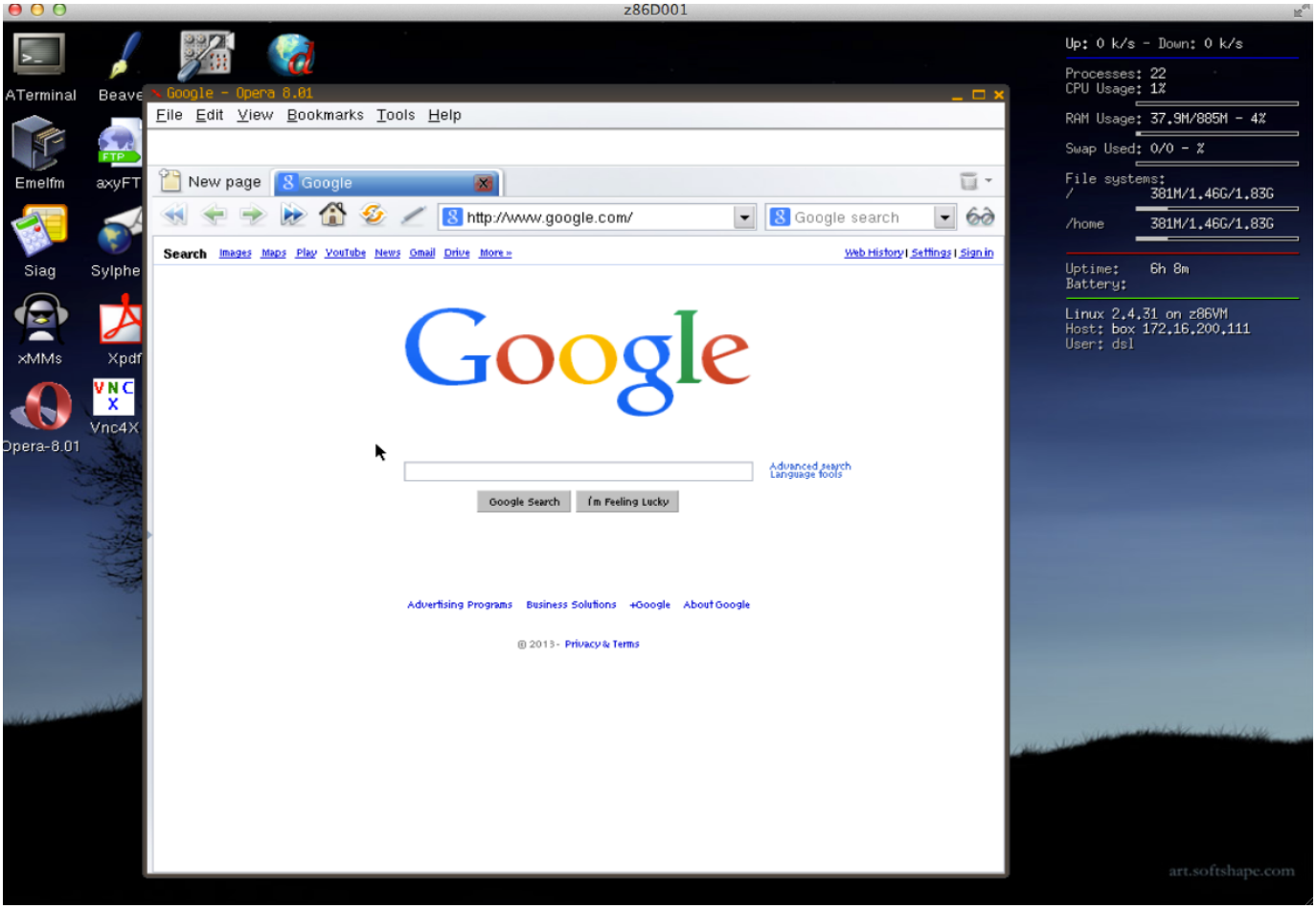


Try the Opera browser

** Added this section on 6/5/14.

As mentioned in the Known Problems section, the Firefox browser is not currently working with the Beta 3 systems. However, the Opera browser does work. The icon is to the lower left of the icons on the DSL Desktop, signified by the large O.

When you mouse click on this, a browser will open and the default page is currently set to google.com. Here's a view of a successful opening.



8. Running the SME8

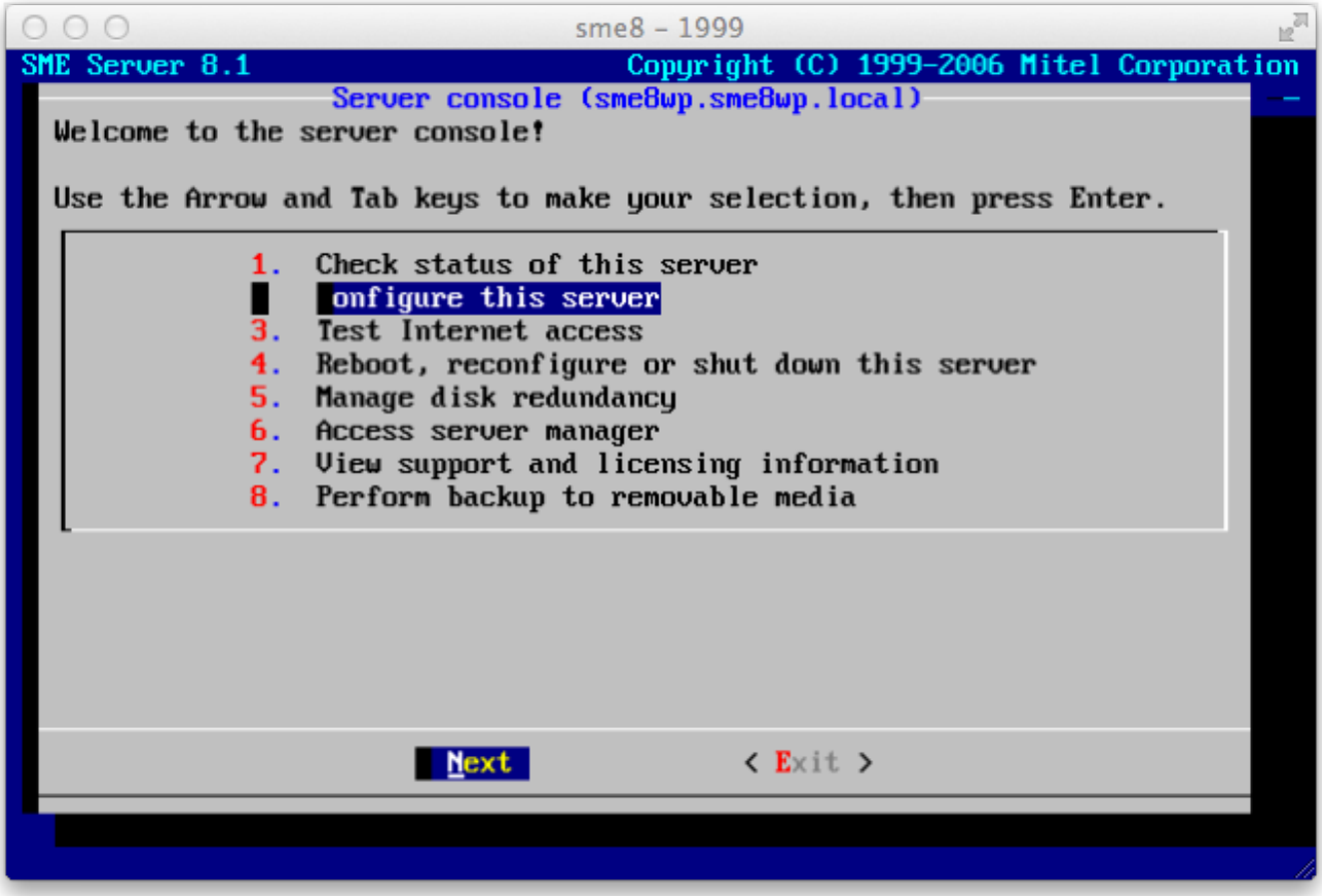
Enter: z86 (start=sme8 This isn't necessary if you just issued z86 (start=all

Start a VNC Client with your installation's z/VM TCP/IP IP address and the port number recorded in the PROFILE Z86 file (8, in the Installation Process) for SME8.

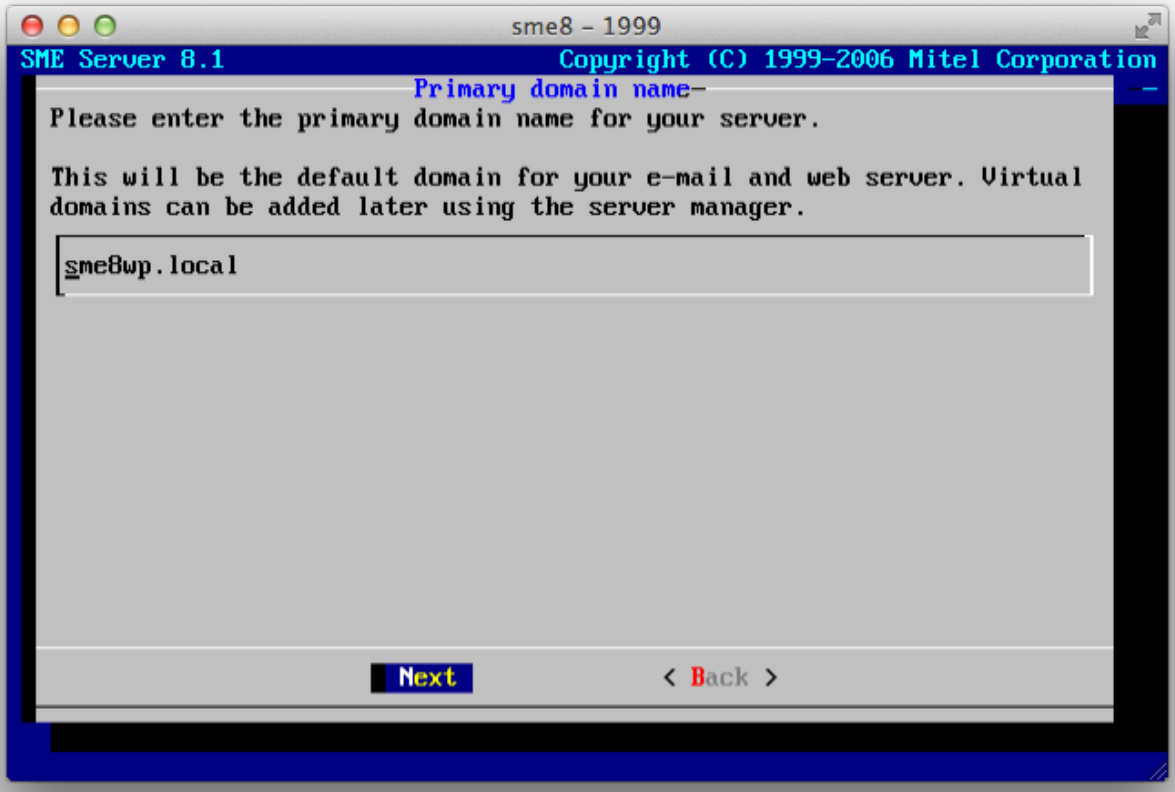
It's important to watch boot up the first time you start a new x86 operating system image to ensure there are no boot up errors. See ***Appendix C: Trouble Shooting*** for some assistance. One area we've seen is when the MAC address for network is incorrect. Otherwise, if you get a system prompt with everything prior to that as OK, proceed with the following.

SME8 will boot and present a login panel. Log in as 'admin' with password of 'Noodle01'.

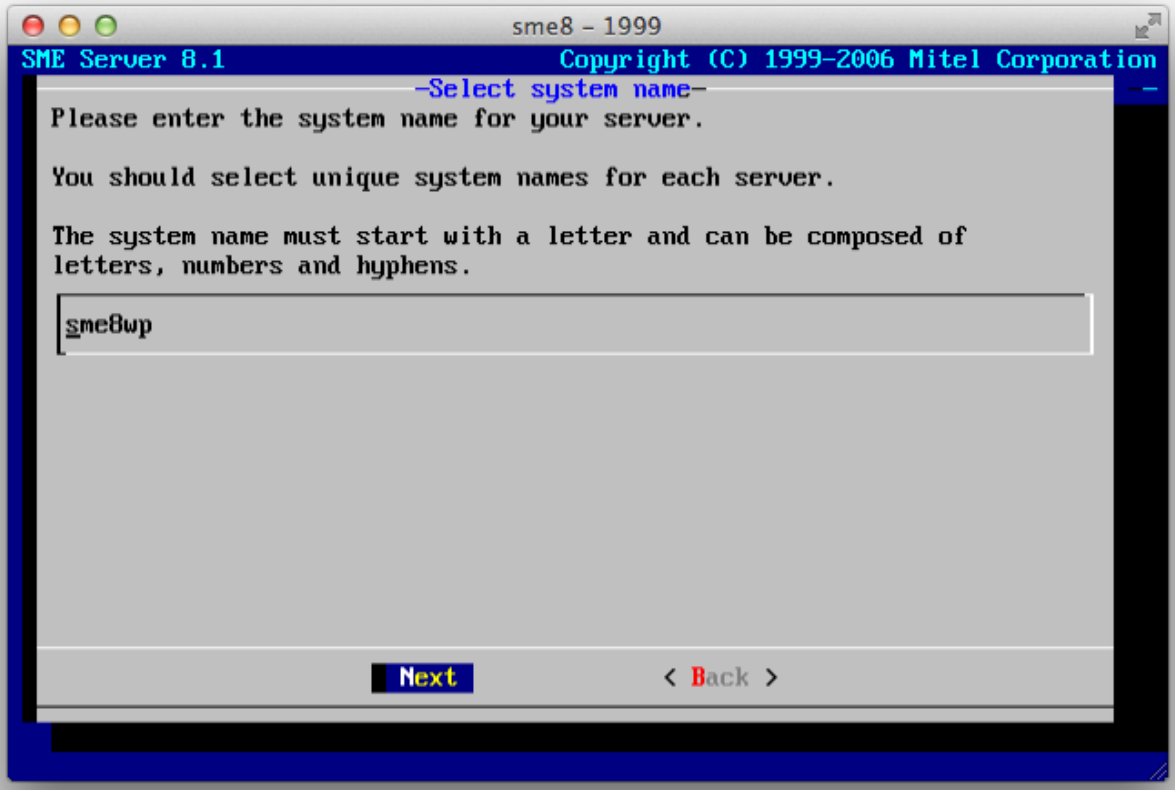
When the server console is presented, select “2. Configure this server” and “Next”.



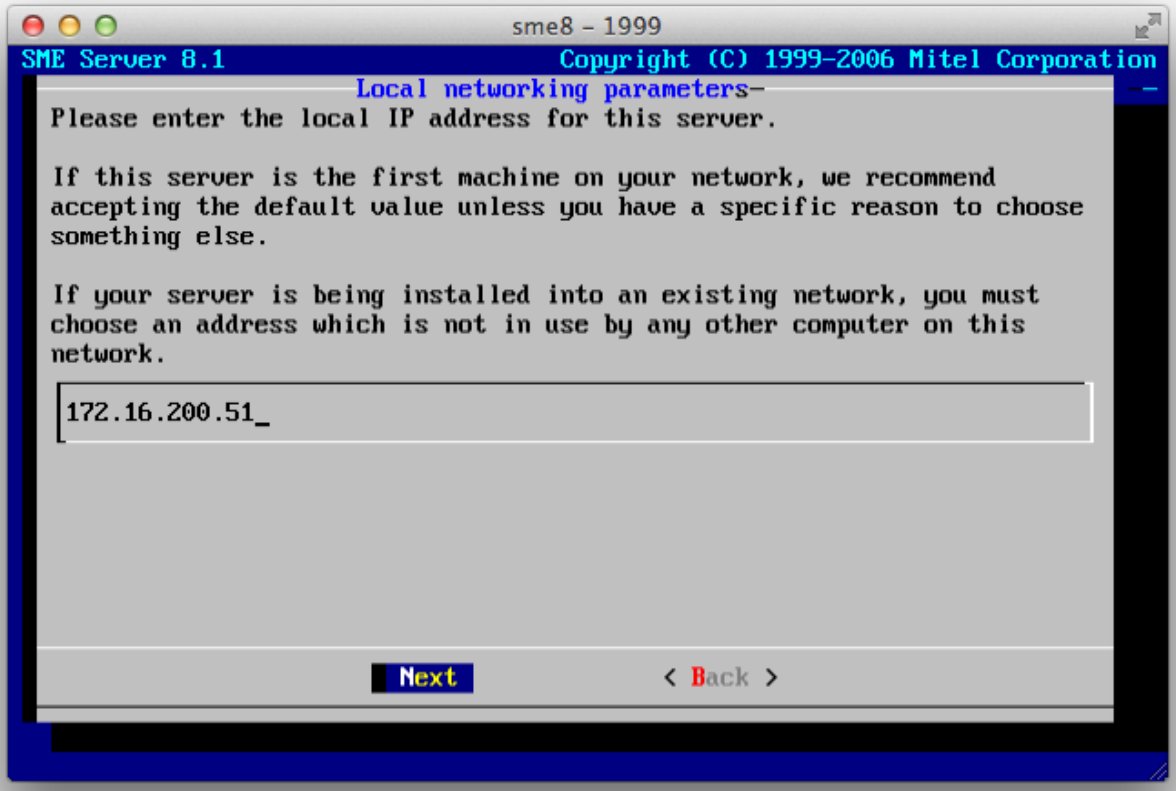
Change the primary domain name or accept the default and “Next”.



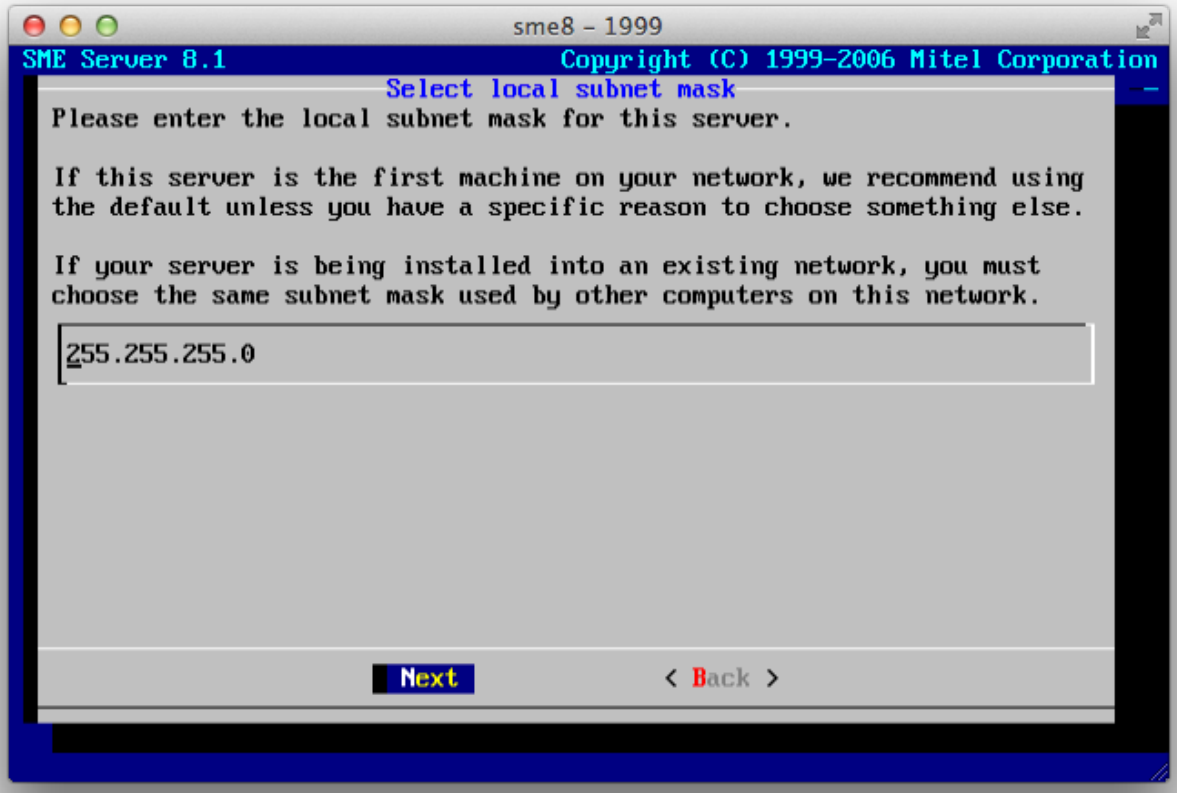
Change the system server name or accept the default and “Next”.



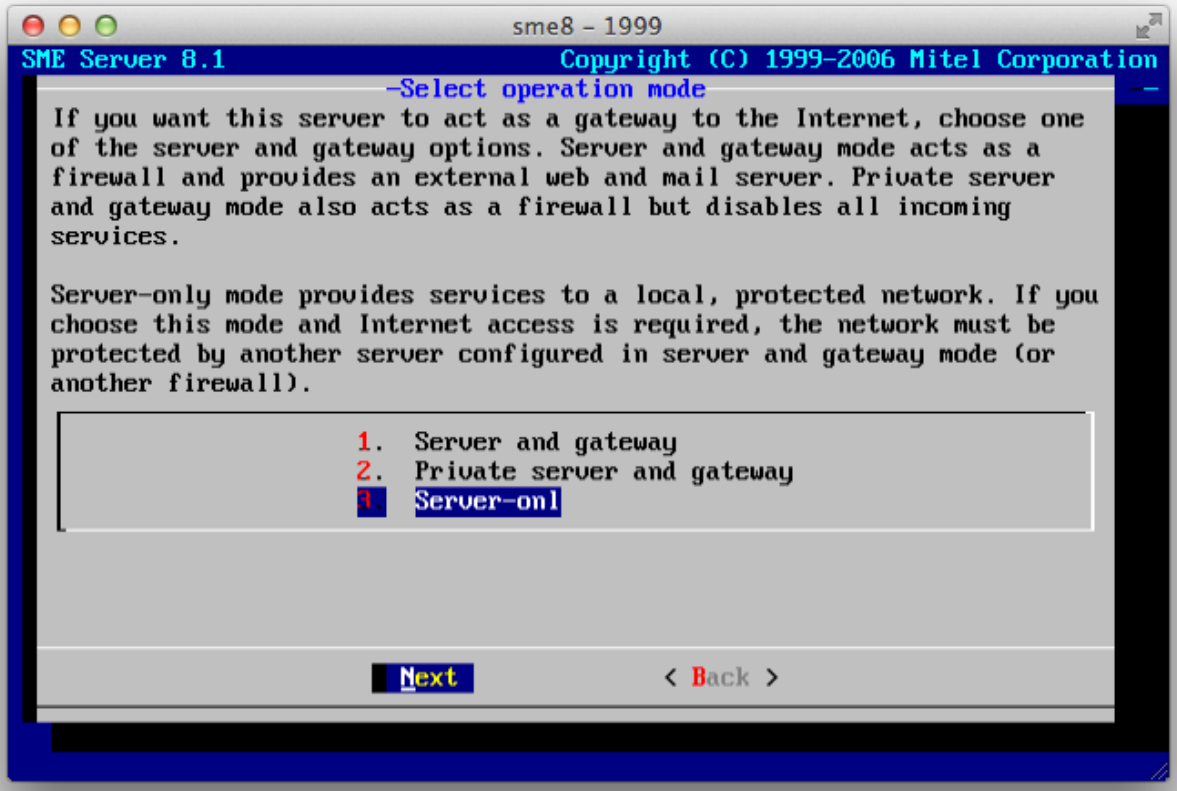
Enter an available local IP address on your z/VM system and “Enter”.



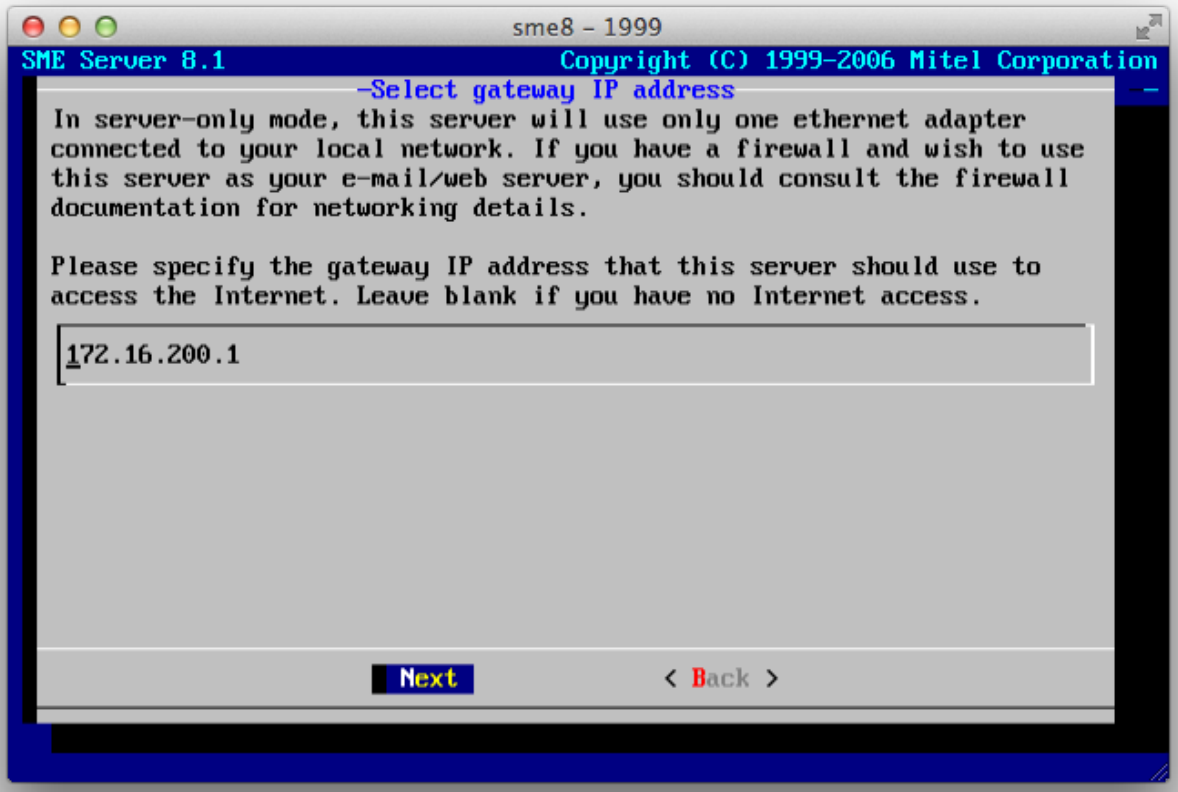
Enter a local subnet mask or accept the default and “Enter”.



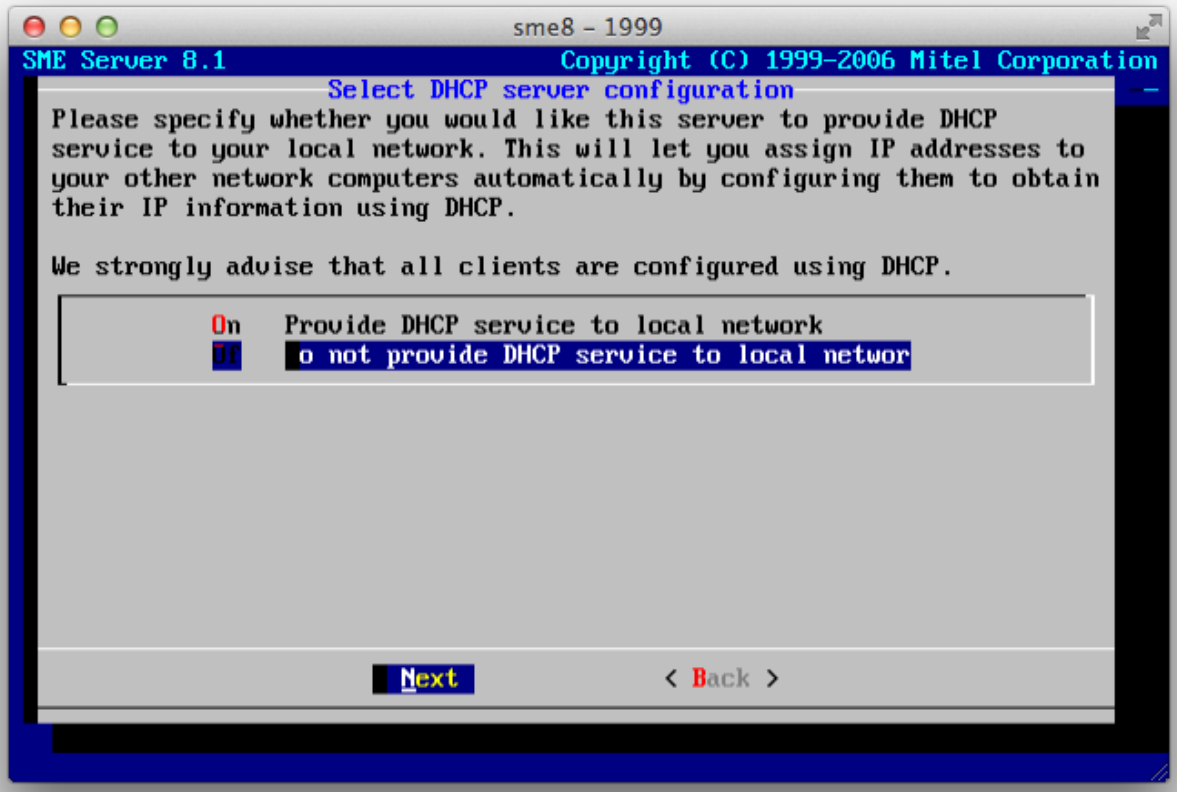
Highlight “3 Server Only”, and “Enter”.



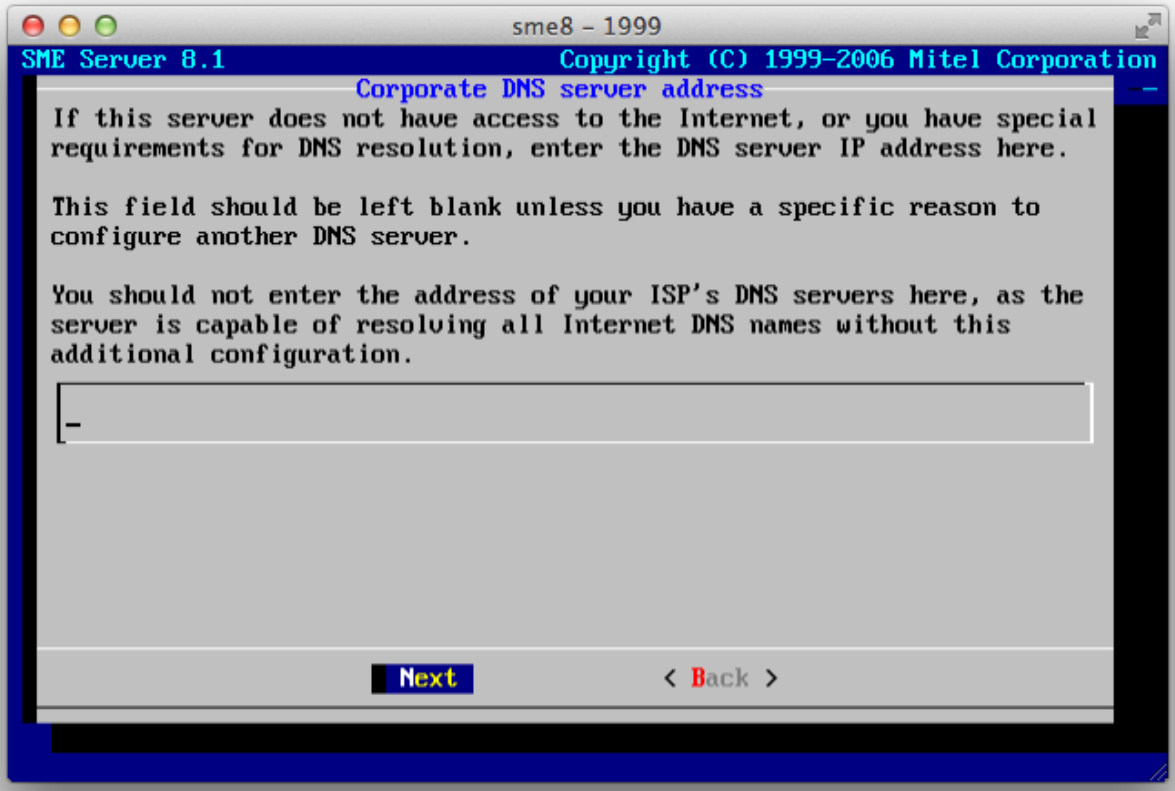
Enter the local gateway IP address and “Enter”.



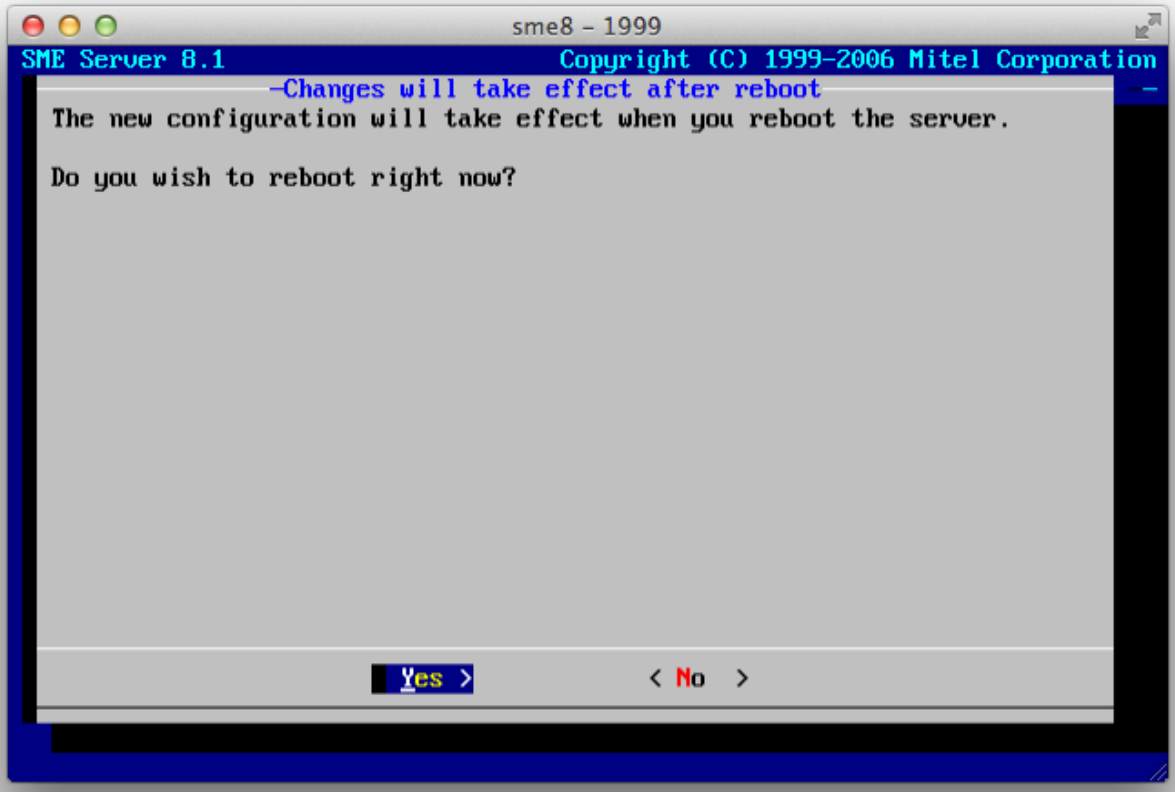
Select “Off” and “Enter”



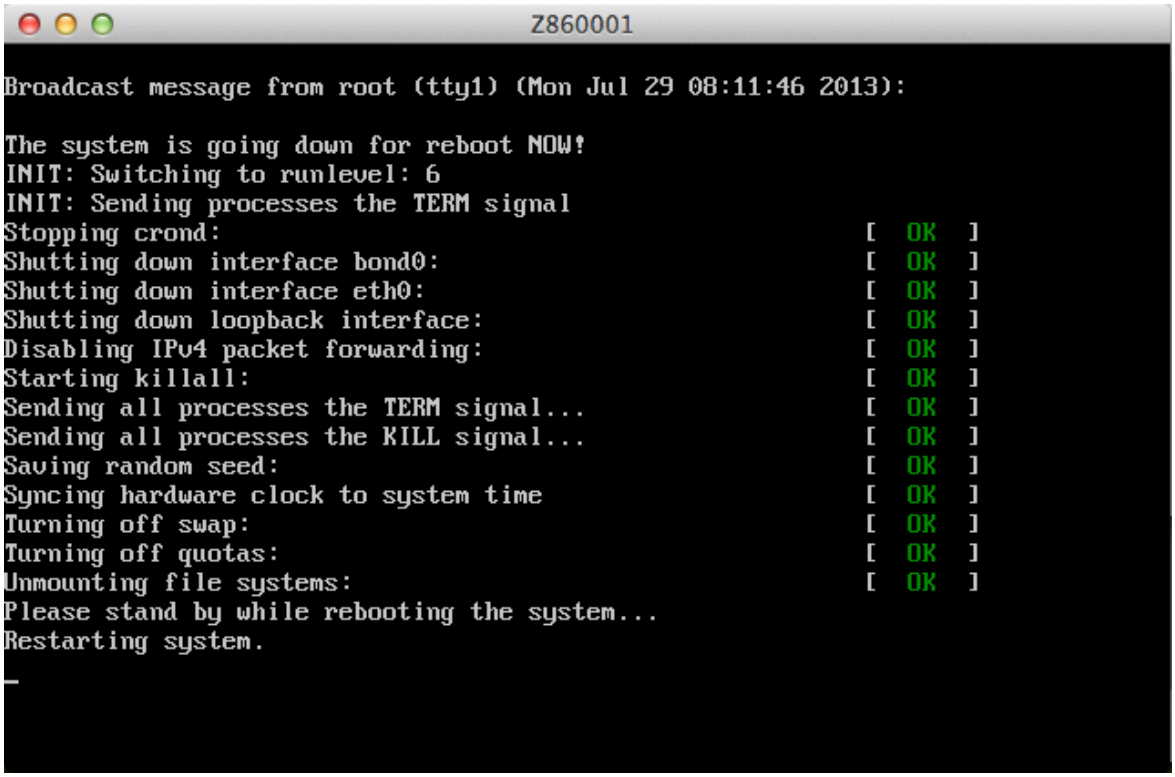
”Enter”.


sme8 - 1999
SME Server 8.1
Copyright (C) 1999-2006 Mitel Corporation
Corporate DNS server address
If this server does not have access to the Internet, or you have special requirements for DNS resolution, enter the DNS server IP address here.
This field should be left blank unless you have a specific reason to configure another DNS server.
You should not enter the address of your ISP's DNS servers here, as the server is capable of resolving all Internet DNS names without this additional configuration.
Next
< Back >

”Enter”.


sme8 – 1999
SME Server 8.1
Copyright (C) 1999-2006 Mitel Corporation
Changes will take effect after reboot
The new configuration will take effect when you reboot the server.
Do you wish to reboot right now?
Yes
No

When all processes are stopped, “Restarting system” will be displayed.

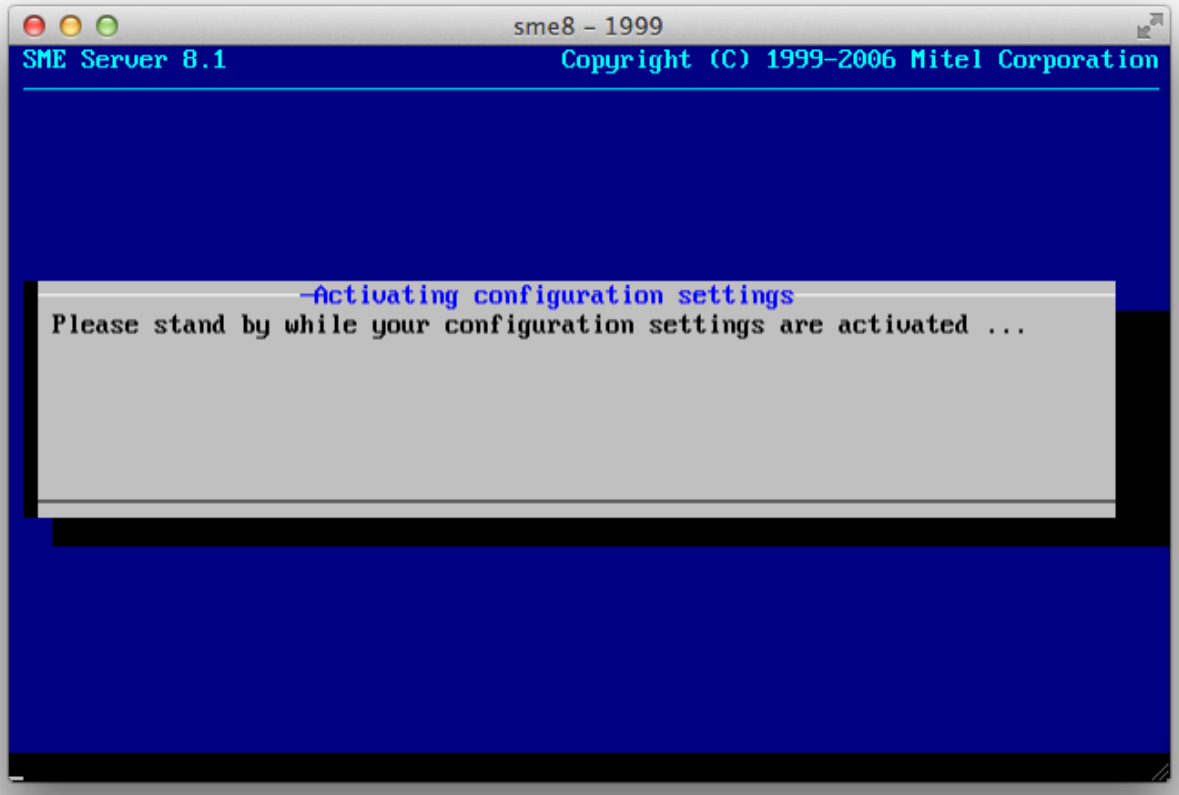


At this point, the SME user id must be stopped and restarted for the reconfiguration to take place.

From the Z86MNT z/VM id,
Enter: **z86 (stop=sme8)**
Enter: **YES** to force the SME off

After the stop completes,
Enter: **z86 (start=sme8)**

SME will boot as before and will display the panel:



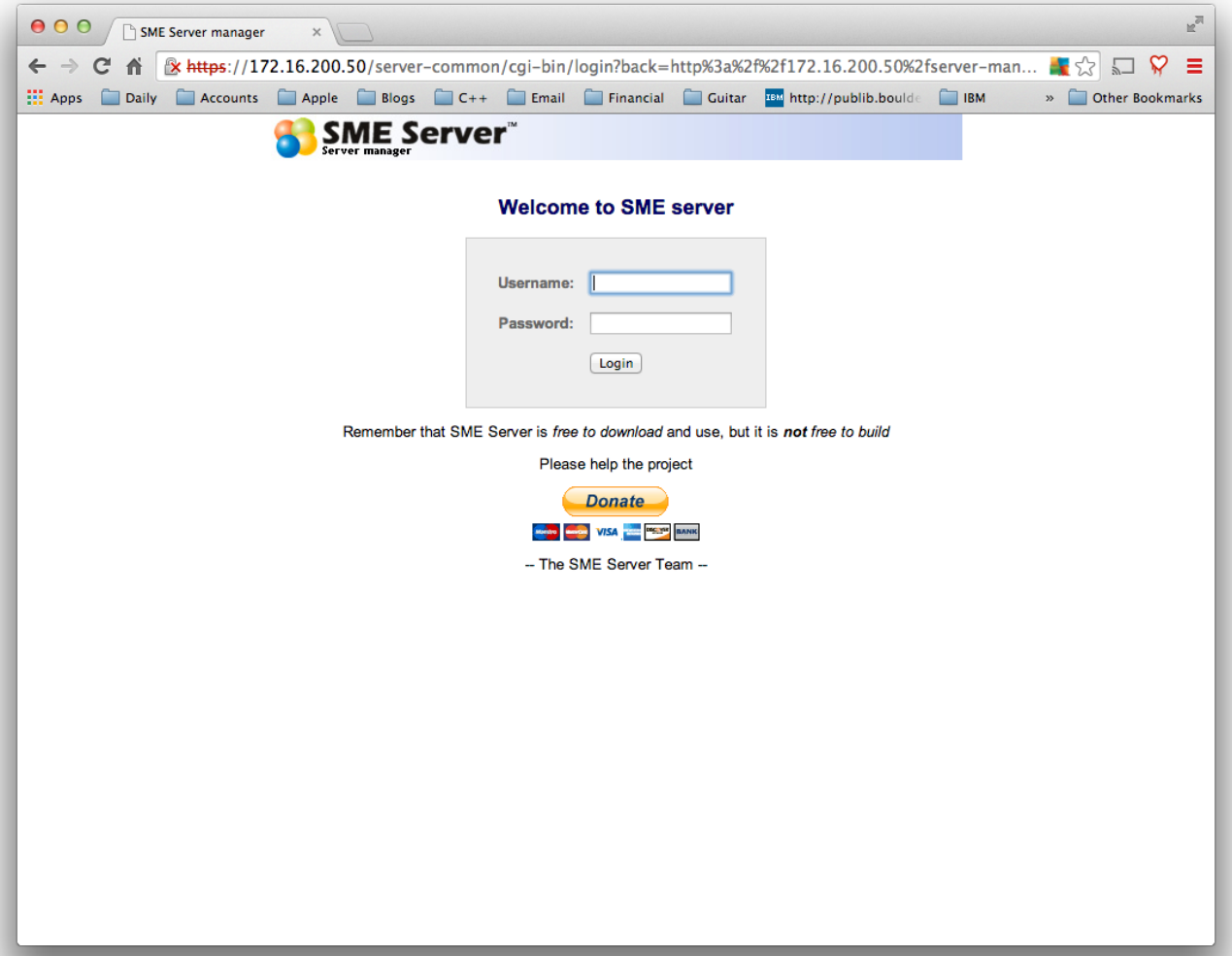
After the reconfiguration, SME will reboot and again display the login request. The SME Server is now operational.

From a Browser enter:

https://<your IP address>/server-manager

where <your IP address> is the IP address you configured previously. It must also be all lower case as it is case sensitive.

The SME Server following login panel will appear.
Enter Username: admin
Password: Noodle01



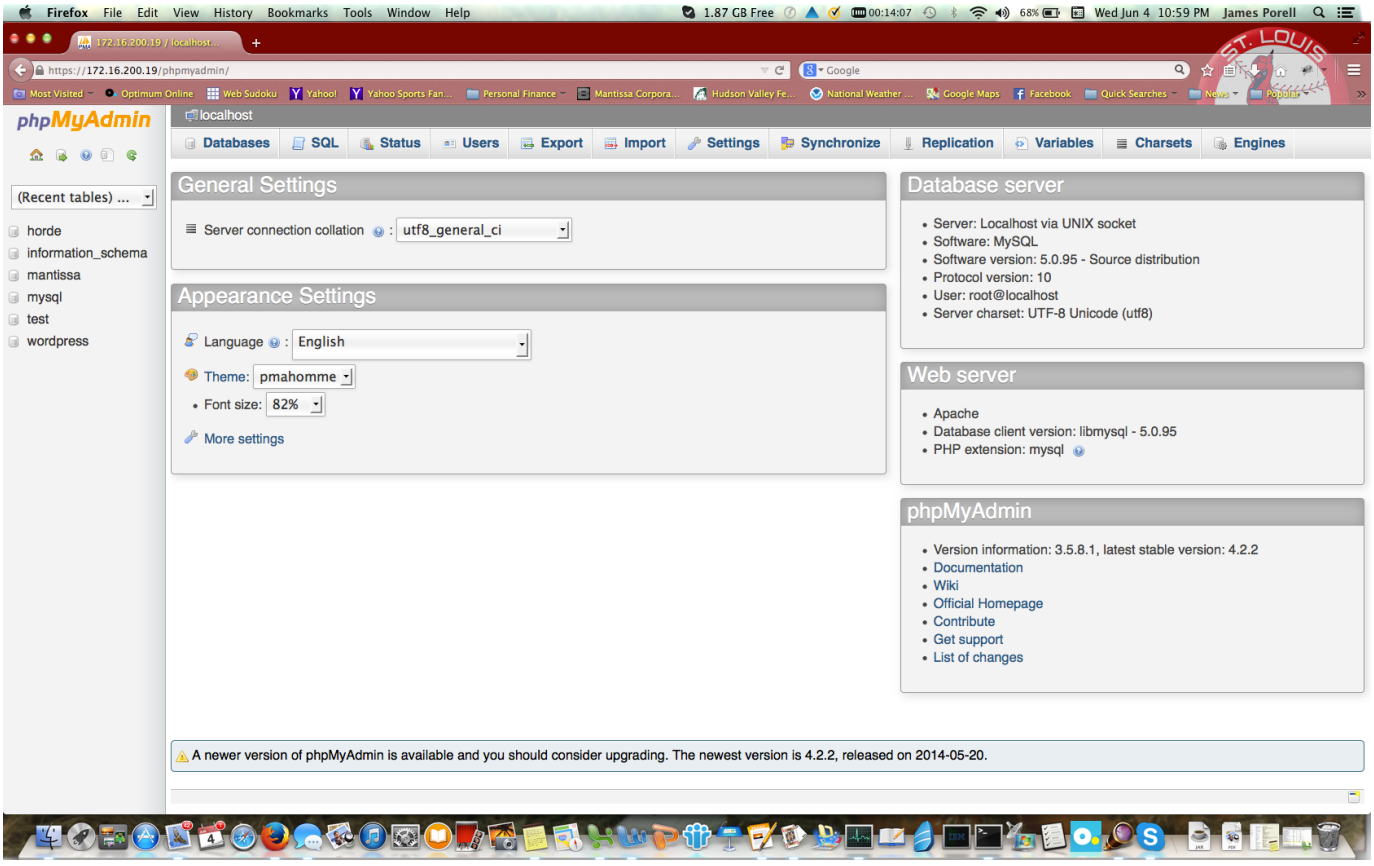
Opening phpmyadmin

** added 06/05/14
Similar to starting the SME server-manager.
From a Browser enter:

https://<your IP address>/phpmyadmin

where <your IP address> is the IP address you configured previously. It is also case sensitive.

The SME Server following login panel will appear.
Enter Username: admin
Password: Noodle01



Let us know how you did!

At Mantissa, we'd greatly appreciate your feedback on the performance and usage, even though it is just testing on your part.

Collecting additional performance information

After test image customization and subsequent IPL activate z86VM stats reporting by entering (on the DEMODSL console):

//son <=== Stats on command

System Response:

```
Echo __ //SON

Command issued=SON

11:21:06 cmu (mmudisp ) - ________ Stats display enabled
```

Capturing Performance Data

We are interested in several performance metrics you may obtain. The first is test script duration. Upon script termination a message ("Script took XX seconds to run") recording duration is displayed. Please record that number. This duration may be longer than with DSL owing to increased trace overhead.

The next metric of interest is resting CPU consumption. Every ten seconds CPU consumption is reported. After you see the GUI appear in the VNC session, let the system remain idle for several minutes and collect a text clip of the statistics output from the VM guest console. It should look something like this.

```
11:35:00 cmu (cmumaint) - _______ x86 mips high(0) interval(0) avg over 88 cpu sec(0) x86im(0) vcpu(1045004166)
11:36:10 cmu (cmumaint) - _______ x86 mips high(0) interval(0) avg over 89 cpu sec(0) x86im(0) vcpu(1046444810)
11:37:30 cmu (cmumaint) - _______ x86 mips high(0) interval(0) avg over 91 cpu sec(0) x86im(0) vcpu(1048526709)
11:37:40 cmu (cmumaint) - _______ x86 mips high(0) interval(0) avg over 91 cpu sec(0) x86im(0) vcpu(1048636213)
```

Finally, we would like to see output produced by a z86VM probe command. By entering (on the z86 user console):

//probe <=== Probe command

on the VM guest console you should get a series of messages similar to these:

```
z86VM - Probe Name(CACHEINSTRUCTIONLINESIZE ) Count(256) Desc(Cache Instruction Cache Line Size
z86VM - Probe Name(CACHEINSTRUCTIONTOTALL1 ) Count(65536) Desc(Cache Instruction Total Level 1 Size
z86VM - Probe Name(CACHEINSTRUCTIONTOTALL2 ) Count(1572864) Desc(Cache Instruction Total Level 2 Size
z86VM - Probe Name(CACHEDATALINESIZE ) Count(256) Desc(Cache Data Cache Line Size
z86VM - Probe Name(CACHEDATATOTALLEVEL1 ) Count(131072) Desc(Cache Data Total Level 1 Size
z86VM - Probe Name(CACHEDATATOTALLEVEL2 ) Count(1572864) Desc(Cache Data Total Level 2 Size
z86VM - Probe Name(MMURTEREPAIRS ) Count(2527) Desc(MMU RTE repairs
z86VM - Probe Name(EXTINTCOMPARATORSECONDS ) Count(1075) Desc(External Interrupt Comparator Calculated Sec
z86VM - Probe Name(EXTINTTIMER ) Count(90065) Desc(External Interrupt for Timer
z86VM - Probe Name(EXTTOTAL ) Count(202467) Desc(External Interrupt Total for A-L-L
z86VM - Probe Name(PIT0TICKS ) Count(106919) Desc(TimeX Total Pit0 ticks
z86VM - Probe Name(PIT0POSTS ) Count(107547) Desc(TimeX Total Pit0 posts
z86VM - Probe Name(PIT2TICKS ) Count(21383) Desc(TimeX Pit2 ticks
z86VM - Probe Name(PIT2POSTS ) Count(106927) Desc(TimeX Pit2 posts
```

Clip and save.
Send Mantissa your results

We would appreciate an email containing:

test script run time,
resting CPU consumption,
//probe command output
model number of your machine.

Problem reporting:

If z86VM ABENDs for any reason, clear the z/VM guest console of the abending z86VM user and provide the following information:
"x'5B5000' contains 16 64 bit registers, followed by 16 32 bit registers, followed by the ABEND PSW at 5B5100. The failing instruction address will be at 5B510C
"On the z86 Console, to display the registers and PSW, Enter:

CP D T5B5000.200

This will produce a dump of the system diagnostic work area (SDWA). It will look something like this:

```
V005B5000 00000000 00018261 00000000 00000000 06 *......b/........*
V005B5010 00000000 00000000 00000000 00087580 *.................*
V005B5020 00000000 C0825DA0 00000000 00000F9C *.....b).........*
V005B5030 00000000 FFFFA020 00000000 00080F9B *.................*
V005B5040 00000000 0572389E 00000000 C0425848 *.................*
```

```
V005B5050 00000000 0015EA00 00000000 FFFFA000 *................*
V005B5060 00000000 1CF61900 00000000 1C962128 *.....6.......o..*
V005B5070 00000000 00000000 00000000 00000F9D *................*
V005B5080 00018261 00000000 00000000 00087580 *..b/............*
V005B5090 C0825DA0 00000F9C FFFFA020 00080F9B *.b)............*
V005B50A0 0572389E C0425848 0015EA00 FFFFA000 *................*
V005B50B0 1CF61900 1C962128 00000000 00000000 *.6...o..........*
V005B50C0 00000000 00000000 00000000 00000000 *................*
V005B50D0 00000001 00000001 00000001 00000001 *................*
V005B50E0 00000000 00000001 00000000 00000001 *................*
V005B50F0 00000000 00000000 00000000 00000000 *................*
V005B5100 47045001 80000000 00000000 000DD252 *..&...........K.*
V005B5110 00000101 00470440 01800000 00000000 *....... ........*
V005B5120 0005775F F4000600 00000000 00000000 *...¬4...........*
V005B5130 00000000 00000000 D7000000 00000000 *........P.......*
V005B5140 00000000 00000000 00000000 00000000 *................*
V005B5150 to 005B51FF suppressed line(s) same as above ....
```

In this Example, the failing instruction address is DD252. So to capture that data, we'll need data before and after this failure. In this case, enter:

`CP D TDD200.100`

```
V000DD200 A5080001 D201D1C0 8102E310 81220071 06 *v...K.J.a.T.a...*
V000DD210 E310D1B8 0024D700 D1D4D1D4 E310D1B8    *T.J...P.JMJMT.J.*
V000DD220 00710A00 0000E3F0 00080071 A7F40739    *......T0....x4..*
V000DD230 920081A8 E31080EC 0016E3B0 10100016    *k.ayT.....T.....*
V000DD240 95D73138 A774040B D2073108 8020B2B2    *nP..x...K.......*
V000DD250 3100E390 30480004 9A47815C B2790200    *..T.......a*....*
V000DD260 58103134 5010B208 E310000E 00715820    *....&...T.......*
V000DD270 31304010 B231E310 30480004 5010B22D    *.. ...T.....&...*
V000DD280 E3F0D1D5 0024A7F9 0000BFF3 B233E3F0    *T0JN..x9...3..T0*
V000DD290 80F00018 BEF3B233 E3F0D1D5 0004E300    *.0...3..T0JN..T.*
V000DD2A0 B010001E B9160000 5000B014 94F7B015    *........&...m7..*
V000DD2B0 9108B013 A7E40006 9608B015 94F7B014    *j...xU..o...m7..*
V000DD2C0 4830B231 592080F4 A77401E0 B9040043    *.......4x.......*
V000DD2D0 B9040014 9102B011 A7E40054 9101B213    *....j...xU..j...*
V000DD2E0 A774004D 9601B13C 89400003 E370B1F2    *x..(o...i ..T..2*
V000DD2F0 001EB916 0077E374 70000071 E3407006    *......T.....T ..*
```

Email a copy of output from this storage dump. If possible, leave the guest active and await instructions .

How to Install a New x86 Operating System image

This will be updated in a future release

IMPORTANT USAGE NOTES

VNC Server

z86VM ships with a VNC Server operating. However, this is different than other industry standard VNC servers. It's a boot strap version that was uniquely developed by Mantissa for z/VM. For this VNCServe function, a single IP address applies to the z/VM system image. Each z86 guest user has it's own VNC port address. For this reason, the vncport variable is defined for each z86 user in the PROFILE Z86. See below for multiple guests sharing a system image for using the industry standard VNC Server capabilities.
When signing into the terminal session or desktop client, you'll go to your favorite VNC Client (e.g. JollyFastVNC or tightvnc) and use the IP address of the z/VM system and the VNCport specified for that specific user.
Another thing about this VNC Server is the mouse movement isn't completely synchronized with the host system. Two "mice" pointers may appear on the screen. The Control or CTRL key can be pushed and the mice can be moved together to synchronize them. This may be required often in a session. Using the industry standard VNC server eliminates this problem.

Running with multiple zIO Servers

At Mantissa, for test and development, we have multiple IO servers running so that each developer/tester can control their own environment. This should also be true for customers that have (and should) separated their Production and Development/Test systems. The zIO server acts as the hub for networking communications. As a result, each of the zIO servers must have a separate MAC address. This can be established through z/VM commands, but then once defined, that MAC address may need to be part of individual guest accesses as well. As a result, the MACID variable in the PROFILE Z86 is important to establish and maintain those differences for each zIO server.

Shutting down a z86 guest

Important to shutdown the guest properly to avoid file system integrity issues. Actually haven't hit any of those in a while, but better safe than sorry. We'd had guests up, but disconnected, for many days in a row, without a problem.
For Linux, at the command line: **`shutdown -h now`**

Multiple z86 guests sharing a single system image

e.g. vmimage in PROFILE Z86. There are two considerations - the network address and the file system access.

Network access

At Mantissa, we initially configured our virtual systems with static IP addresses. We also used our home grown VNC Server that has mouse tracking issues. Neither of these help with system cloning! Should you go that route, there were several changes necessary for the SME server.

1. If you are running multiple zIO servers, you may need to modify the MAC id within the guest operating system. On the SME Server instances that are shipping with this release, that is done by editing a file.
    a. use the vi editor to modify the MAC address, if necessary.
    b. **`vi /etc/sysconfig/network-scripts/ifcfg-eth0`** and change the mac@ to your system defined interface. This may require a reboot of the system for the networking to take effect.
2. Using the SME-SERVER Console which is addressable when signing into the admin user id from a telnet or other system console, choose Configure server option and run through the prompts. Update the IP @ through these menus and it will change the IP @ in all required places in /etc. This may also require a system reboot to take effect.
3. Start the VNC Server function by signing into the root id on a telnet session. Issue find / -name vncserver*. Once you find the directory, issue a cd command to that directory and type in vncserver. This will start the daemon on your behalf. Then you should be able to go to your VNC Client and sign in using the IP address of the SME Server image and port 5900.
4. To avoid changing the static IP address for each new z86 user, an organization can start a DHCP server. This will automatically give each system OS it's own IP address when the system is started.

File System Sharing

At this time, we, at Mantissa, haven't tested with a SAN file system or Network File system to completely separate z86 users within the file system. This should work, just as any shared file system works but it hasn't been tested by Mantissa yet.

Mantissa development effort is two-fold: Create & Exploit

At Mantissa, we are trying to split our development effort into two major parts.

1. Create and improve the base z86VM code
   1.1. performance improvements in the base code
   1.2. Usability enhancements in the management of the code
   1.3. functional enhancements
2. Exploit the code via x86 operating systems to solve customer problems
   2.1. Install different x86 operating systems
      2.1.1. Provide initial validation that a given OS image can work
      2.1.2. document the installation process so that businesses don't have to be dependent on Mantissa to install their own OS images
   2.2. Leverage those operating systems to solve business problems
      2.2.1. This is similar to work that would be done on native x86 hardware or virtual platforms, but the goal is to demonstrate it is simpler to deploy and cost competitive to alternative hardware implementations.

The exploitation doesn't require that Mantissa try it first. However, we realize that to establish product value, we must do these type of tests and create proof points with quantifiable value statements. We would love to see you try some of these configurations in your own business and provide us with some results. We'll be happy to work with your business to make any improvements that you might identify.

Some examples of customer value activities to try

1. Run experiments and send us the results. These include:
   1. Sudoku - performance script
   2. Screen shots of different interfaces. Need to include IP @ or URL in pic so we know it's internal network vs. real Intel
      1. DSL home screen with message in TERM: Hello z86VM (DSL has IP @ on screen)
      2. PHPMyadmin web screen with URL
      3. Wordpress default screen with URL
      4. Server-Manager web screen with URL
2. How about posting youtube video of your experience
3. Known problems list - any customer that wants to try to work in those areas is welcome to do so.

4. File server - Priority 2
5. Print server - Priority 3
6. Firefox or other browser with external access
7. Youtube or other streaming media capability

8. fully functional Linux Desktop Priority 4 - DSL is okay, but I don't believe it's going to be a big commercial seller….mostly because a commercial customer - our target audience - wants vendor supported software. We are NOT that vendor. The choices could be: Ubuntu, RH or SLES . while Ubuntu is cool, if there are no vendors that will do service contracts, we should try RH or SLES. Free is good for "public cloud". Priced is better for Private clouds.
    1. **Do we know enough about the Virtual Box installs that Antonio does for distributions to be able to explain what's needed for other people to do installs.**
    2. **It would be cool to have a customer "deploy" their favorite distro and share it with us.**
    3. I think we need to do the Benchmark primitives on file and network before we offer this up.
9. VDI of desktop systems - Priority 9. Probably based on Virtual Box technology. Reality is, my demo proves that zVM provides the virtualization necessary to be successful, but the VNC client has usability and graphics support issues. We need to look into RDP, SPICE and NX client as client connector replacements for VNC.
10. Libre Office and other tools to show potential for desktop replacement
11. JAVA applications - need to record performance time, somehow.
12. Use of a Desktop system to host z/OS operator's consoles (a product, such as Tivoli Enterprise Console vs. TN3270). Demonstrates the ability to eliminate end user management of console software.
13. Full LAN virtual desktop experience: LDAP or Active Directory, file server, print server, Desktop connectors (RDP, NX, SPICE) - first one here gets free license to z86VM for one year?

Things Mantissa has done and will document via the web

1. word press personalization
2. Cloning word press - use of duplicator and the big clues and importance of knowing to chmod 777 the source directory and knowing the contents of wp-config.php to answer the questions when installing the clone.

Appendix A - Setting up the z/VM VSWITCH

The following procedure is an example that can be used as a guide to create the required VSW86 VSWITCH.

1. Query the z/VM default VSWITCH Controllers. Pre-configured controllers DTCVSW1 and DTCVSW2 are provided starting with z/VM Release 5, Version 2.0:

```
Enter: Q CONTROLLER
HCPSWQ2835E CONTROLLER * does not exist
Ready(02835);
```

2. If the Controllers are not logged on, XAUTOLOG DTCVSW1 and DTCVSW2:

```
Enter: XAUTOLOG DTCVSW1
Command accepted
AUTO LOGON  ***       DTCVSW1  USERS = 22
Ready;
HCPCLS6056I XAUTOLOG information for DTCVSW1: The IPL command is verified
by the IPL command processor.

Enter: XAUTOLOG DTCVSW2
Command accepted
Ready;
AUTO LOGON  ***       DTCVSW2  USERS = 23
HCPCLS6056I XAUTOLOG information for DTCVSW2: The IPL command is verified
by the IPL command processor.
```

3. Query the Controllers again to assure they are available:

```
Enter: Q CONTROLLER
Controller DTCVSW1   Available: YES   VDEV Range: 0600-F000 Level 620
  Capability: IP ETHERNET VLAN_ARP GVRP    LINKAGG    ISOLATION
              NO_ENSEMBLE NO_INMN
Controller DTCVSW2   Available: YES   VDEV Range: 0600-F000 Level 620
  Capability: IP ETHERNET VLAN_ARP GVRP    LINKAGG    ISOLATION
              NO_ENSEMBLE NO_INMN
Ready;
Query z/VM OSA Adapters:
Enter: Q OSA ALL
OSA  0020 ATTACHED TO ZOS113   0300 DEVTYPE OSA         CHPID 02 OSD
OSA  0021 ATTACHED TO ZOS113   0301 DEVTYPE OSA         CHPID 02 OSD
OSA  0022 ATTACHED TO ZOS113   0302 DEVTYPE OSA         CHPID 02 OSD
OSA  0060 ATTACHED TO TCPIP    0060 DEVTYPE OSA         CHPID 02 OSD
OSA  0061 ATTACHED TO TCPIP    0061 DEVTYPE OSA         CHPID 02 OSD
OSA  0062 ATTACHED TO TCPIP    0062 DEVTYPE OSA         CHPID 02 OSD
OSA  0050 FREE    , OSA  0051 FREE    , OSA  0052 FREE    , OSA  0053 FREE
OSA  0054 FREE    , OSA  0055 FREE
OSA  0063 OFFLINE , OSA  0064 OFFLINE , OSA  0065 OFFLINE
An OSA Agent was not found.
Ready;
```

4. Any OSA Adapter can be used. In this example, OSA Adapter RDEV 0063 is available, but offline. Vary the OSA Adapter triplet addresses 0063-0065 online:

```
Enter: VARY ONLINE 063-065
0063 varied online
0064 varied online
0065 varied online
3 device(s) specified; 3 device(s) successfully varied online
Ready;
```

5. Query z/VM OSA Adapters again to assure that Adapter is free to be used:

```
Q OSA ALL
OSA  0020 ATTACHED TO ZOS113   0300 DEVTYPE OSA CHPID 02 OSD
OSA  0021 ATTACHED TO ZOS113   0301 DEVTYPE OSA CHPID 02 OSD
OSA  0022 ATTACHED TO ZOS113   0302 DEVTYPE OSA CHPID 02 OSD
OSA  0060 ATTACHED TO TCPIP    0060 DEVTYPE OSA CHPID 02 OSD
OSA  0061 ATTACHED TO TCPIP    0061 DEVTYPE OSA CHPID 02 OSD
OSA  0062 ATTACHED TO TCPIP    0062 DEVTYPE OSA CHPID 02 OSD
OSA  0050 FREE    , OSA  0051 FREE    , OSA  0052 FREE    , OSA  0053 FREE
OSA  0054 FREE    , OSA  0055 FREE    , OSA  0063 FREE    , OSA  0064 FREE
OSA  0065 FREE
An offline OSA was not found.
An OSA Agent was not found.
Ready;
```

6. Define the VSWITCH using the selected OSA Adapter address 0063:

```
Enter: CP DEFINE VSWITCH VSW86 RDEV 0063 CONNECT ETHERNET
VSWITCH SYSTEM VSW86 is created
Ready;
HCPSWU2830I VSWITCH SYSTEM VSW86 status is ready.
HCPSWU2830I DTCVSW1 is VSWITCH controller for device 0063.P00.
```

7. Query the VSWITCH:

```
Enter: QUERY VSWITCH VSW86 DETAILS
VSWITCH SYSTEM VSW86    Type: QDIO    Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-00-00-00-2B    MAC Protection: Unspecified
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
 Uplink Port:
  RDEV: 0063.P00 VDEV: 0600 Controller: DTCVSW1
    Uplink Port Connection:
      RX Packets: 0          Discarded: 0          Errors: 0
      TX Packets: 0          Discarded: 0          Errors: 0
      RX Bytes: 0                    TX Bytes: 0
      Device: 0600  Unit: 000   Role: DATA       Port: 2049
Ready;
```

8. Grant userid Z86IO access to VSWITCH VSW86:

```
Enter: CP SET VSWITCH VSW86 GRANT Z86IO
Command complete
Ready;
```

9. Query access to the VSWITCH to assure that Z86IO is an authorized userid:

```
Enter: QUERY VSWITCH VSW86 ACCESSLIST
VSWITCH SYSTEM VSW86    Type: QDIO    Connected: 0    Maxconn: INFINITE
  PERSISTENT  RESTRICTED    ETHERNET                  Accounting: OFF
  USERBASED
  VLAN Unaware
  MAC address: 02-00-00-00-00-2B    MAC Protection: Unspecified
  State: Ready
  IPTimeout: 5         QueueStorage: 8
  Isolation Status: OFF
    Authorized userids:
      Z86IO
```

```
 Uplink Port:
  RDEV: 0063.P00 VDEV: 0600 Controller: DTCVSW1
Ready;
```

The VSWITCH configuration is now complete and available for z86VM.

Appendix B - Sample INSTALL Log

Enter: **z86 (install log more**

```
2014-05-29 10:36:00 -------------------------
2014-05-29 10:36:00 Z86 PARMS
2014-05-29 10:36:00 Z86      EXEC     A2: 2014-05-21 14:32:27
2014-05-29 10:36:00 PROFILE  Z86      A: 2014-05-29 10:04:21
2014-05-29 10:36:00 z86VM machines configured: 2
2014-05-29 10:36:00 TCP/IP Stack: TCPIP
2014-05-29 10:36:00 -------------------------
2014-05-29 10:36:00 TCPIP Stack: TCPIP
2014-05-29 10:36:00 -------------------------
2014-05-29 10:36:00 1 I/O Servers
2014-05-29 10:36:00   ID: Z86IO
2014-05-29 10:36:00   VNICADR: 0100
2014-05-29 10:36:00   VSWITCH: VSW86
2014-05-29 10:36:00   MACID:
2014-05-29 10:36:00   Module: ZIO5 MODULE F: 2014-04-03 15:03:08
2014-05-29 10:36:00   Origin: 10000
2014-05-29 10:36:00   Boot vadr: 0195
2014-05-29 10:36:00   x86HD: 0200 ZV0048 10016 Cyls
2014-05-29 10:36:00   x86HD: 0201 ZV0051 10016 Cyls
2014-05-29 10:36:00   ------------------------
2014-05-29 10:36:00 -------------------------
2014-05-29 10:36:00 2 z86VM Servers
2014-05-29 10:36:00   ID: DEMODSL
2014-05-29 10:36:00   I/O Server: Z86IO
2014-05-29 10:36:00   Module: V0R3C3M MODULE A: 2014-05-21 12:55:28
2014-05-29 10:36:00   Origin: 10000
2014-05-29 10:36:00   Boot vadr: 0194
2014-05-29 10:36:00   BootDev: HD0
2014-05-29 10:36:00   BootType: HD
2014-05-29 10:36:00   VNC Port: 8700
2014-05-29 10:36:00   Memory: 900M
2014-05-29 10:36:00   HD Memory: 6G
2014-05-29 10:36:00   Sector: 512
2014-05-29 10:36:00   VM Image: DSL2012   IMG    A: 2014-05-06 13:31:53
2014-05-29 10:36:00   PC Image: DSL.IMG
2014-05-29 10:36:00   ------------------------
2014-05-29 10:36:00   ID: SME8
2014-05-29 10:36:00   I/O Server: Z86IO
2014-05-29 10:36:00   Module: V0R3C3M MODULE A: 2014-05-21 12:55:28
2014-05-29 10:36:00   Origin: 10000
2014-05-29 10:36:00   Boot vadr: 0194
2014-05-29 10:36:00   BootDev: HD0
2014-05-29 10:36:00   BootType: HD
2014-05-29 10:36:00   VNC Port: 8701
2014-05-29 10:36:00   Memory: 900M
2014-05-29 10:36:00   HD Memory: 7G
2014-05-29 10:36:00   Sector: 512
2014-05-29 10:36:00   VM Image: SME8110 DEMOWP   A: 2014-04-28 20:20:17
```

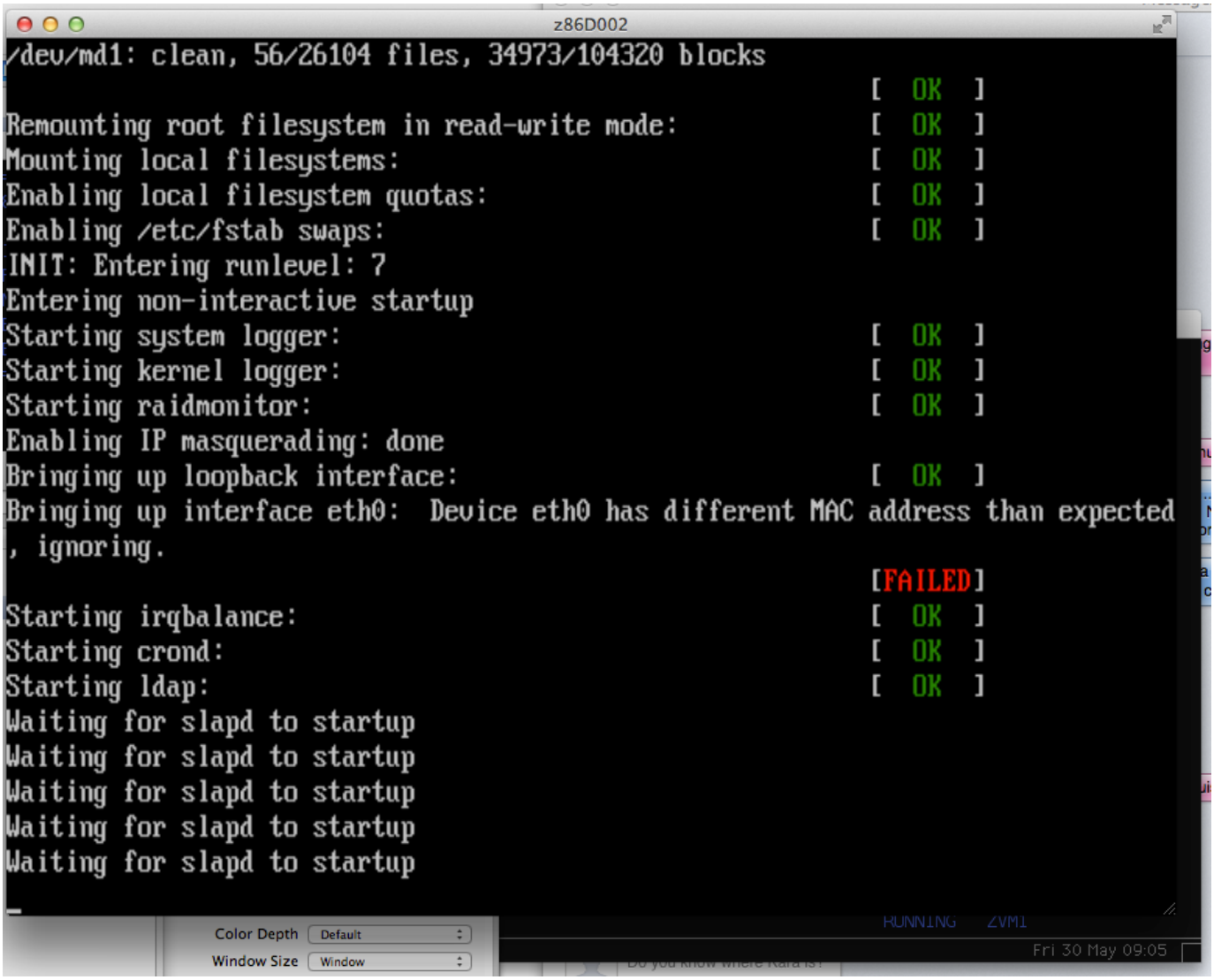
```
2014-05-29 10:36:00   PC Image: SME.IMG
2014-05-29 10:36:00   ------------------------
2014-05-29 10:36:00 --------------------------
2014-05-29 10:36:00 Are you sure you want to force I/O Server Z86IO
2014-05-29 10:36:00 Answer YES to FORCE Z86IO or NO to skip
2014-05-29 10:36:07 YES
2014-05-29 10:36:08 FORCE Z86IO rc: 0
2014-05-29 10:36:09 Initializing DEMODSL 0194
2014-05-29 10:36:09 Format DEMODSL 0194 10 Cylinders rc: 0
2014-05-29 10:36:09 Format Recomp DEMODSL 0194 9 Cylinders rc: 0
2014-05-29 10:36:09 IPLPARMS: IPLPARMS IO=Z86IO,TCPIP=TCPIP,NICCUU=0100,VNC=8700
2014-05-29 10:36:09 CMD_
2014-05-29 10:36:09 Installed V0R3C3M MODULE A 2014-05-21 12:55:28 on DEMODSL 0194
rc: 0
2014-05-29 10:36:09 Initializing SME8    0194
2014-05-29 10:36:09 Format SME8    0194 10 Cylinders rc: 0
2014-05-29 10:36:09 Format Recomp SME8    0194 9 Cylinders rc: 0
2014-05-29 10:36:09 IPLPARMS: IPLPARMS IO=Z86IO,TCPIP=TCPIP,NICCUU=0100,VNC=8701
2014-05-29 10:36:09 CMD_
2014-05-29 10:36:09 Installed V0R3C3M MODULE A 2014-05-21 12:55:28 on SME8    0194
rc: 0
2014-05-29 10:36:09 Initializing Z86IO 0195
2014-05-29 10:36:09 Format 0195 10 Cylinders rc: 0
2014-05-29 10:36:09 Format Recomp 0195 9 Cylinders rc: 0
2014-05-29 10:36:09 SALIPL 0195 (MODULE  ZIOMAIN ORIGIN 10000
2014-05-29 10:36:09 Installed Stand-Alone Program Loader on 0195 rc: 0
2014-05-29 10:36:09 Installed ZIO5 MODULE F 2014-04-03 15:03:08 on Z86IO 0195 rc: 0
2014-05-29 10:36:09 --------------------------
2014-05-29 10:36:09 Formatting Z86IO x86 volumes
2014-05-29 10:36:09  Format x86 0200 complete.
2014-05-29 10:38:01 Format 0200 10016 Cylinders rc: 0
2014-05-29 10:38:01  Format x86 0201 complete.
2014-05-29 10:39:54 Format 0201 10016 Cylinders rc: 0
2014-05-29 10:39:59 Initializing Z86IO x86 Extents
2014-05-29 10:39:59 Sending: Z86IO FINIT 0200 ZIO001
2014-05-29 10:41:29  x86 0200 Complete
2014-05-29 10:41:29 Sending: Z86IO FINIT 0201 ZIO002
2014-05-29 10:42:59  x86 0201 Complete
2014-05-29 10:43:04 SMSG Z86IO ADDUSER DEMODSL
2014-05-29 10:43:07 SMSG Z86IO ADDFILE DEMODSL HD DSL.IMG 512 6G
2014-05-29 10:43:10 SMSG Z86IO ADDBOOT DEMODSL MEM=900M HD0=DSL.IMG BOOT=HD0 TYPE=16
CPU=FFFFA000
2014-05-29 10:43:13 DEMODSL created on Z86IO.
2014-05-29 10:43:13 Installing DSL2012   IMG    A (2014-05-06 13:31:53) as DSL.IMG
for DEMODSL
2014-05-29 10:43:24 zvcopy Z86IO DEMODSL PC DSL.IMG DSL2012   IMG    A   rc: 0
2014-05-29 10:43:25 --------------------------
2014-05-29 10:43:25 SMSG Z86IO ADDUSER SME8
2014-05-29 10:43:28 SMSG Z86IO ADDFILE SME8    HD SME.IMG 512 7G
2014-05-29 10:43:31 SMSG Z86IO ADDBOOT SME8    MEM=900M HD0=SME.IMG BOOT=HD0 TYPE=16
CPU=FFFFA000
2014-05-29 10:43:34 SME8    created on Z86IO.
```

```
2014-05-29 10:43:34 Installing SME8110 DEMOWP   A (2014-04-28 20:20:17) as SME.IMG
for SME8
2014-05-29 10:45:22 zvcopy Z86IO SME8    PC SME.IMG SME8110 DEMOWP   A   rc: 0
2014-05-29 10:45:23 --------------------------
```

Appendix C - Trouble Shooting

MAC Address or HWADDR Failure

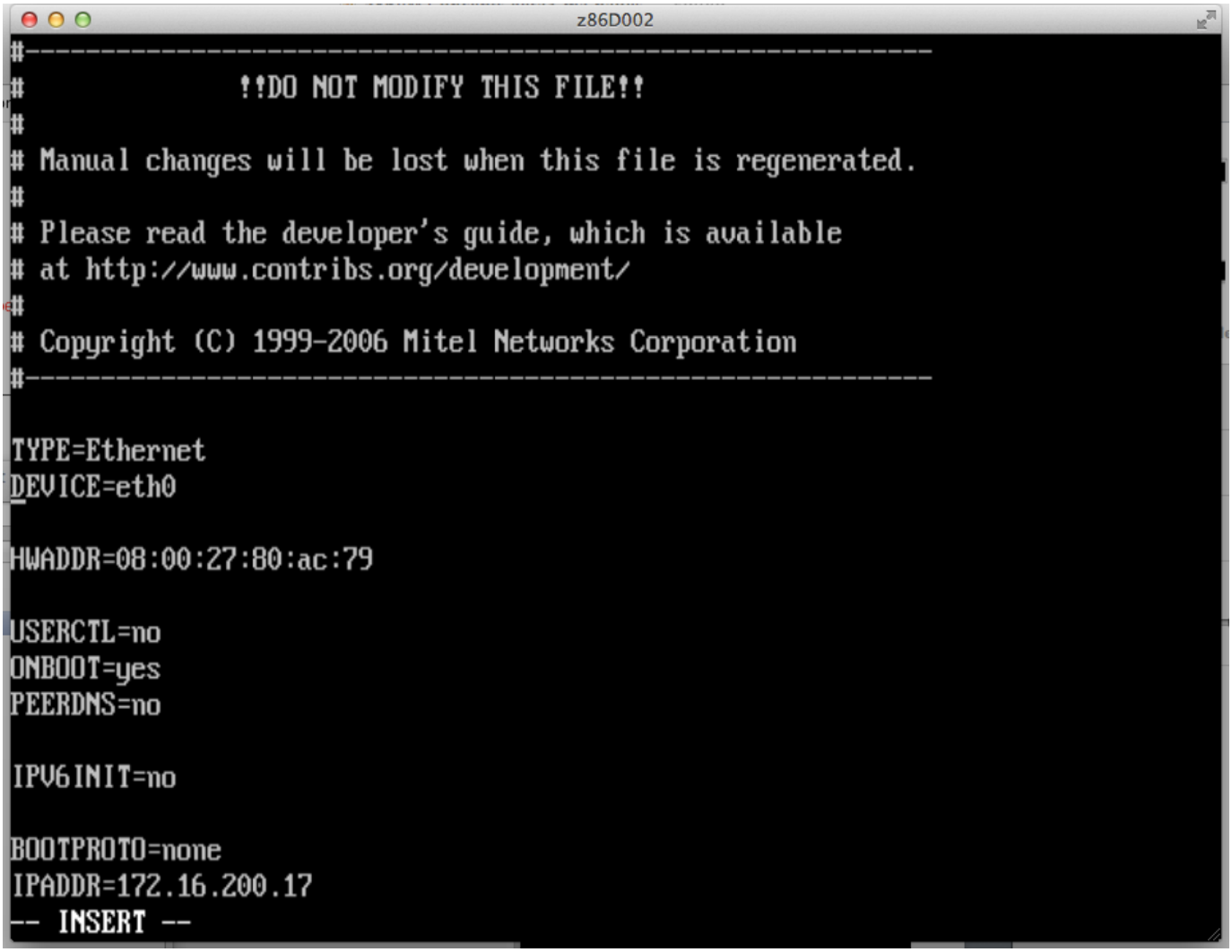
When multiple zIO Servers are running within the same z/VM image, different MAC addresses are necessary to manage communications. When the MAC address of an x86 image doesn't match the MAC address of the zIO server, you may see a message that looks like the following:



Allow the system to boot up to completion and requesting a logon prompt. For this SME distribution, logon with the id: root and password: Noodle01

At the command prompt, use the vi editor on file:
Enter: **`vi /etc/sysconfig/network-scripts/ifcfg-eth0`**

You may see a file that looks like this:

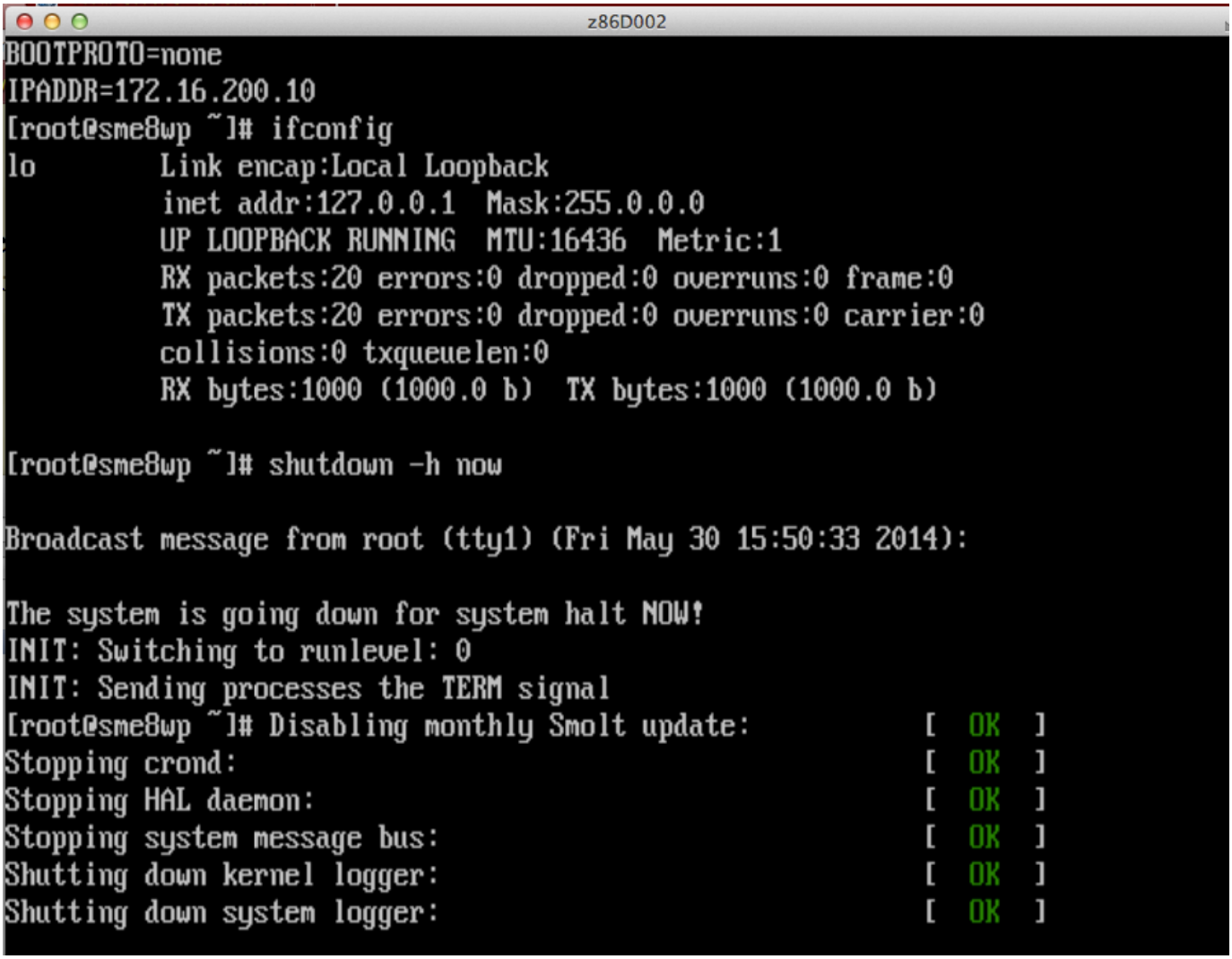


using the vi edit commands, change the HWADDR. The address you specify must be the same MACID that you specify in ***3. Define z86 user ids***
You can also change the IP address here, but you'll also have to use the SME ADMIN console actions, described in the section: ***Running the SME8 Server*** to make the changes complete in each file necessary.

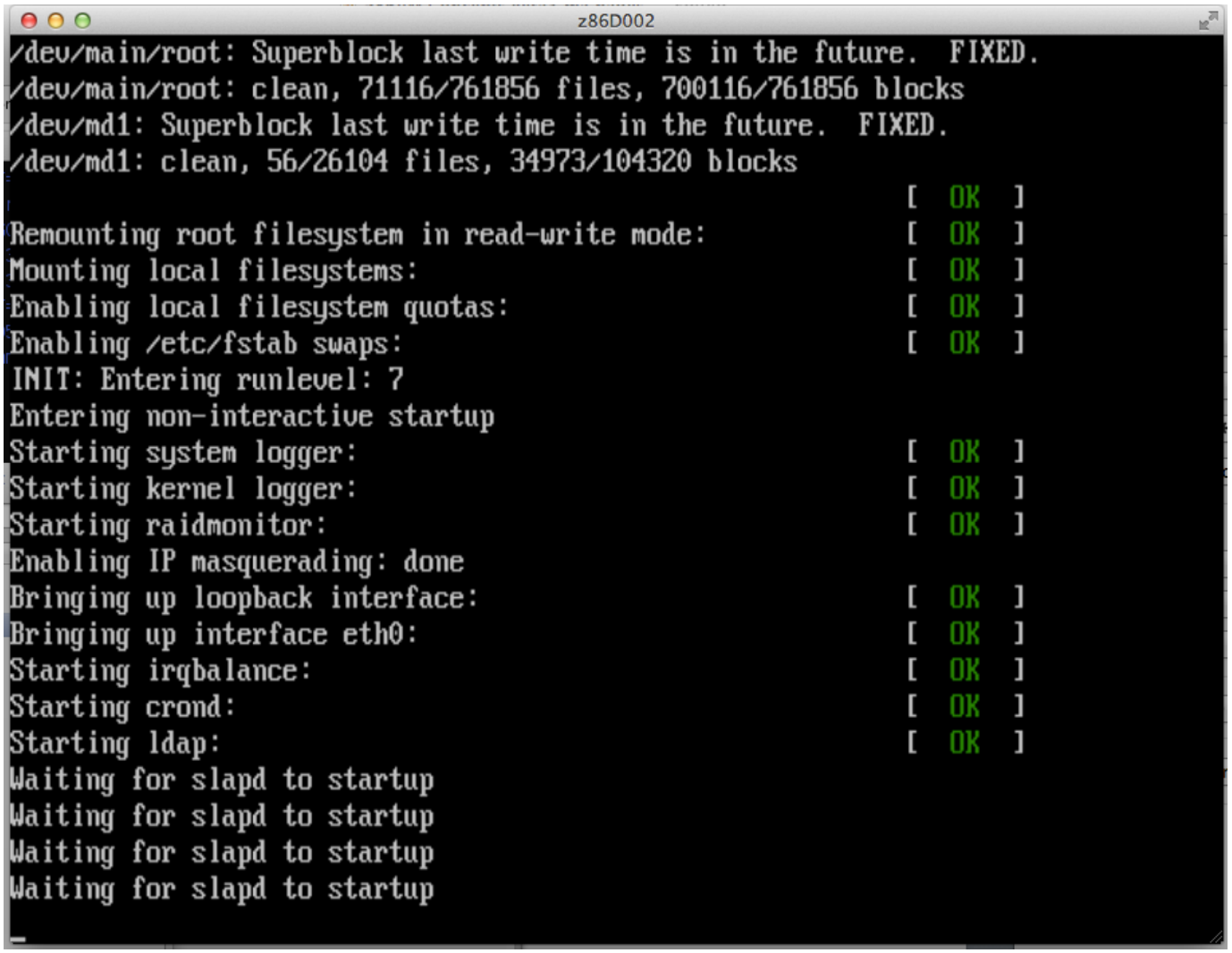
```
#------------------------------------------------------------
#               !!DO NOT MODIFY THIS FILE!!
#
# Manual changes will be lost when this file is regenerated.
#
# Please read the developer's guide, which is available
# at http://www.contribs.org/development/
#
# Copyright (C) 1999-2006 Mitel Networks Corporation
#------------------------------------------------------------

TYPE=Ethernet
DEVICE=eth0

HWADDR=02:00:00:ff:ff:03

USERCTL=no
ONBOOT=yes
PEERDNS=no

IPV6INIT=no

BOOTPROTO=none
IPADDR=172.16.200.10
```

Save the file upon completion of editing

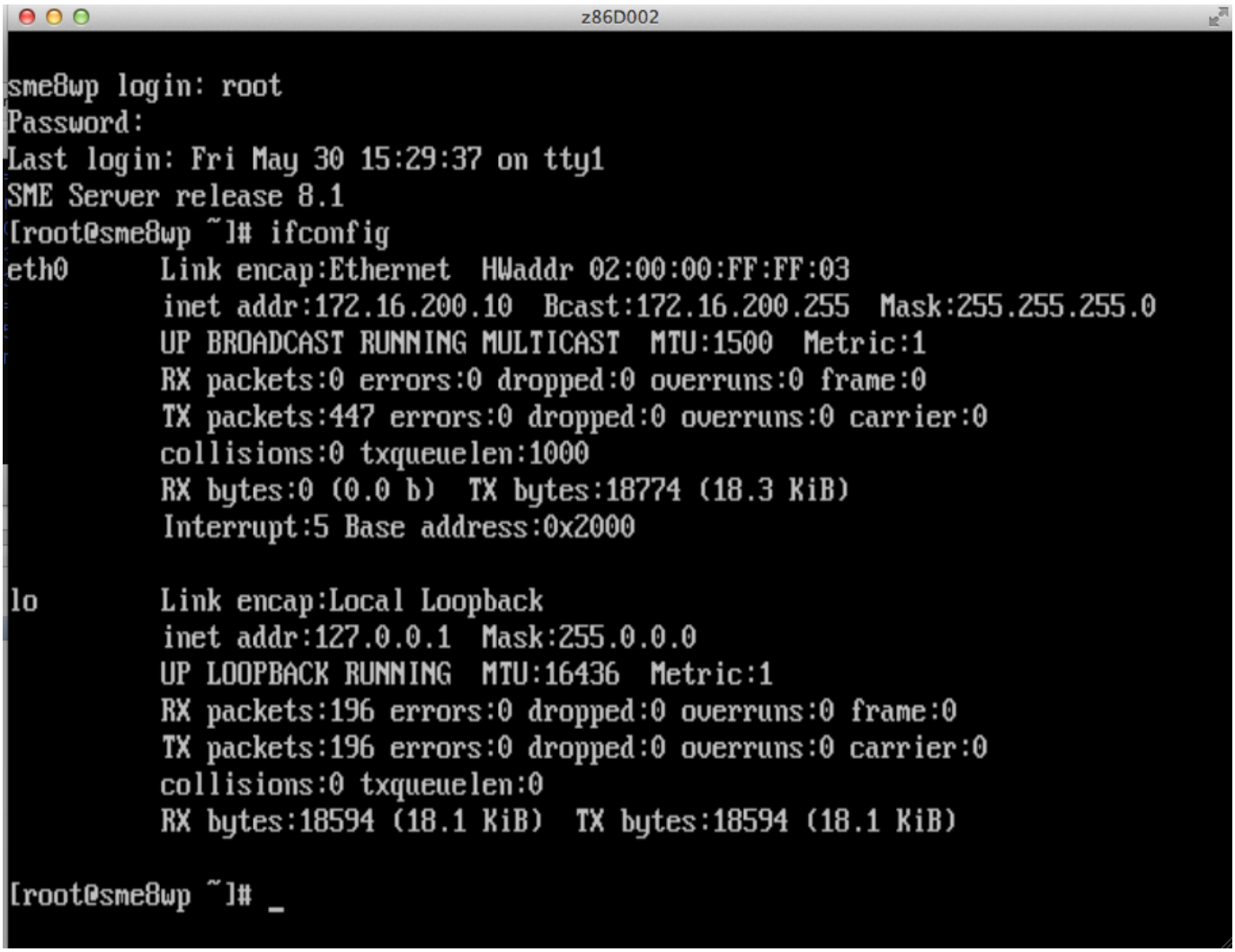
You'll need to shutdown the server at this point. Enter: **`Shutdown -h now`**



You'll need to go to either the z86MNT userid and issue **`z86 (start=sme8`** or you can logon to user SME8 and issue **`CP IPL 194`** . Watch during boot up to ensure that the eth0 connection is now OK.

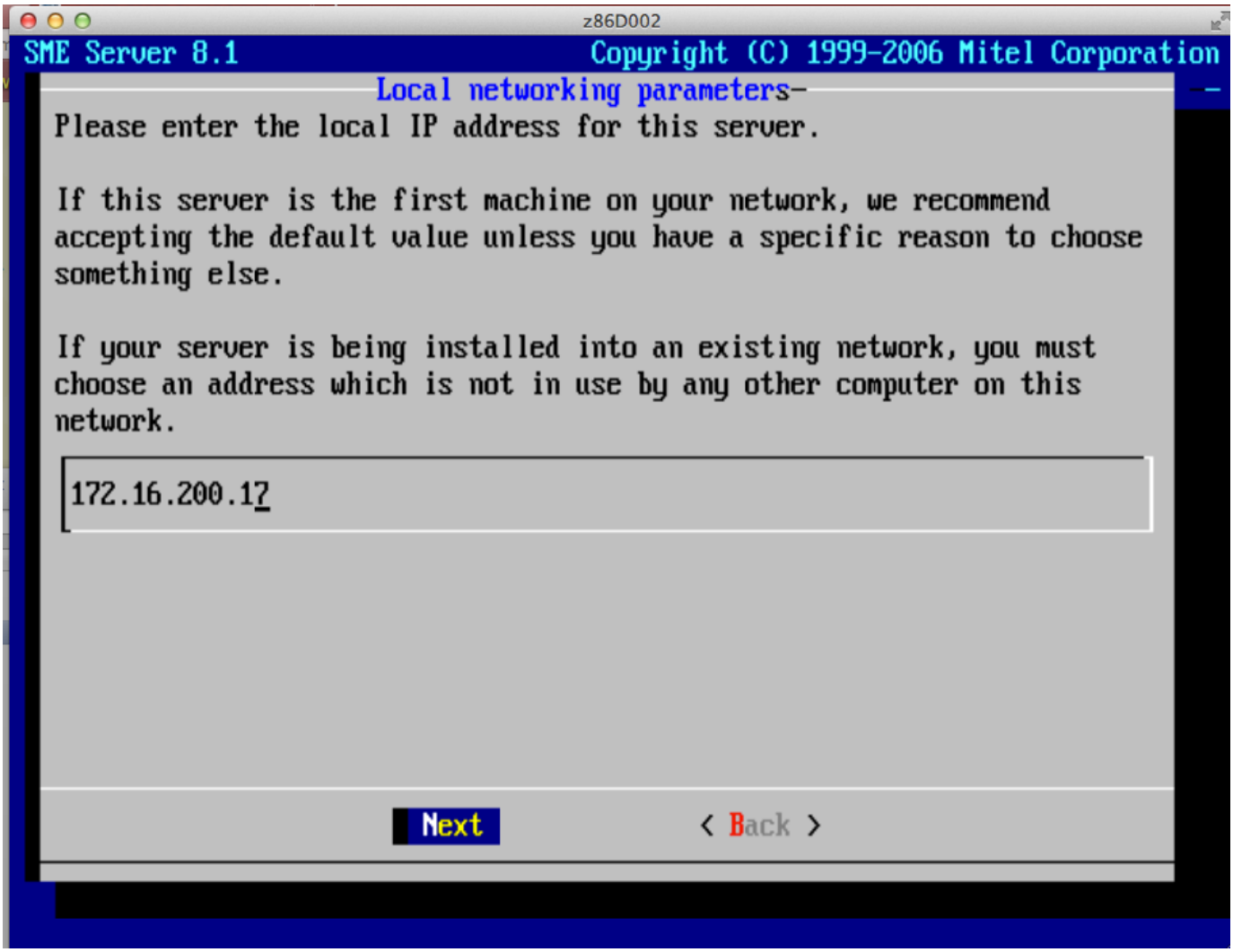


Logon again with user: root and password: Noodle01.
Enter: **`ifconfig`** to check that the network is connected properly:



Please note, you are only checking to see that the HWADDR or MAC address is correct now. While the IP address is correct, it may not be operational because it is incorrect in other files.

In order to correct the IP address in all places, please logoff the root user by entering: exit at a command line.
At the logon prompt, sign in as user: admin with password: Noodle01
This will bring you to the SME Admin console. Go through the configuration changes as outlined in ***Starting the SME Server*** section. As you may recall, we changed the IP address from 172.16.200.17 to 172.16.200.10 in the ifcfg-eth0 file. As you can see below, that change was not sufficient.



Change this to the proper IP address for your environment. In this case, it would be changed to 172.16.200.10
Complete the configuration as directed and re-boot the system as described earlier for the changes to become permanent.

When you are done with this session, you should save this system image to avoid future changes in case you need to re-install any system images.
To shutdown the SME image from a console window enter: **`shutdown -h now`**

Once shutdown is complete, go to the TN3270 session on z/VM associated with SME8 and logoff that system image. You must log off. Disconnect is not sufficient.

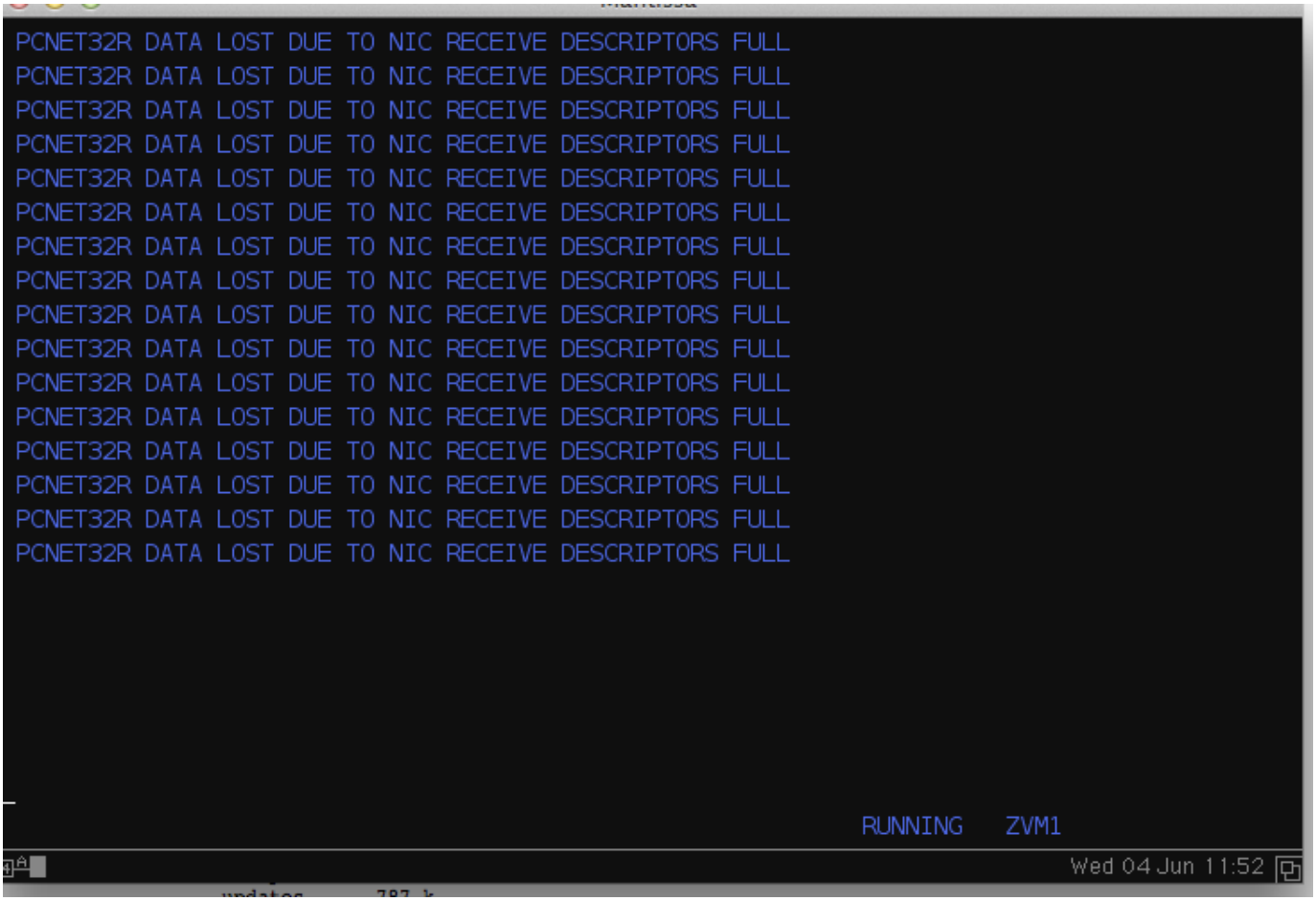
Then, logon to the z86MNT userid
Enter: **`zvcopy z86io SME8 CMS sme.img SMENEW1 IMG A`**

You may also want to update your PROFILE Z86 file to point to this new operating system image (**`vmimage = SMENEW1 IMG A`**) instead of the original copy that was used.

Data Lost Due to NIC receive descriptors full

** Added 06/04/14

On the maintenance user or even the IO server or z86 guest user, you may see a message that repeats often that looks like this:

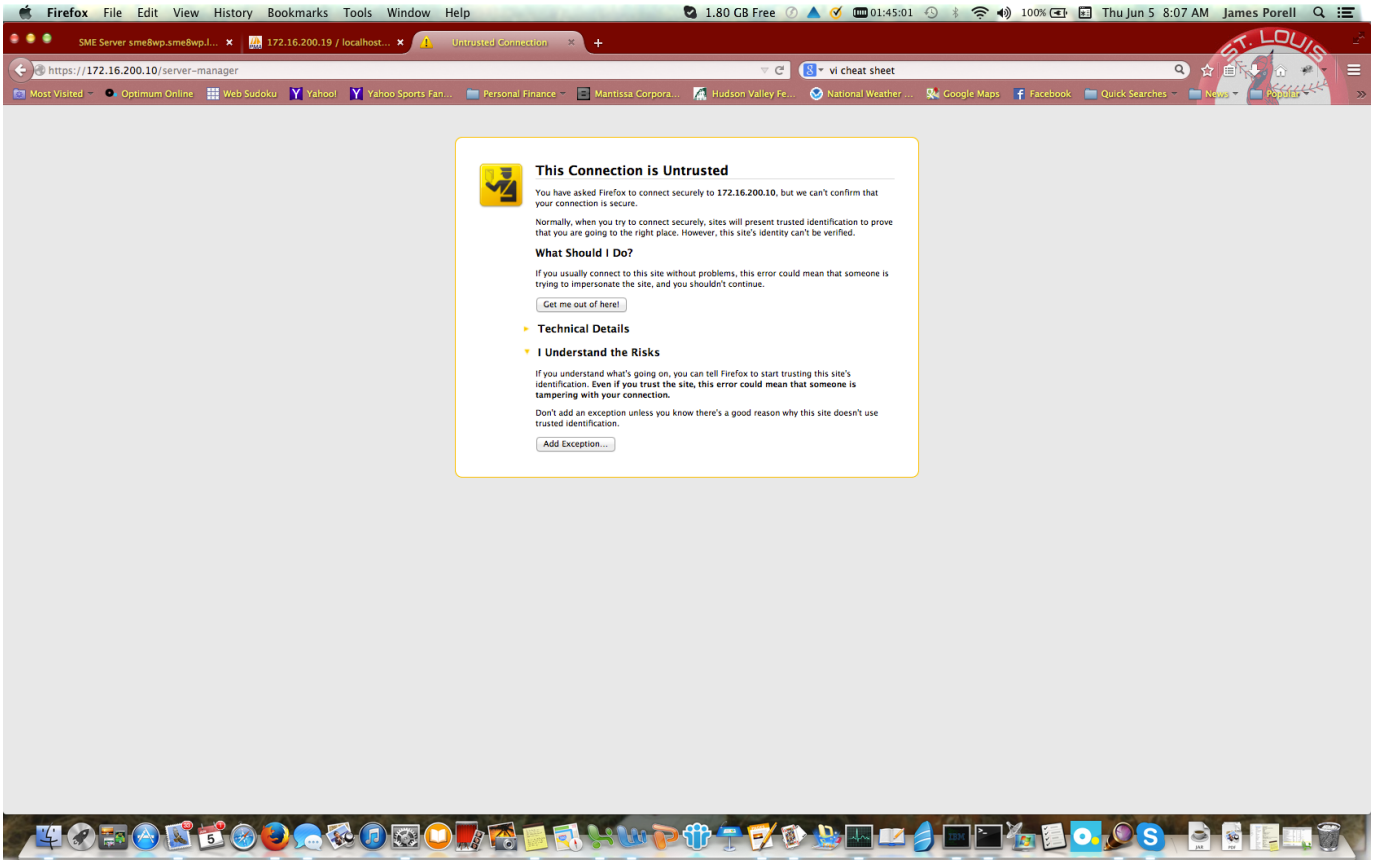


Do not be alarmed, this really isn't a failure. It is a message that was put in during the development process to let us know when buffers overflow due to heavily used network traffic by a single guest or multiple guests. The data request is actually rerun and should process successfully. As a new user, I was concerned that this was an error that I should do something about, but our development team assured me to "Remain Calm".

Connection is not trusted

** Added 06/05/14

The first time you boot up your SME Server Manager (http://<your IP @>/server-manager) you may be presented with a “scary” security error message. You are getting this because the system image was built on one hardware platform and then moved to another platform for operations.



Click on Add Exception. You’ll be taken to another screen where it asks you to save this exception. Do that and you should be good from now on.

We are extremely security conscious at Mantissa. This is beta code. If it were production level code, we’d be setting up a unique certificate for this image.

Network doesn't work for either DSL or SME image when both are started

**Updated 06/05/14
This could be a conflict because they are both assuming they have the same MAC address and only the first one to boot with it will be able to operate. For example, the DSL is up and running with a valid IP @ and operates properly. The SME image may appear to be up and running properly. The IFCONFIG command could even give you a valid eth0 port with a valid IP @. But, unfortunately it is an island in the network. It cannot get to other systems in the network and they cannot get access to it.

This requires that each OS image, or in reality, each z86VM guest image under z/VM have a unique MAC address. The solution requires changes to the VM definition of the IO server as well as updates to the PROFILE Z86 for each z86VM user. Please see ***3. Define z86 user ids*** for the proper set up.

Poor Performance of Sudoku results - results of bad z/VM paging set up

One reason to capture performance results from the Beta experience is to look at z/VM system set up to make sure it is optimized. One of the beta customers had a terrible Performance test result on a z196. It was five times worse than the Mantissa z114. That's not supposed to be possible.
Upon examination of the paging subsystem, there were many problems found. The following reference page shows a top list of things to consider in your paging subsystem to ensure that z/VM is performing optimally.

Configuring a z/VM System to Page Well

Here are some guidelines for how to set up a z/VM paging system to give it every advantage in supporting your workload.

1. Define about 25% of the partition's total storage as XSTORE, up to a maximum of 2 to 4 GB. This is a general rule of thumb. We will give you some XSTORE tuning tips later on.
2. Use enough paging packs so that the packs run no more than about 50% full. As a rough start for this, calculate the total size of your logged-on guests, plus their VDISKs, plus their shared data spaces. Subtract the total storage for the partition. Then double that number. Roughly, the answer is how much page space you will need in the worst case. You will find this calculation sizes your paging space quite generously. After you run for a while, you might decide to add or remove some paging volumes. That's fine.
3. Remember that paging well is all about being able to run more than one paging I/O at a time. This means you should spread your paging space over as many volumes as possible. Get yourself lots of little paging volumes, instead of one or two big ones. The more paging volumes you provide, the more paging I/Os z/VM can run concurrently.
4. Make all of your volumes the same size. Use all 3390-3s, or 3390-9s, or whatever. When the volumes are unequally sized, the smaller ones fill and thereby become ineligible as targets for page-outs, thus restricting z/VM's opportunity for paging I/O concurrency.
5. A disk volume should be either all paging (cylinders 1 to END) or no paging at all. Never allocate paging space on a volume that also holds other kinds of data, such as spool space or user minidisks.
6. Think carefully about which of your DASD subsystems you choose for paging. Maybe you have DASD controllers of vastly different speeds, or cache sizes, or existing loads. When you decide where to place paging volumes, take the DASD subsystems' capabilities and existing loads into account.
7. Within a given DASD controller, volume performance is generally sensitive to how the volumes are placed. Work with your DASD people to avoid poor volume placement, such as putting all of your paging volumes into one rank.
8. If you can avoid ESCON chpids for paging, do it. An ESCON chpid can carry only one I/O at a time. FICON chpids can run multiple I/Os concurrently: 32 or 64, depending on the generation of the FICON card.
9. If you can, run multiple chpids to each DASD controller that holds paging volumes. Consider two, or four, or eight chpids per controller. Do this even if you are using FICON.
10. If you have FCP chpids and SCSI DASD controllers, you might consider exploiting them for paging. A SCSI LUN defined to the z/VM system as an EDEV and ATTACHed to SYSTEM for paging has the very nice property that the z/VM Control Program can overlap I/Os to it. This lets you achieve paging I/O concurrency without needing multiple volumes. However, don't run this configuration if you are CPU-constrained. It takes more CPU cycles per I/O to do EDEV I/O than it does to do classic ECKD I/O.
11. Make sure you run with a few reserved slots in the CP-owned list, so you can add paging volumes without an IPL if the need arises.

This material was taken from an IBM z/VM web page.

VNC Connection Failures

There are a number of connection failures that are possible. Some may be bugs in z86VM code, some may be bugs in the VNC code. Some are unique to the VNC client being utilized. We'll try to tackle a couple of them. Not all experiences will be similar to these.

We'll start with using the TIGHTVNC client.

Socket Read errors or Socket Write errors

We shipped the latest version of tighvnc with z86VM Beta 3. We had a customer that used the version from Beta 2, which was downlevel. As that is a Java application, it seemed to have it's own set of issues when attempting to connect. By upgrading, via the internet, to the latest version of tightvnc, those issues went away.

Connection not made

When initially attempt to connect to SME8 or DEMODSL, make sure of two things:

1. IP address in the VNC session is pointing to the IP @ of the z/VM operating system. Once that boots up, you should be able to switch to an x86 hosted VNC session.
2. Make sure the port address is using what is in PROFILE Z86. Our examples are 1999 and 2001. Don't use the industry standard 5900 that comes with the product.

Connection made, but the screen is gray or black

This is typically an issue with incorrect screen resolution or incompatible screen resolution. On the menu bar, typically at the top of the screen, go to the View option. You can specify Actual size first. If the characters appear, you should be in business. Once they appear, you can resize the screen successfully.

If that doesn't work, sign on to the z/VM Guest you are trying to connect with, for example, SME8 or DSLDEMO. At the command prompt, issue VNC RESET. The VNC connection will close. Re-open it via the VNC Client. It should be working now.

VNC Screen Frozen

The text is there, but no commands can be input. This author has hit that problem a couple of times. It was kind of like trying to jump start a car….a tease that it will work, since the screen appears correct, I've tried VNC RESET and it comes to the same place. At this point, my only success has been to re-boot the image. I've just listed this as it appears to be an

aging problem. The VNC session has opened and closed numerous times until it can no longer accept input again.

Desire to run Multiple VNC windows on a single system.

Tightvnc application only allows a single connection at a time from it's Java based window. Within Mantissa, we have users that open up terminal sessions and type in tightvnc & in order to get additional background windows started.
We've seen people try to change the IP address of the z/VM host, though, to the IP address of the DSL window. That's not going to work with the VNC port that was specified in the PROFILE Z86.

Only a single VNC client per VNC Port specified in PROFILE Z86

The VNC Server, shipped with z86VM, was only intended as a stop gap measure until the IP network was running and other presentation viewers could be started natively within the x86 environment, such as vncserver, RDP, NX and SPICE protocols.
As a result, there is a limit of one VNC session per operating system image running under a z86VM guest.
If you desire multiple windows to a single x86 OS guest, then you'll need to start the vnc-server for that guest. You should be able to connect to that guest via it's unique IP address (not the z/VM operating systems IP @). and ports 5900 to 5903.

Commercial VNC Clients

There are several advantages of paid for VNC clients.

1. Profiling - multiple, different set ups that are point and click to open them
2. Multiple windows open at once
3. Patch fixes and support for problems.

We are not saying that you must use one of these products. Anyone that runs multiple VNC windows at once, though, will see great usability benefits.