



Gregory Geiselhart  
Tung-Sing Chong  
Michael Donovan

# Linux on IBM server zSeries and S/390: Cloning Linux Images in z/VM

## Preface

There are many ways to clone multiple Linux for @server zSeries images under a z/VM system. This Redpaper discusses and expands on one example of this process, and is intended to help customers create many Linux for zSeries systems in a repeatable manner, with minimal effort.

## Objectives

The paper has the following objectives:

- ▶ Show the details of our cloning environment
- ▶ Describe the cloning process
- ▶ Provide an example of our cloning process
- ▶ Provide samples of the code used to clone Linux for zSeries images

# Introduction

With the increasing interest in Linux for zSeries under z/VM, systems administrators may be faced with the need to create dozens (or even hundreds) of new Linux guest machines. As you can imagine, the process of repeatedly installing similar Linux images from scratch can quickly become tedious, so z/VM systems administrators need a simple, efficient, and repeatable process for creating these images on a z/VM system.

For this paper, we chose to examine a cloning process that was originally developed for a Linux World convention several years ago. The “Linux Images Cloning Demo” created multiple copies of TurboLinux 6.5.1, SuSE SLES-7.0, and RedHat 7.2 images. The demonstration package also used Inter-User Communication Vehicle (IUCV) links as point-to-point network connections from the cloning process and the newly created cloned images. The original package documentation and code can be found at:

<http://www.vm.ibm.com/devpages/chongts>

In the paper we only discuss an environment for defining and cloning multiple images of SuSE SLES-7.0 pro. Also, we have converted the IUCV network model of the original demonstration package to a network model based on z/VM Guest Local Area Network (VM LAN) support.

# Methodology

As mentioned, there are many different ways to clone Linux images under z/VM. In our case, we decided to use z/VM functions and facilities as much as possible. This cloning process uses the following:

- ▶ The CMS REXX interpreter
- ▶ The z/VM Directory Maintenance Facility (DirMaint)
- ▶ The z/VM DASD Dump Restore (DDR) utility
- ▶ z/VM TCP/IP stacks
- ▶ The GA level of SuSE SLES7 pro
- ▶ VM LAN as point-to-point network connections

No additional Linux packages are required, other than what is shipped with the GA Linux distributions. No modification of the GA levels of the Linux distributions is required, so there's no need to rebuild the Linux kernel. This cloning process uses DirMaint to manage the z/VM user directory, thus avoiding the need to directly modify any part of the z/VM system.

The cloning demonstration consists of two parts:

- ▶ An application which, when started, will continuously clone up to 249 Linux images (we refer to this as *continuous cloning*). A Web interface is used to monitor status as the application progresses.
- ▶ An application which will clone a single Linux image (we refer to this as *on-demand cloning*). The application is initiated using a Web interface.

## z/VM system requirements

Before beginning our cloning adventure, we need to prepare our z/VM system for an increasing number of Linux guest virtual machines. We need to do a bit of capacity planning for the number of images we might create, and we also need to make certain we are licensed to use DirMaint.

## Paging, SPOOL, and DASD pool considerations

During the development of this paper, and in the experiences from past work with the demonstration package, we compiled several “rules of thumb” for including and defining paging and SPOOL space and defining the DirMaint DASD pool:

- ▶ For approximately each 40 to 45 Linux images, add one 3390-3 for paging space. Eight 3390-3 packs should suffice for about 300 images. Ensure all the paging volumes are added to the SYSTEM CONFIG file, which is found on the MAINT CF1 disk.
- ▶ You need 500 cylinders of 3390 space for each Linux image with a shared read-only (R/O) /usr and /usr/src and DASD for swap. The 500 cylinders break down to:
  - 300 cylinders for a read-write (R/W) /
  - 200 cylinders for swap

One 3390-3 pack can contain 6 cloned images.

- ▶ You need 300 cylinders of 3390 space for each Linux image with a shared R/O /usr and /usr/src and using VDISK for swap. This is 300 cylinders for a R/W /.

One 3390-3 pack can contain 10 to 11 cloned images.

- ▶ You need 1000 cylinders of 3390 space for each Linux image with a shared R/O /usr and a R/W /usr/src. The 800 cylinders break down to:
  - 300 cylinders for a R/W /
  - 500 cylinders for a R/W /usr/src
  - 200 cylinders for swap.

One 3390-3 pack can clone 3 images.

The consoles of the LXOPR, LXMSTR, LXADMIN, and LXCNTL virtual machines are spooled for debugging purposes. You should add 300 cylinders for SPOOL space to hold these console logs.

## DirMaint considerations

DirMaint is preinstalled on all z/VM Version 4 systems; however, it comes disabled on the preinstalled z/VM system. To use this cloning process as documented, you must set DirMaint to an enabled state. Refer to the *Program Directory for IBM z/VM Directory Maintenance Facility Feature* to enable DirMaint and place DirMaint in production. You also need to have APAR VM63033 installed if you are running with z/VM 4.3.0.

## Virtual machine definitions

The cloning process described in this paper requires some VM user definitions. We classify these users according to the role they play in the process. The general category of VM user definitions and the role those users play can be summarized as follows:

<b>Master virtual machines</b>	These manage the resources (DASD and network configuration) needed to clone Linux images.
<b>Controller virtual machines</b>	These control the operation of the cloning process.
<b>Cloned Linux virtual machines</b>	These form the virtual machine for the cloned Linux guests.
<b>TCP/IP stack virtual machines</b>	These define the network topology for the cloning environment.

In the following sections, we describe the virtual machine definitions in some detail.

## The LXDFLT directory profile

In order to simplify virtual machine definition, a profile containing common virtual machine options and definitions is used. The profile definition (LXDFLT DIRECT) is shown in Example 1 on page 5.

### Example 1 The LXDFLT DIRECT profile

---

```
PROFILE LXDFLT
  IPL      CMS PARM AUTOCR
  IUCV     ALLOW
  MACHINE  XA
  OPTION   QUICKDSP
  CONSOLE  0009      3215 T      LXCNTL
  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
  LINK     TCPMAINT  0592 0592  RR
```

---

Following are explanations of the detail shown in the example:

1. Start CMS when the virtual machine is started.
2. Authorize the virtual machine to create an inter-user communications vehicle (IUCV) communication path to another virtual machine.
3. Designate the virtual machine as capable of simulating either ESA/390 architecture or z/Architecture.
4. Enable the virtual machine to be added immediately to the dispatch list whenever it has work to perform.
5. Spool console output to the LXCNTL virtual machine.
6. Definitions for a virtual reader, printer, and punch.
7. Read-only links to MAINT user disks for access to system commands.
8. Read-only links to TCPMAINT user disks for access to TCP/IP commands.

## The master virtual machine definitions

The cloning process relies on three master virtual machines to manage resources to be allocated for the cloned Linux guests. (Master users run primarily as Linux guests.)

<b>LXMNT</b>	This manages the minidisks containing the Linux installations to be cloned.
<b>LXMSTR</b>	This manages IP address assignment for the cloned Linux images.
<b>LXDISP</b>	This runs an instance of the Apache Web server to display the number of images created during continuous cloning.

## The LXMNT virtual machine

The LXMNT virtual machine defines minidisks used to install a Linux guest. These minidisks serve as the master copy of a Linux image for subsequent cloning. Three different types of Linux clones are supported using this user's minidisk allocation. The LXMNT virtual machine definition is shown in Example 2.

### Example 2 The LXMNT DIRECT profile

---

```
USER LXMNT LXMNT 128M 128M G
  INCLUDE LXDFLT
  MDISK 0191 3390 1101 100 LX651A MR ALL 1
  *****
  * Only SuSE master image is used *
  * 0151 is "/" and it is for R/O "usr" and "/usr/src" *
  * 0153 is "/usr" *
  * 0154 is "/usr/src" *
  * 0155 is swap *
  * 5251 is "/" for R/O "usr" and R/W "/usr/src" *
  *****
  MDISK 0151 3390 0301 0300 LX651A MR ALL 2
  MDISK 0153 3390 3539 3138 LX3DA1 MR ALL 3
  MDISK 0154 3390 2151 0500 LX651A MR ALL 4
  MDISK 0155 3390 3339 0200 LX3DA1 MR ALL 5
  MDISK 5251 3390 0601 0300 LX651A MR ALL 6
  SPECIAL 0700 QDIO 3 SYSTEM DEMOCONT 7
```

---

Following are explanations of the detail shown in the example:

1. Defines a 191 minidisk used for the A-disk CMS guest.
2. Defines a 151 minidisk used for the / filesystem of the Linux guest.
3. Defines a 153 minidisk used for the /usr filesystem of the Linux guest.
4. Defines a 154 minidisk used for the /usr/src filesystem of the Linux guest.
5. Defines a 155 minidisk used for the swap device of the Linux guest.
6. Defines a 5251 minidisk to be used as a DDR copy of the 151 minidisk.
7. Defines a simulated QDIO NIC using virtual devices 0700, 0701, and 0702. The NIC is coupled to the persistent VM Guest LAN named DEMOCONT.

## The LXMSTR virtual machine

The LXMSTR virtual machine manages IP address assignments for cloned Linux images. This virtual machine has read-write access to the master copy of the Linux installation minidisks (owned by LXMNT). The LXMSTR virtual machine definition is shown in Example 3 on page 7.

### Example 3 The LXMSTR DIRECT profile

---

```
USER LXMSTR LXMSTR 128M 256M G
INCLUDE LXDFLT
*****
* Link to Linux master images R/W "/"s *
*****
* 1251 is the R/W "/" for SuSE images that have R/O "/usr"
LINK LXMNT 0151 1251 MR
* 5251 is the R/W "/" for SuSE images that have R/W "/usr/src"
LINK LXMNT 5251 5251 MR
*
MDISK 0191 3390 3275 0025 430W01 MR ALL
*****
* 0151 is LXMSTR's R/W "/" *
* 0153 is LXMSTR's R/W "/usr" *
* 0154 is LXMSTR's R/W "/usr/src" *
* 0155 is LXMSTR's swap *
*****
MDISK 0151 3390 0001 0300 LX651A MR ALL
MDISK 0153 3390 6877 3138 LX3CA3 MR ALL
MDISK 0154 3390 1351 0500 LX651A MR ALL
MDISK 0155 3390 6677 0200 LX3CA3 MR ALL
SPECIAL 0700 QDIO 3 SYSTEM DEMOCONT
```

Following are explanations of the detail shown in the example:

1. Link to the 151 minidisk of the LXMNT user in read/write mode.
2. Link to the 5251 minidisk of the LXMNT user in read/write mode
3. Defines a 191 minidisk used for the A-disk CMS guest.
4. Defines a 151 minidisk used for the / filesystem of the Linux guest.
5. Defines a 153 minidisk used for the /usr filesystem of the Linux guest.
6. Defines a 154 minidisk used for the /usr/src filesystem of the Linux guest.
7. Defines a 155 minidisk used for the swap device of the Linux guest.
8. Defines a simulated QDIO NIC using virtual devices 0700, 0701, and 0702.  
The NIC is coupled to the persistent VM Guest LAN named DEMOCONT.

### The LXDISP virtual machine

The LXDISP user serves as a Linux guest running an Apache server. As clones are produced during the continuous cloning process, the Web server will track the current count of clones produced. In Example 4 on page 8, we show the LXDISP virtual machine definition.

#### Example 4 The LXDISP DIRECT profile

---

```
USER LXDISP LXDISP 128M 128M ABCG
  INCLUDE LXDFLT
  SPECIAL 0700 QDIO 3 SYSTEM DEMOCONT 1
  MDISK 0191 3390 3300 0025 430W01 MR READPASS
  MDISK 0151 3390 0201 3138 LX3DA1 MR READPASS WRITPASS MULTPASS 2
  MDISK 0155 3390 0001 0200 LX3DA1 MR READPASS WRITPASS MULTPASS
```

---

Following are explanations of the detail shown in the example:

1. Defines a simulated QDIO NIC using virtual devices 0700, 0701, and 0702. The NIC is coupled to the persistent VM Guest LAN named DEMOCONT.
2. Defines 151 and 155 minidisks for the Linux / filesystem and swap device, respectively.

## The controller virtual machine definitions

The controller virtual machines manage overall cloning operation. Three users comprise the controller:

<b>LNXCNTL</b>	This acts as a secondary console for all Linux images (including all cloned Linux guests and master VM users running as Linux guests).
<b>LXOPR</b>	This controls the continuous cloning aspect. This virtual machine owns the code necessary to clone Linux images.
<b>LXADMIN</b>	This controls the on-demand cloning aspect. This virtual machine runs an instance of the Webshare Web server to initiate on-demand cloning.

Controller virtual machine operate as CMS guests.

### The LXCNTL virtual machine

Console output produced by the Linux guests is directed to the LXCNTL virtual machine console. This allows for both monitoring of a Linux guest, and communication with a Linux guest should the need arise.

For more information on how to set up and use a secondary console, see the section on “Single Console Image Facility” in *z/VM V4R3.0 Virtual Machine Operation*, SC24-6036. Example 5 on page 9 illustrates the LXCNTL virtual machine definition.



### *Example 5 The LXCNTL DIRECT profile*

---

```
USER LXCNTL LXCNTL 128M 128M ABG
  INCLUDE LXDFLT
  SHARE RELATIVE 1000
  MDISK 0191 3390 0151 02 LNXU1R MR ALL
```

---

## **The LXOPR virtual machine**

The code needed to run the cloning demonstration resides on minidisks owned by the LXOPR virtual machine. This virtual machine controls the continuous cloning process. In Example 6, we show the LXOPR virtual machine definition.

### *Example 6 The LXOPR DIRECT profile*

---

```
USER LXOPR LXOPR 16M 128M ABCG
  INCLUDE IBMDFLT
  IPL CMS
  IUCV ALLOW
  MACHINE ESA
  OPTION QUICKDSP
  SHARE RELATIVE 1000
  LINK MAINT 051D 051D MR
  LINK MAINT 05E5 05E5 RR
  MDISK 0191 3390 0611 0050 LNXU1R MR ALL
  * 7191 for SuSE 191 with VDISK for swap
  MDISK 7191 3390 0661 0002 LNXU1R MR ALL 1
  * 8191 for SuSE 191 with DASD for swap
  MDISK 8191 3390 0663 0002 LNXU1R MR ALL 2
```

---

Following are explanations of the detail shown in the example:

1. Defines a 7191 minidisk for use as a CMS A-disk (virtual address 191) by Linux clones utilizing VDISK swap
2. Defines a 8191 minidisk for use as a CMS A-disk (virtual address 191) by Linux clones utilizing DASD swap

## **The LXADMIN virtual machine**

This LXADMIN virtual machine controls the on-demand cloning process. Cloning is initiated through a Web interface provided by a Webshare instance running in this virtual machine.

Cloning requests are forwarded from the Webshare interface to LXADMIN for processing. The LXADMIN virtual machine definition is shown in Example 7 on page 10.

*Example 7 The LXADMIN DIRECT profile*

---

```
USER LXADMIN LXADMIN 6M 128M ABCG
  INCLUDE IBMDFLT
  IPL      CMS
  IUCV    ALLOW
  MACHINE ESA
  SHARE   RELATIVE 1000
  LINK    MAINT    051D 051D MR
  LINK    MAINT    05E5 05E5 RR
  MDISK   0191     3390 0665 50 LNXU!R MR ALL
```

---

## The Linux clone virtual machine definitions

The cloning demonstration supports three types of Linux clones flavors (all based on a SuSE 7.0 SLES Professional distribution):

- LXSUSEI**            A clone consisting of a 200 cylinder 3390 DASD swap device with a shared, read-only /usr/src filesystem.
- LXSUSEIV**        A clone consisting of a VDISK swap device with a shared, read-only /usr/src filesystem.
- LXSUSERW**        A clone consisting of a 200 cylinder 3390 DASD swap device with a non-shared, read-write /usr/src filesystem.

**Note:** All three clone flavors share a common read-only /usr filesystem.

In order to simplify the cloning process, we use virtual machine prototypes to assist in dynamically creating the clone virtual machines.

### The LXSUSEI prototype

The LXSUSEI clone prototype is shown in Example 8.

*Example 8 The LXSUSEI PROTODIR profile*

---

```
USER LXUSER LXUSER 64M 128M G
  INCLUDE LXDFLT
  LINK LXMNT 0153 0153 RR
  LINK LXMNT 0154 0154 RR
  LINK LXOPR 8191 0191 RR
  AMDISK 0151 3390 AUTOG 300 LXDEMO MR PWS ALL LXWR LXMU
  AMDISK 0155 3390 AUTOG 200 LXDEMO MR PWS ALL LXWR LXMU
```

---

1  
2  
3  
4  
5

Following are explanations of the detail shown in the example:

1. Defines a read-only link to the LXMNT 0153 minidisk for the Linux guest /usr filesystem
2. Defines a read-only link to the LXMNT 0154 minidisk for the Linux guest /usr/src filesystem
3. Defines a read-only link to the LXOPR 8191 minidisk for the CMS guest A-disk
4. Defines a read-write minidisk allocation for the Linux guest / filesystem
5. Defines a read-write minidisk allocation for the Linux guest swap device

### The LXSUSEIV prototype

The LXSUSEIV clone prototype is shown in Example 9.

*Example 9 The LXSUSEIV PROTODIR profile*

---

```
USER LXUSER LXUSER 64M 128M G
  INCLUDE LXDFLT
  LINK LXMNT 0153 0153 RR
  LINK LXMNT 0154 0154 RR
  LINK LXOPR 7191 0191 RR
  AMDISK 0151 3390 AUTOG 300 LXDEMO MR PWS ALL LXWR LXMU
```

---

1  
2  
3  
4

Following are explanations of the detail shown in the example:

1. Defines a read-only link to the LXMNT 0153 minidisk for the Linux guest /usr filesystem
2. Defines a read-only link to the LXMNT 0154 minidisk for the Linux guest /usr/src filesystem
3. Defines a read-only link to the LXOPR 7191 minidisk for the CMS guest A-disk
4. Defines a read-write minidisk allocation for the Linux guest / filesystem

### The LXSUSERW prototype

The LXSUSERW clone prototype is shown in Example 10 on page 12.

*Example 10 The LXSUSERW PROTODIR profile*

---

```
USER LXUSER LXUSER 64M 128M G
  INCLUDE LXDFLT
  LINK LXMNT 0153 0153 RR
  LINK LXOPR 8191 0191 RR
  AMDISK 0151 3390 AUTOG 300 LXDEMO MR PWS ALL LXWR LXMU
  AMDISK 0154 3390 AUTOG 500 LXDEMO MR PWS ALL LXWR LXMU
  AMDISK 0155 3390 AUTOG 200 LXDEMO MR PWS ALL LXWR LXMU
```

---

Following are explanations of the detail shown in the example:

1. Defines a read-only link to the LXMNT 0153 minidisk for the Linux guest /usr filesystem
2. Defines a read-only link to the LXOPR 8191 minidisk for the CMS guest A-disk
3. Defines a read-write minidisk allocation for the Linux guest / filesystem
4. Defines a read-write minidisk allocation for the Linux guest /usr/src filesystem
5. Defines a read-write minidisk allocation for the Linux guest swap device

## The alternate TCP/IP stack virtual machine definitions

The demonstration uses several VM Guest LANs in its network topology. We define two virtual machines to manage two of those networks (see “Network overview” on page 13 for a description of the network topology).

### The TCPIP111 virtual machine

The TCPIP111 virtual machine is the z/VM TCP/IP stack for the DEMOCONT network shown in Figure 1 on page 14. We show the virtual machine definition in Example 11.

*Example 11 The TCPIP111 DIRECT profile*

---

```
USER TCPIP111 TCPIP111 128M 256M ABG
  INCLUDE TCPCMSU
  IUCV ALLOW
  IUCV ANY PRIORITY
  IUCV *CCS PRIORITY MSGLIMIT 255
  OPTION QUICKDSP SVMSTAT MAXCONN 1024 DIAG98 APPLMON
  SHARE RELATIVE 3000
  LINK TCPMAINT 0591 0591 RR
  LINK TCPMAINT 0592 0592 RR
  LINK TCPMAINT 0298 0198 RR
  MDISK 0191 3390 0021 0005 LNXU1R MR RTCP/IP WTCPIP MTCPIP
```

---

## The TCPIP112 virtual machine

The TCPIP112 virtual machine is the z/VM TCP/IP stack for the DEMOPICK network shown in Figure 1 on page 14. We show the virtual machine definition in Example 11.

### *Example 12 The TCPIP112 DIRECT profile*

---

```
USER TCPIP112 TCPIP112 128M 256M ABG
INCLUDE TCPCMSU
IUCV ALLOW
IUCV ANY PRIORITY
IUCV *CCS PRIORITY MSGLIMIT 255
OPTION QUICKDSP SVMSTAT MAXCONN 1024 DIAG98 APPLMON
SHARE RELATIVE 3000
LINK TCPMAINT 0591 0591 RR
LINK TCPMAINT 0592 0592 RR
LINK TCPMAINT 0398 0198 RR
MDISK 0191 3390 0606 0005 LNXU1R MR RTCPIP WTCPIP MTCPIP
```

---

## TCP/IP stack configuration

The cloning demonstration defines three VM Guest LAN networks:

<b>DEMOCNT</b>	The network on which Linux images produced by the continuous cloning process will be connected. The TCPIP111 user acts as this network's TCP/IP stack.
<b>DEMOPICK</b>	The network on which Linux images produced by the on-demand cloning process will be connected. The TCPIP112 user acts as this network's TCP/IP stack.
<b>DEMOTCP</b>	A network to tie DEMOCNT and DEMOPICK to the outside. The TCPIP user acts as this network's TCP/IP stack.

## Network overview

In Figure 1 on page 14, we illustrate the demonstration's network topology.

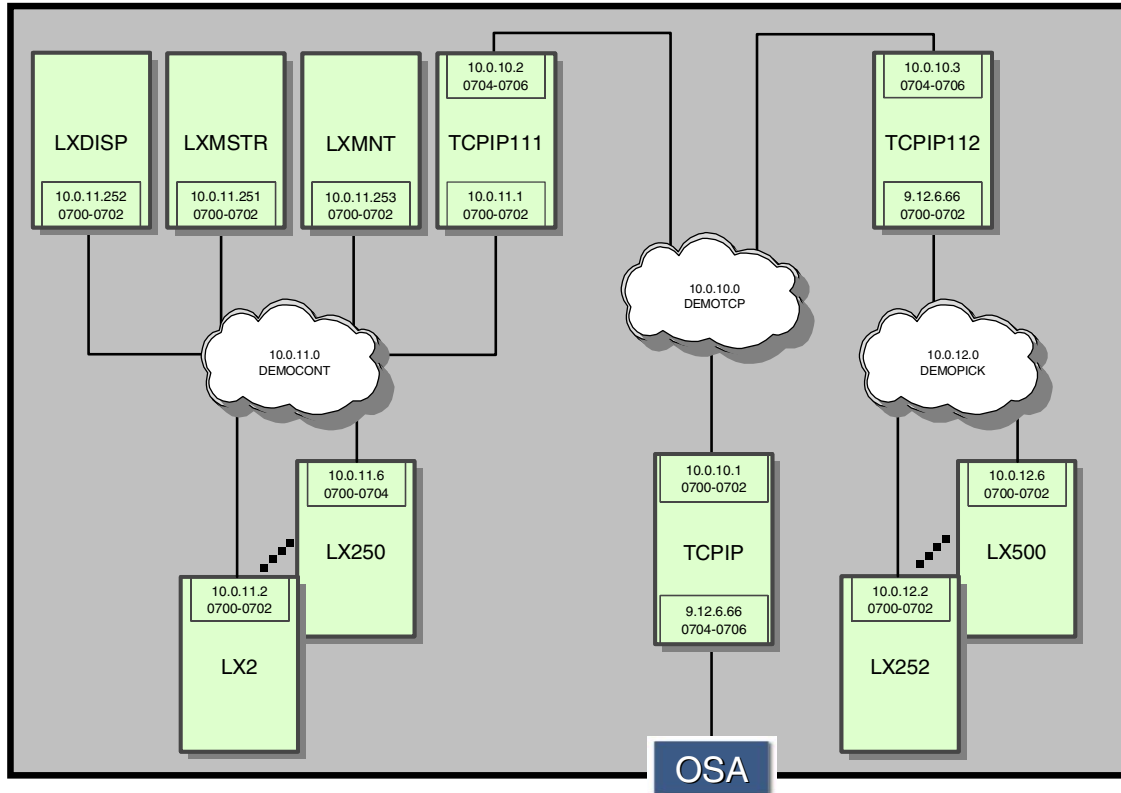


Figure 1 Network topology

Linux images produced by the continuous cloning process run in virtual machines LNX2 through LNX250 and are connected to the DEMOCONT LAN. Linux images produced by the on-demand cloning process run in virtual machines LNX252 through LNX500 and are connected to the DEMOPICK LAN.

## The VM Guest LAN definitions

To create a VM Guest LAN, we use the **DEFINE LAN** command. In this situation, the LAN should be configured as *persistent* (that is, the LAN should remain defined even if the owner has logged off and all guests disconnected). We use QDIO persistent Guest LANs. To define three LANs, use these commands:

```
DEFINE LAN DEMOTCP OWNERID SYSTEM TYPE QDIO
DEFINE LAN DEMOCONT OWNERID SYSTEM TYPE QDIO
DEFINE LAN DEMOPICK OWNERID SYSTEM TYPE QDIO
```

**Tip:** To make the Guest LAN definitions permanent across IPLs, add the commands to the SYSTEM CONFIG file.

**Note:** Definition of a persistent LAN requires class B privilege - issue the commands from the appropriate VM user (such as MAINT). For more information on the **DEFINE LAN** command, refer to *z/VM V4R3.0 CP Planning and Administration*, SC24-6043.

## TCPMAINT virtual machine modifications

To centralize management of the two additional TCP/IP stacks, we allocate two minidisks owned by the TCPMAINT virtual machine for the TCPIP111 and TCPIP112 configuration files. In Example 13, we show the additional MDISK statements allocations required for the TCPMAINT virtual machine.

*Example 13 Additional TCPMAINT minidisk definitions*

---

```
MDISK 0298 3390 0588 0009 LNXU1R MR RTCPMAIN WTCPMAIN MTCPMAIN
MDISK 0398 3390 0597 0009 LNXU1R MR RTCPMAIN WTCPMAIN MTCPMAIN
```

---

The 298 minidisk will be used for the TCPIP111 configuration files, and the 398 minidisk is for TCPIP112 configuration files.

## TCP/IP stack definitions

To simplify TCPIP111 and TCPIP112 stack configuration, we use the existing TCP/IP configuration files as templates. Following are the steps needed to begin configuration of the stacks:

1. Add the additional minidisk allocations (shown in Example 13) to the TCPMAINT user directory.

**Tip:** Use DirMaint to simplify the process of modifying the user directory.

2. Logon as user TCPMAINT (picking up the new minidisk definitions).
3. CMS-format the 298 and 398 minidisks.
4. Copy the existing TCPMAINT 198 minidisk contents to both the 298 and 398 minidisks (these serve as the templates for the TCPIP111 and TCPIP112 stacks).

Use this command sequence:

```
ACCESS 198 D
ACCESS 298 E
ACCESS 398 F
COPYFILE * * D = = E (OLDDATE
COPYFILE * * D = = F (OLDDATE
```

The TCP/IP stacks are now ready to be customized. Two files for each stack require modification:

- ▶ The PROFILE for the stack
- ▶ The DTCPARMS file for the stack

We first outline the changes required, and then discuss the steps needed to implement the modifications in “Obtaining and installing the sample code” on page 19.

## Customizing TCPIP111

To customize the TCPIP111 stack, we first copy the existing PROFILE TCPIP configuration to the TCPIP111 configuration minidisk (virtual device 298), and then modify the network configuration to match the topology shown in Figure 1 on page 14. In Example 14, we show the relevant modifications required.

### *Example 14 The PROFILE TCPIP111 definition*

---

```
DEVICE QD00 OSD 700 PORTNAME DTCP
LINK DEM00 QDIOETHERNET QD00

DEVICE QD02 OSD 704 PORTNAME DPICK
LINK DEM02 QDIOETHERNET QD02

HOME

10.0.10.2 DEM00
10.0.11.1 DEM02

; (End of HOME address information)

GATEWAY
10 = DEM00 1500 0.255.255.0 0.0.10.0
10 = DEM02 1500 0.255.255.0 0.0.11.0
DEFAULTNET 10.0.10.1 DEM00 1500 0
;
; (End of GATEWAY Static Routing information)

START QD00
START QD02
```

---



We also need to create a DTCPARMS file for the stack. As with the TCPIP111 PROFILE, we begin with a copy of the TCPIP DTCPARMS and add modifications specific to this environment. The relevant additions to the DTCPARMS file are shown in Example 15.

*Example 15 The TCPIP111 DTCPARMS definition*

---

```
:nick.TCPIP111 :type.SERVER :class.STACK
                :exit.TCPIP111
```

---

Finally, we need to add the TCPIP111 EXEC exit referred to in the TCPIP111 DTCPARMS file. This exit will define two simulated NICs and couple those NICs to their respective VM Guest LANs. In Example 16, we show the exit definition.

*Example 16 The TCPIP111 EXEC exit*

---

```
/* */
'PIPE CP DEFINE NIC 700 QDIO'
'PIPE CP COUPLE 700 SYSTEM DEMOTCP'
'PIPE CP DEFINE NIC 704 QDIO'
'PIPE CP COUPLE 704 SYSTEM DEMOCONT'
```

---

## **Customizing TCPIP112**

We need to customize the TCPIP112 stack with changes that are similar to those done for the TCPIP111 stack.

In Example 17 on page 18, we show the relevant modifications to the PROFILE TCPIP112 file.

*Example 17 The PROFILE TCPIP112 definition*

---

```
DEVICE QD00 OSD 700 PORTNAME DTCP
LINK DEM00 QDIOETHERNET QD00

DEVICE QD02 OSD 704 PORTNAME DPICK
LINK DEM02 QDIOETHERNET QD02

HOME

10.0.10.3 DEM00
10.0.12.1 DEM02

; (End of HOME address information)

GATEWAY
10          =          DEM00    1500    0.255.255.0 0.0.10.0
10          =          DEM02    1500    0.255.255.0 0.0.12.0
DEFAULTNET 10.0.10.1  DEM00    1500    0
;
; (End of GATEWAY Static Routing information)

START QD00
START QD02
```

---

In Example 18, we show the relevant modifications to the TCPIP112 DTCPARMS file.

*Example 18 The TCPIP112 DTCPARMS definition*

---

```
:nick.TCPIP112 :type.SERVER :class.STACK
               :exit.TCPIP112
```

---

In Example 19, we show the relevant modifications to the TCPIP112 EXEC exit.

*Example 19 The TCPIP112 EXEC exit*

---

```
/* */
'PIPE CP DEFINE NIC 700 QDIO'
'PIPE CP COUPLE 700 SYSTEM DEMOTCP'
'PIPE CP DEFINE NIC 704 QDIO'
'PIPE CP COUPLE 704 SYSTEM DEMOPICK'
```

---

## Customizing TCP/IP

Changes to a running TCP/IP stack should only be made using OBEYFILES and the **OBEY** command. To ensure changes to the stack persist across IPLs and restarts of the stack, the changes should also be added to the stack's PROFILE.

In Example 20, we show the changes required to the TCPIP user to reflect the network topology expressed in the TCPIP OBEYFILE. Similar modifications should be added to the TCPIP PROFILE if the changes are to be persistent.

*Example 20 The LXDEMO TCPIP OBEYFILE*

---

```
DEVICE QD00 OSD 700 PORTNAME DTCP
LINK DEM00 QDIOETHERNET QD00

HOME
  9.12.6.66 OSA2324L
  10.0.10.1 DEM00

GATEWAY
  9          =          OSA2324L 1492      0.255.255.0 0.12.6.0
  10         =          DEM00    1500      0.255.255.0 0.0.10.0
  10         10.0.10.2  DEM00    1500      0.255.255.0 0.0.11.0
  10         10.0.10.3  DEM00    1500      0.255.255.0 0.0.12.0
  10         9.12.6.97  OSA2324L 1492      0.255.255.0 0.0.1.0
  10         9.12.6.97  OSA2324L 1492      0.255.255.0 0.0.2.0
  10         9.12.6.97  OSA2324L 1492      0.255.255.0 0.0.3.0
  10         9.12.6.97  OSA2324L 1492      0.255.255.0 0.0.4.0
DEFAULTNET  9.12.6.75  OSA2324L 1500      0

START QD00
```

---

The LXDEMO TCPIP file should be stored on the TCPMAINT 592 minidisk.

**Tip:** For convenience, sample copies of these files are supplied in the additional materials available with this Redpaper; to obtain them, see “Additional Material” on page 42.

## Obtaining and installing the sample code

The sample configuration files outlined in this redpaper, and the scripts required to operate the cloning demonstration, are available online:

<ftp://www.redbooks.ibm.com/redpapers/REDP0301>

For instructions on how to obtain the sample code, see “Additional Material” on page 42.

Download the `lxopr191.vmarc` file to your workstation using binary mode. The material is packaged in VMARC format (to install VMARC, see “Install VMARC” on page 20).

We outline the steps needed to install the demonstration in the following sections. The package will be installed by the LXOPR VM user (as discussed in “The LXOPR virtual machine” on page 9).

## Enable DirMaint

You must first ensure that the DirMaint facility is enabled (see “DirMaint considerations” on page 4 for details).

## Prepare the LXOPR virtual machine

In order to use the LXOPR VM user to install the demonstration, LXOPR must first be defined to the VM user directory and its VM minidisks must be prepared, as follows:

1. Logon as MAINT (or any authorized DirMaint administrator).
2. Create a LXOPR DIRECT profile as shown in Example 6 on page 9, and issue the command:

```
DIRM ADD LXOPR
```

3. Next, logon as LXOPR.
4. CMS FORMAT the 191, 7191, and 8191 minidisks.
5. Access the 191 minidisk as filemode A, 8191 as filemode F, and 7191 as filemode G:

```
ACCESS 191 A  
ACCESS 8191 F  
ACCESS 7191 G
```

6. Using ftp, copy the lxopr191.vmarc file from your workstation to the LXOPR 191 disk (accessed as the CMS A-disk).

**Important:** Be sure to specify the `locsite fix 80 ftp` option before performing the transfer.

The VMARC package is now ready to be installed.

## Install VMARC

We need the VMARC package in order to install the demonstration. VMARC may be obtained from the z/VM Download Library Web site:

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

Follow the installation instructions.

## Install the demonstration code

The lxopr191.vmarc package is now ready to be installed on the LXOPR 191 minidisk.

1. To extract the code:

```
VMARC UNPACK LXOPR VMARC A = = A
```

2. Copy the DirMaint DIRMSAPI EXEC file to the LXOPR 191 disk:

```
LINK DIRMAINT 11F 11F RR  
ACCESS 11F T  
COPYFILE DIRMSAPI EXEC T = = A (OLDDATE
```

3. Set up the A-disks for the cloned Linux images:

```
COPYFILE PROF7191 EXEC A PROFILE EXEC G (OLDDATE  
COPYFILE MKSWAP EXEC A MKSWAP EXEC G (OLDDATE  
COPYFILE PROF8191 EXEC A PROFILE EXEC F (OLDDATE
```

The 7191 minidisk is now ready to act as the 191 A-disk for Linux clones using a VDISK swap device, and the 8191 will be the 191 A-disk for Linux clones using a swap device.

## Add the LXDFLT profile to DirMaint

We need to add the common user profile definition to DirMaint before creating users that rely on that profile. The LXDFLT PROFILE (shown in Example 1 on page 5) can be found on the LXOPR 191 minidisk. To add it, an authorized DirMaint user must link to this disk and add the definition by using these commands:

```
LINK LXOPR 191 491 RR  
DIRM ADD LXDFLT
```

## Prepare the LXCNTL virtual machine

The secondary console virtual machine requires some preparation, as follows:

1. Logon as MAINT (or any authorized DirMaint administrator) and create a LXCNTL PROFILE as shown in Example 5 on page 9. Create the LXCNTL virtual machine:

```
DIRM ADD LXCNTL
```

2. Logon as LXCNTL.

3. Format the 191 minidisk.
4. Create a PROFILE EXEC on the 191 A-disk (shown in Example 21).
5. Logoff LXCNTL.

*Example 21 The LXCNTL user PROFILE EXEC*

---

```
/* LXCNTL's PROFILE EXEC */
EXEC VMLINK TCPMAINT 592 <* R>
CP SET PF12 RETRIEVE
CP TERMINAL HOLD OFF
CP TERMINAL MORE 0 0
CP SET RUN ON
```

---

## Prepare the LXADMIN virtual machine

The LXADMIN virtual machine controls on-demand cloning. Portions of the demonstration specific to on-demand cloning are packaged in the lxadm191.vmarc package, which is available in the lxopr191.vmarc package itself. In addition, we need to install the Webshare Web server on the virtual machine, as follows:

1. Logon as MAINT (or any authorized DirMaint administrator) and create a LXADMIN PROFILE as shown in Example 7 on page 10. Create the LXADMIN virtual machine:

```
DIRM ADD LXADMIN
```

2. Logon as LXADMIN.
3. Format the 191 minidisk.
4. Install the lxadm191.vmarc package on the A-disk:

```
LINK LXOPR 191 291 RR
ACCESS 291 B
COPYFILE LXADM191 VMARC B = = A (OLDDATE)
VMARC UNPACK LXADM191 A = = A
```

5. Copy the DirMaint DIRMSAPI EXEC file to the LXADMIN 191 disk:

```
LINK DIRMAINT 11F 11F RR
ACCESS 11F T
COPYFILE DIRMSAPI EXEC T = = A (OLDDATE)
```

## Install Webshare

We use the Webshare Web server as the on-demand cloning front end. You can obtain Webshare from this site:

<http://www.neonsys.com/solutions/shadow/webshare.asp>

To install it, do the following:

1. Copy the cmshtpd.vmarc file to your workstation (be sure to use binary mode).
2. Using ftp, copy cmshtpd.vmarc from your workstation to the LXADMIN 191 minidisk.

**Important:** Be sure to specify the `locsite fix 80` ftp option before performing the transfer.

3. Unpack the package using the following command, and follow the installation instructions provided with the package.

```
VMARC UNPACK CMSHTPD VMARC A = = A
```

4. Edit the HTTPD CONFIG A file and change the line:

```
CGIUSERS=
```

to:

```
CGIUSERS=LXADMIN
```

5. Edit the WEBSHARE FILELIST A file and add the following lines to the end:

```
LXADMIN HTML A
```

```
ALLIMG HTML A
```

6. Edit the HTBIN FILELIST A file and add the following line to the end:

```
LXADM *CGI A
```

7. Edit the HTLIB FILELIST A file and add the following lines to the end:

```
TUX_SITT JPG A
```

```
TUX_DANC JPG A
```

The Webshare server is now configured for on-demand cloning.

## Implement the TCP/IP configuration changes

We are now ready to implement the network topology outlined in “TCP/IP stack configuration” on page 13. The steps to take are as follows:

1. Modify the TCPMAINT user to include the additional minidisk allocations, as outlined in “TCPMAINT virtual machine modifications” on page 15.
2. Logon as MAINT (or any authorized DirMaint administrator).
3. Create the TCPIP111 and TCPIP112 virtual machines by adding their respective user profiles to the VM directory.

Using the profiles shown in Example 11 and Example 12 on page 13, simply issue these commands:

```
DIRM ADD TCPIP111
```

```
DIRM ADD TCPIP112
```

4. Copy the contents of the TCPIP 191 minidisk to the 191 minidisks of the TCPIP111 and TCPIP112 virtual machines:
  - a. Logon to TCPIP111.
  - b. Copy the TCPIP 191 minidisk:

```
LINK TCPIP 191 291 RR RTCPIP  
ACCESS 291 T  
COPYFILE * * T = = A (OLDDATE
```
  - c. Logoff TCPIP111.
  - a. Logon to TCPIP112.
  - b. Copy the TCPIP 191 minidisk:

```
LINK TCPIP 191 291 RR RTCPIP  
ACCESS 291 T  
COPYFILE * * T = = A (OLDDATE
```
  - c. Logoff TCPIP112.
5. Create a base TCP/IP stack using the procedure outlined in “TCP/IP stack definitions” on page 15.
6. Customize the TCPIP111 and TCPIP112 stacks as outlined in “Customizing TCPIP111” on page 16 and “Customizing TCPIP112” on page 17.
7. Activate the TCP/IP changes. Assuming the OBEYFILE name is LXDEMO TCPIP, perform the following sequence:
  - a. Logon to TCPMAINT.
  - b. Issue the command:

```
OBEYFILE LXDEMO TCPIP R ALL TCPIP
```
  - c. Logoff TCPMAINT.

## Prepare DirMaint for cloning

Some customization of DirMaint is required, as described in the following sections.



## Authorize DirMaint administrators

The LXOPR and LXADMIN users need to be authorized as DirMaint administrators, as follows:

1. Logon as MAINT (or any authorized DirMaint administrator).
2. Receive a copy of the AUTHFOR CONTROL file from DirMaint:

```
DIRM SEND AUTHFOR CONTROL
```

The file will be sent to your reader. Receive it to the A-disk.

3. Edit the file, adding the lines indicated in Example 22.
4. Send the revised file back to DirMaint:

```
DIRM FILE AUTHFOR CONTROL
```

5. Activate the changes:

```
DIRM RLDC
```

*Example 22 The AUTHFOR CONTROL file additions*

---

ALL	LXOPR	*	150A	ABCDEFGHIJKLMN	OPQRSTUVWXYZ	1234567890
ALL	LXOPR	*	140A	ABCDEFGHIJKLMN	OPQRSTUVWXYZ	1234567890
ALL	LXADMIN	*	150A	ABCDEFGHIJKLMN	OPQRSTUVWXYZ	1234567890
ALL	LXADMIN	*	140A	ABCDEFGHIJKLMN	OPQRSTUVWXYZ	1234567890

---

## Add a DASD group to the DirMaint virtual machine

We next need to add a DASD group from which Linux clones will be allocated minidisk space. From an authorized DirMaint administrator, do the following:

1. Receive a copy of the EXTENT CONTROL file from DirMaint:

```
DIRM SEND EXTENT CONTROL
```

The file will be sent to your reader. Receive it to the A-disk.

2. Edit the file adding the lines indicated in Example 23 to the :REGIONS and :GROUPS sections.

**Note:** The additions to be made are highlighted in the example.

3. Send the revised file back to DirMaint:

```
DIRM FILE EXTENT CONTROL
```

4. Activate the changes:

```
DIRM RLDE
```

*Example 23 The EXTENT CONTROL file additions*

---

```
:REGIONS.  
*RegionId      VolSer      RegStart    RegEnd      Type  
430RES        430RES        001         3338        3390-03  
430W01        430W01        001         3338        3390-03  
LNXU1R        LNXU1R        001         3338        3390-03  
LX651A        LX651A        001         3338        3390-03  
LX660E        LX660E        001         3338        3390-03  
LX660F        LX660F        001         3338        3390-03  
:END.  
:GROUPS.  
ANY           LNXU1R  
LXDEMO        LX651A LX660E LX660F  
:END.
```

---

### **Add the Linux clone prototypes to DirMaint**

Now that LXOPR is an authorized DirMaint administrator, it can add the Linux clone prototypes found on its 191 minidisk. As LXOPR, issue these commands:

```
DIRM FILE LXSUSEI PROTODIR  
DIRM FILE LXSUSEIV PROTODIR  
DIRM FILE LXSUSERW PROTODIR
```

### **Install the LXMNT Linux guest**

We now need to install a Linux guest on the LXMNT virtual machine.

**Important:** Make certain the LXMSTR user is logged off before installing the LXMNT guest.

1. Logon as MAINT (or any authorized DirMaint administrator) and create a LXMNT PROFILE as shown in Example 2 on page 6. To create the LXMNT virtual machine:

```
DIRM ADD LXMNT
```

2. Logon to the LXMNT user.
3. Format the 191, 151, 153, 154, 155, and 5251 minidisks.
4. Install the SuSE SLES-7 Professional distribution. The important configuration parameters to use during installation are:
  - a. Select option **3 OSA-Gigabit Ethernet of OSA-Express Fast Ethernet** as the network device.
  - b. For ethernet device addresses, use 0x0700,0x0701,0x0702.

- c. Use 10.0.11.253 as the host IP address.
  - d. Use 10.0.11.1 as the gateway IP address.
  - e. Specify the 151 minidisk for the / filesystem.
  - f. Specify the 153 minidisk for the /usr filesystem.
  - g. Specify the 154 minidisk for the /usr/src filesystem.
  - h. Specify the 155 minidisk for the swap device.
  - i. Select **SuSE Network** from the load configuration menu.
  - j. Set the root password to linux4vm.
5. Once Linux is installed, shut down the Linux guest and reboot it:

**IPL 151**

Now we are ready to modify the /etc/fstab file in the Linux guest. We will change the /usr and /usr/src mounts (on the 153 and 154 minidisks respectively) to be read-only. See Example 2 on page 6 for the LXMMNT user minidisk definitions. These filesystems will form the read-only /usr and /usr/src mounts used by the Linux clones.

### Change the /usr and /usr/src mounts

In Example 24 we show the modifications made to the /etc/fstab file. In this case we have changed the attributes of the mount from defaults to ro (read-only).

*Example 24 The LXMMNT Linux guest /etc/fstab file*

---

/dev/dasda1	swap	swap	defaults	0	0
/dev/dasdb1	/	ext2	defaults	1	1
/dev/dasdc1	/usr	ext2	ro	1	2
/dev/dasdd1	/usr/src	ext2	ro	1	2
proc	/proc	proc	defaults	0	0
# End of YaST-generated fstab lines					

---

After modifying the /etc/fstab file, shut down the Linux guest in preparation for the next step.

### Copy the root filesystem to the 5251 minidisk

Next we copy the root filesystem residing on the 151 minidisk to the 5251 minidisk. This new root filesystem will be used by Linux clones that have a read-only /usr mount with a read-write /usr/src mount. The steps to take are:

1. Use the CP DDR utility to copy the 151 minidisk to the 5251 minidisk.
2. Boot the Linux guest using the new root filesystem:

**IPL 5251**

3. Modify the `/etc/fstab` file to change the attributes of the `/usr/src` mount back to the original defaults as shown in Example 25.
4. Shut down the Linux guest and logoff the LXMNT virtual machine.

*Example 25 The LXMNT Linux guest /etc/fstab file for the 5251 minidisk*

---

<code>/dev/dasda1</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>/dev/dasdb1</code>	<code>/</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>1</code>
<code>/dev/dasdc1</code>	<code>/usr</code>	<code>ext2</code>	<code>ro</code>	<code>1</code>	<code>2</code>
<code>/dev/dasdd1</code>	<code>/usr/src</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>2</code>
<code>proc</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code># End of YaST-generated fstab lines</code>					

---

We are now ready to install and configure the LXMSTR Linux guest.

## Install the LXMSTR Linux guest

We now need to install a Linux guest on the LXMSTR virtual machine.

**Important:** Make certain the LXMNT user is logged off before installing the LXMSTR guest.

1. Logon as MAINT (or any authorized DirMaint administrator) and create a LXMSTR PROFILE as shown in Example 3 on page 7. To create the LXMSTR virtual machine:
 

```
DIRM ADD LXMSTR
```
2. Logon to the LXMSTR user.
3. Format the 191, 151, 153, 154, and 155 minidisks.
4. Install the SuSE SLES-7 Professional distribution. The important configuration parameters to use during installation are:
  - a. Select option **3 OSA-Gigabit Ethernet of OSA-Express Fast Ethernet** as the network device.
  - b. For ethernet device addresses, use `0x0700,0x0701,0x0702`.
  - c. Use `10.0.11.251` as the host IP address.
  - d. Use `10.0.11.1` as the gateway IP address.
  - e. Specify the 151 minidisk for the `/` filesystem.
  - f. Specify the 153 minidisk for the `/usr` filesystem.
  - g. Specify the 154 minidisk for the `/usr/src` filesystem.
  - h. Specify the 155 minidisk for the swap device.

- i. Select **SuSE Network** from the load configuration menu.
  - j. Set the root password to `linux4vm`.
5. Once Linux is installed, shut down the Linux guest and reboot it:

**IPL 151**

At this point, we are ready to prepare the Linux guest to mount the 1251 and 5251 minidisks that will serve as read-only root filesystem for the Linux clones (see Example 3 on page 7 for the LXMSTR user minidisk definitions).

## Mounting the read-only root filesystems for Linux clones

We need to modify the `/etc/zipl.conf` file (to define the 1251 and 5251 minidisks to the Linux guest), and the `/etc/fstab` file in the LXMSTR Linux guest. A telnet session should be used to connect to the Linux guest when making these changes.

First, we change the `/etc/zipl.conf` file. In Example 26, we show the file contents with the additional entries highlighted.

*Example 26 The LXMSTR Linux guest /etc/zipl.conf file*

---

```
[defaultboot]
default=ipl

[ipl]
target=/boot/zipl
image=/boot/kernel/image
#ramdisk=/boot/initrd
parameters="dasd=0155,0151,0153,0154,1251,5251 root=/dev/dasdb1 noinitrd"
[dumptape]
target=/boot
dumpto=/dev/rtibm0
```

---

Next, in Example 27 we show the additional filesystem definitions in `/etc/fstab`.

*Example 27 The LXMSTR Linux guest /etc/fstab file*

---

<code>/dev/dasda1</code>	<code>swap</code>	<code>swap</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code>/dev/dasdb1</code>	<code>/</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>1</code>
<code>/dev/dasdc1</code>	<code>/usr</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>2</code>
<code>/dev/dasdd1</code>	<code>/usr/src</code>	<code>ext2</code>	<code>defaults</code>	<code>1</code>	<code>2</code>
<b><code>/dev/dasde1</code></b>	<b><code>/suser</code></b>	<b><code>ext2</code></b>	<b><code>defaults</code></b>	<b><code>1</code></b>	<b><code>2</code></b>
<b><code>/dev/dasdf1</code></b>	<b><code>/susew</code></b>	<b><code>ext2</code></b>	<b><code>defaults</code></b>	<b><code>1</code></b>	<b><code>2</code></b>
<code>proc</code>	<code>/proc</code>	<code>proc</code>	<code>defaults</code>	<code>0</code>	<code>0</code>
<code># End of YaST-generated fstab lines</code>					

---

**Important:** Remember to create the /suser and /susew directories as stub mountpoints.

Once these changes have been made, issue the **zipl** command (to make the /etc/zipl.conf changes take effect). Then shut down and reboot the Linux guest to ensure the new minidisks are seen by Linux.

**Tip:** At this point, you should be able to see the two Linux root filesystems created during the “Install the LXDMNT Linux guest” on page 26 step.

## Install the LXDISP Linux guest

We now install a Linux guest on the LXDISP virtual machine. This guest will run an Apache server to display the number of Linux clones created during the continuous cloning exercise.

1. Logon as MAINT (or any authorized DirMaint administrator) and create a LXDISP PROFILE as shown in Example 5 on page 9. To create the LXDISP virtual machine, issue:

**DIRM ADD LXDISP**

2. Logon to the LXDISP user.
3. Format the 191, 151, and 155 minidisks.
4. Install the SuSE SLES-7 Professional distribution. The important configuration parameters to use during installation are:
  - a. Select option **3 OSA-Gigabit Ethernet of OSA-Express Fast Ethernet** as the network device.
  - b. For ethernet device addresses, use 0x0700, 0x0701, and 0x0702.
  - c. Use 10.0.11.252 as the host IP address.
  - d. Use 10.0.11.1 as the gateway IP address.
  - e. Specify the 151 minidisk for the / filesystem.
  - f. Specify the 155 minidisk for the swap device.
  - g. Select **SuSE Network** from the load configuration menu.
  - h. Set the root password to `linux4vm`.
5. Once Linux is installed, shut down the Linux guest and reboot it:

**IPL 151**

## Configure the Apache server

We now continue by configuring Apache to use content provided as part of the LXOPR191 VMARC package.

1. Using ftp, copy the lxdisp.tar file found on the LXOPR 191 minidisk to the LXDISP Linux guest. Expand the tar file to the /usr/local/httpd/htdocs directory.
2. Customize the /etc/httpd/httpd.conf Apache configuration file for the LXDISP Linux guest.

**Tip:** To install the blinking banner found on the continuous cloning Web pages, you need to download the fader.js JavaScript file found at:

<http://www.webreference.com/dhtml/column27/allCode1.html>

Install this in the /usr/local/httpd/htdocs directory.

## Contents of the LXOPR191 VMARC package

The LXOPR191 VMARC package contains REXX execs and data files used by both the continuous cloning and the on-demand cloning processes.

## Scripts used for both continuous cloning and on-demand cloning

Following is a list of the files used for both continuous cloning and on-demand cloning, and explanations of their purposes.

### NEWSUSE EXEC

This creates a Linux clone from the LXSUSEI prototype. It is called by LXADM CGI and the MAKESUSE EXEC.

### NEWSUSEV EXEC

This creates a Linux clone from the LXSUSEI prototype. It is called by LXADM CGI and the MKSUSEV EXEC.

### SENDLX EXEC

This sends a command to a Linux image.

### RMLX EXEC

This deletes a Linux image. It is called by LXADM CGI and the CLEANUP EXEC.

## **PURGELX EXEC**

This forces a Linux image off the z/VM system and deletes the image.

## **DIRMSAPI EXEC**

This creates a Linux clone virtual machine using DirMaint.

## **IPADDR LIST**

This contains a set of variable names and values for several items used by the cloning processes. These items include:

- ▶ The value for the maximum number of Linux images to be cloned by the continuous cloning process.
- ▶ The Linux image number with which to begin continuous cloning.
- ▶ The IP addresses needed by the various parts of the cloning process.
- ▶ The latter portion of the fully qualified domain name of the cloned Linux images.
- ▶ The names of the z/VM Guest LANS.
- ▶ The names and IP addresses of our additional TCP/IP stacks.

If you choose to change our default values, simply update the IPADDR LIST on the LXOPR 191 minidisk.

## **IPADDRES FILEETH0**

This is a map of IP addresses and Linux image user IDs used by NEWSUSE EXEC, NEWSUSEV EXEC, and SUSELX EXEC.

## **PT2PT FILEETH0**

The gateway IP address for each Linux image. In this case, the gateway IP address is 10.0.11.251 for Linux images LX1 to image LX250, and 10.0.12.251 for Linux images LX251 to LX500.

## **MODULES SUSEETH0**

This is the input file used by NEWSUSE EXEC, NEWSUSEV EXEC, and SUSELX EXEC to create the `/etc/modules.conf` file for a SuSE Linux image.

## **RCONFIG SUSEETH0**

This is the input file used by NEWSUSE EXEC, NEWSUSEV EXEC, and SUSELX EXEC to create the `/etc/rc.config` file for a SuSE Linux image.



## **HOSTS SUSEETH0**

This is the input file used by NEWSUSE EXEC, NEWSUSEV EXEC and SUSELX EXEC to create the /etc/hosts file for a SuSE Linux image.

## **HOSTNAME SUSEETH0**

This is the input file used by NEWSUSE EXEC, NEWSUSEV EXEC and SUSELX EXEC to create the /etc/HOSTNAME file for a SuSE Linux image.

## **ROUTE SUSEETH0**

This is the input file used by NEWSUSE EXEC, NEWSUSEV EXEC and SUSELX EXEC to create the /etc/route.conf file for a SuSE Linux image.

## **DDRCMDS FILE**

This is the DDR command file for creating images from the LXSUSEI prototype. Used by NEWSUSE EXEC.

## **DDRCMDV FILE**

This is the DDR command file for creating images from the LXSUSEIV prototype. It is used by the NEWSUSEV EXEC.

## **NETRC DATA**

The cloning execs issue FTP commands to move the network configuration files to a new Linux image. When this file exists, the FTP performs an automatic login of the user specified in this file. For this demonstration package, this file contains a single line:

```
login root password linux4vm
```

## **Scripts used solely for continuous cloning**

The following content is used only for the continuous cloning process.

### **MAKESUSE EXEC**

This is the main exec to start the demonstration and create 249 Linux images with DASD swap space (the default maximum of 249 can be found on the MAX\_IMAGE statement in the IPADDR LIST). The START\_IMAGE statement in IPADDR LIST defines which Linux image to create first (the default is the LX2 user).

### **MKSUSEV EXEC**

This is the main exec to start the demonstration and create 249 Linux images with VDISK swap space.

## **CLEANUP EXEC**

This deletes Linux images and resets the counter to 0. Issue `cleanup 5` to delete the LX1 to LX5 Linux images.

## **STLXTCP EXEC**

This is used to start the TCPIP111 and TCPIP112 stacks.

## **STLXVM EXEC**

This is used to start the LXDISP and LXMSTR Linux images.

## **LXCMD EXEC**

This is used to send a command to a Linux image.

## **SHUTLX EXEC**

This is used to shut down the LXDISP and LXMSTR Linux images.

## **FILE TXT**

Images creation time and number status file.

## **CLONING IMAGE**

This is a text file that indicates the type of images being cloned. LXADMIN checks this file before cloning any image.

The check ensures continuous cloning and on-demand cloning do not write to the same master Linux image's / filesystem at the same time.

## **Scripts used solely for on-demand cloning**

The LXADM191 VMARC package contains REXX execs and data files used by the on-demand cloning process. Following is a list of the files and explanations of their purposes.

### **SUSELX EXEC**

This creates a SuSE Linux image LXSUSERW prototype. SUSELX EXEC is called by LXADM CGI.

### **LXADM CGI**

This is the main REXX CGI exec to create and delete Linux images. It is called by LXADMIN HTML when the Web server receives a request to clone a Linux image.

## **DDRCMDSW FILE**

This is the DDR command file for creating images for the LXSUSERW prototype. It is used by SUSELX EXEC.

## **CUSTLIST TXT**

This is a text file that stores customer information. It is modified by LXADM CGI.

## **ADMLIST TXT**

This is a text file that stores an administrative user ID and password. It contains a single line:

```
lxadm linux4vm
```

## **LXADMIN HTML**

This the main HTML for the on-demand cloning Web interface.

## **ALLIMG HTML**

This contains the names of all images that were created. It is modified by the LXADMIN CGI script. The first three lines of this file must be:

```
<html>  
<font size=+2>  
<h1> Images are: <p> </h1>
```

## **CIMAGE HTML**

This contains a list of the Linux images on-demand cloning is allowed to create. The default is 10 Linux images (LX491 to LX500). You can specify any set of Linux image names you want to create here. The CIMAGE HTML file is modified by LXADMIN CGI to remove an image name when the image is created. The last line of this file must be:

```
**end**
```

# **Running the on-demand cloning process**

In this section we describe how to run the on-demand cloning portion of the demonstration.

## **Starting the process**

1. Logon as LXCNTL.
2. Logon as LXOPR.

3. From the LXOPR user ID, start the two additional TCP/IPs:

**STLXTCP**

**Important:** Stop if **STLXTCP** fails, or the TCPIP111 and TCPIP112 stacks do not start. Determine and correct the problem before continuing.

4. From the LXOPR user ID, start the LXMSTR and LXDISP master Linux images:

**STLXVM**

**Important:** Stop if **STLXVM** fails, or the LXMSTR and LXDISP Linux images do not start. Determine and correct the problem before continuing.

5. Logon as LXADMIN and start the WEBSHARE Web server:

**WEBSHARE**

6. Access the on-demand cloning administration page at:

`http://vmaddr/~lxadmin/lxadmin.html`

**Note:** Substitute the IP name or address of the LXADMIN virtual machine for `vmaddr` in the above URL.

An example of what is displayed by your browser can be seen at “Linux image administration Web page” on page 39.

Supply the appropriate fields in the HTML form (use `lxadm` as the Administrative Userid and `linux4vm` as the Password). Select the desired image and submit the request.

This will create one Linux image. When the Linux image is created, a dancing TUX will be displayed, along with the image's name and IP address.

You can click the sitting TUX on the first Web page to see how many images were created.

## Deleting images created by on-demand cloning

On the on-demand cloning administrator Web page, fill in the Image Name, Administrative user ID, and Password. Click **Delete an Image** to delete a Linux image.

You can click the sitting TUX on the administrator Web to find out how many Linux images were created and what their z/VM user IDs are.

## Running the continuous cloning process

There are several execs involved with the continuous cloning portion of this demonstration package. There are also several data files used by the execs. All of these execs and data files are packaged together in the LXOPR191 VMARC file. See “The LXOPR virtual machine” on page 9 for information on extracting these files. (The assumption here is that the execs and data files all reside on the LXOPR 191 minidisk.)

The continuous cloning process is started directly from the LXOPR virtual machine by running either the MAKESUSE EXEC or the MKSUSEV EXEC. These scripts operate as follows:

1. Issue DirMaint commands to create the clone virtual machine.
2. Modify the CMS copies of the Linux network files
3. Use FTP to put these modified files in one of the Linux master images.
4. Make a DDR copy of the Linux master image minidisks on the new Linux images’s minidisks.
5. Boot the new Linux image.
6. Ping the new Linux image. Cloning is complete when the ping is answered.
7. Report the number Linux images created.
8. Repeat these steps until all requested Linux images are cloned.

## Starting the continuous cloning process

1. Logon as LXCNTL.
2. Logon as LXOPR.
3. From the LXOPR user, start the TCPIP111 and TCPIP112 TCP/IP stacks:

```
STLXTCP
```

**Important:** Stop if **STLXTCP** fails, or the TCPIP111 and TCPIP112 stacks do not start. Determine and correct the problem before continuing.

4. From the LXOPR user, start the LXMSTR and LXDISP master Linux images:

```
STLXVM
```

**Important:** Stop if **STLXVM** fails, or the LXMSTR and LXDISP Linux images do not start. Determine and correct the problem before continuing.

- Using the Web browser of your choice, you can access either the standard demonstration Web page:

```
http://vmaddr/tscdemo.html
```

Or you can access the “blinking banner” version (the fader.js JavaScript file is required for the blinking banner):

```
http://vmaddr/speedometer.html
```

**Note:** Substitute the IP name or address of the LXDISP virtual machine for vmaddr in the above URLs.

An example of the Web page displayed in your browser can be seen at, “Continuous cloning speedometer Web page” on page 40. When the Web page comes up, reload it one more time by clicking the reload button.

- To start the continuous cloning demonstration, from the LXOPR user, enter either:

```
MAKESUSE
```

or:

```
MKSUSEV
```

If the Web page is not updated as the new images are created, reload the page.

- Leaving the LXOPR and LXCNTL users logged on, you should observe the cloning process on LXOPR.

**Note:** All messages from the cloned Linux images will appear on the console of the LXCNTL user.

## Deleting images created by continuous cloning

There are two methods of deleting Linux images created by the continuous cloning process. You can delete all the Linux images created, or you can delete individual Linux images.

### Delete all images

Logon as LXOPR and enter the command:

```
cleanup nnn
```

where **nnn** is the number of images to be deleted.

**Note:** Before rerunning the demonstration, you must delete all the cloned Linux images.

If you stopped the MAKESUSE EXEC or MKSUSEV EXEC before they reached the MAX\_IMAGE value, you must add 1 to the number of images displayed on the speedometer Web page as the total number of images to delete. For example, if the speedometer Web page shows 250 images, then run the following command to delete all cloned images:

```
cleanup 251
```

### **Delete a single image**

From LXOPR, you can also use the PURGELX EXEC to delete one image at a time. Run the following command to delete the LX4 Linux image:

```
purge1x 1x4
```

## **The Web pages**

As a point of reference we are including shots of the two main Web pages used in this demonstration package.

### **Linux image administration Web page**

The on-demand cloning exercise begins when you submit your request from the Linux Image Administrator Web page. This page appears as shown in Figure 2 on page 40.

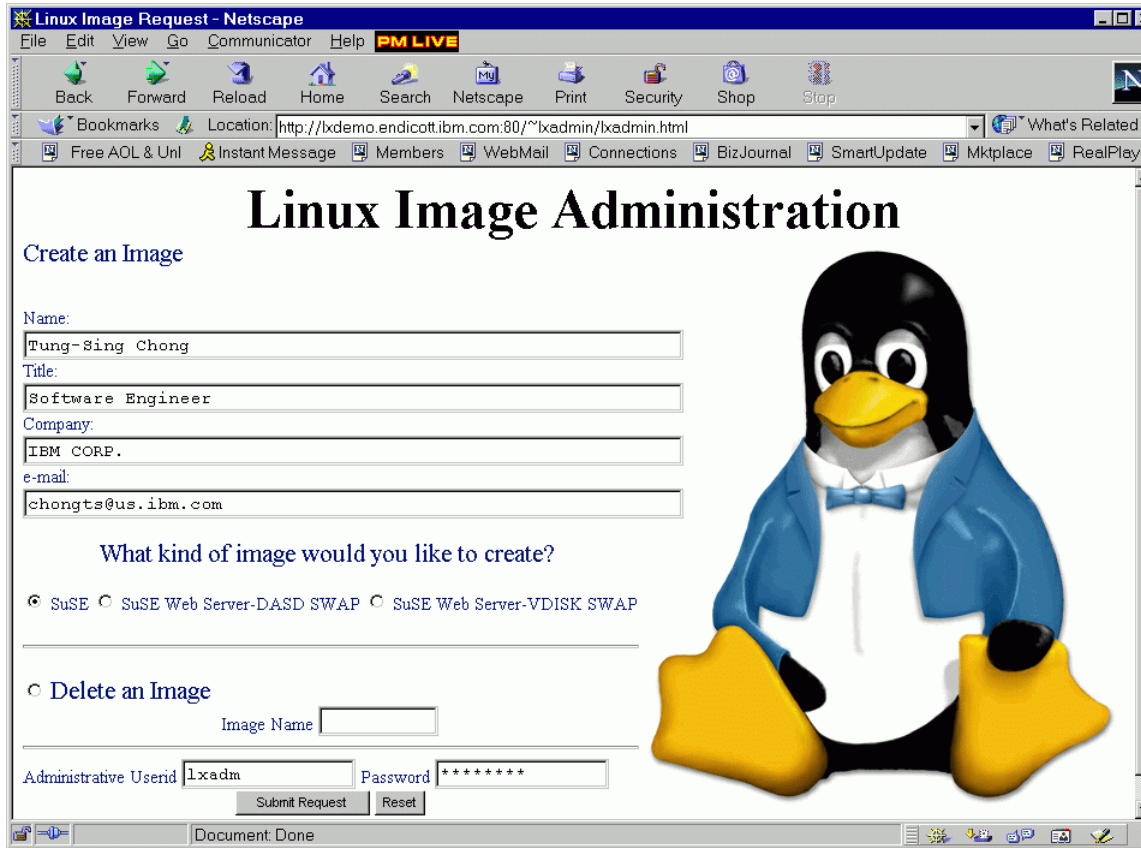


Figure 2 Linux image administration Web page

## Continuous cloning speedometer Web page

When the continuous cloning demonstration runs, LXDISP displays a “speedometer” Web page that shows its progress; refer to Figure 3 on page 41.



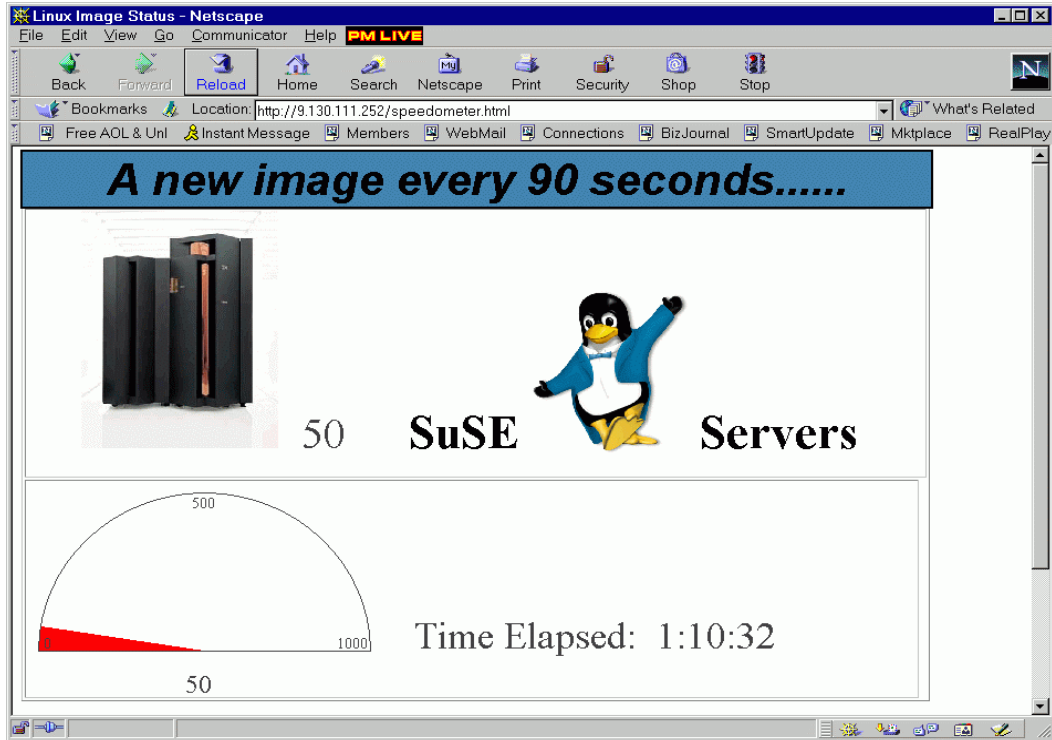


Figure 3 The continuous cloning process speedometer

## Additional Material

This Redpaper refers to additional material which downloaded as described below.

### Locating the Web material

The Web material associated with this Redpaper is available in softcopy on the Internet from IBM Redpaper Redbooks Web Server. Point your browser to

`ftp://www.redbooks.ibm.com/redbooks/REDP0301`

Alternately, you can go to the IBM Redbooks site at

`ibm.com/redbooks`

Select **Additional materials** and open the directory that corresponds to Redpaper form number, REDP0301.

### Using the Web material

The additional material that accompanies this Redpaper includes

- ▶ `1xopr191.vmarc` - the VMARC package containing the sample code.

### System requirements for downloading the Web Material

The following system configuration is recommended:

**Hard disk space:** 1 MEG minimum

**Operating System:** z/VM 4.3.0 or higher, a Linux distribution for IBM zSeries

### How to use the Web Material

Download the `1xopr191.vmarc` package and follow the directions outlined in “Obtaining and installing the sample code” on page 19.

# References

## z/VM Documentation

*z/VM V4R3.0 CP Planning and Administration, SC24-6043*

## Referenced Web sites

- ▶ Tung-Sing Chong's home page  
<http://www.vm.ibm.com/devpages/chongts/>
- ▶ The Webshare Web server home page  
<http://www.neonsys.com/solutions/shadow/webshare.asp>
- ▶ Dynamic Headline Fader, Version 3.0  
<http://www.webreference.com/dhtml/column27/allCode1.html>

## The team that wrote this paper

This Redpaper was produced by of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

**Gregory Geiselhart** is a project leader for Linux on zSeries at the International Technical Support Organization, Poughkeepsie Center.

**Tung-Sing Chong** is a Senior Software Engineer with IBM in Endicott, NY. He joined S/370 development as a VM system programmer in 1987 at the Glendale Programming Laboratory. He holds a masters degree in Computer Science from the State University of New York at Binghamton. Besides VM system programming, he has also been involved in AS/400 IO subsystem development. Currently, he is providing system support for Linux on zSeries as well as working on HMC development for eServer.

**Michael Donovan** is a Senior Software Engineer with IBM in Endicott, NY. He joined VM Development in 1980 at the Glendale Programming Laboratory. He holds a bachelors degree in Computer Science from the State University of New York at Oswego. He has been involved in almost all aspects of the VM product, including testing, packaging, design, development, and service. Most recently, he was a developer on OpenExtensions for z/VM, Java for z/VM, and Language Environment for VM. He has also been involved in the z/VM support of Linux for zSeries.



# Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.*

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on August 28, 2002.




Send us your comments in one of the following ways:

- ▶ Use the online **Contact us** review redbook form found at:  
[ibm.com/redbooks](http://ibm.com/redbooks)
- ▶ Send your comments in an Internet note to:  
[redbook@us.ibm.com](mailto:redbook@us.ibm.com)
- ▶ Mail your comments to:  
IBM Corporation, International Technical Support Organization  
Dept. HYJ Mail Station P099, 2455 South Road  
Poughkeepsie, NY 12601-5400 U.S.A.

## Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

DirMaint™	Redbooks(logo)™ 	z/Architecture™
DPI®	Redbooks™	z/VM™
IBM®	S/390®	zSeries™
MORE™	SP™	

The following terms are trademarks of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both:

Lotus®	Word Pro®
--------	-----------

The following terms are trademarks of other companies:

ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

C-bus is a trademark of Corollary, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET, SET Secure Electronic Transaction, and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, and service names may be trademarks or service marks of others.