S/390 Partners in Development:

EFS Systems on a Linux Base

ThinkPad/EFS systems
Netfinity/EFS systems

Installation, customization, operation

z/OS AD CD-ROM software

Bill Ogden

ibm.com/redbooks
International Technical Support Organization

S/390 Partners in Development:

EFS Systems on a Linux Base

July 2002
Take Note! Before using this information and the product it supports, be sure to read the general information in “Notices” on page vii.

First Edition (July 2002)

This edition applies to the status of ThinkPad/EFS and Netfinity/EFS systems as of July 2002, with the FLEX-ES releases current at this time.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2002.
Note to U.S Government Users - Documentation related to restricted rights - Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.
## Contents

**Notices**  ........................................................................................................... vii  
**Trademarks** ...................................................................................................... viii

**Preface** ............................................................................................................ ix  
**The author** ......................................................................................................... ix  
**Comments welcome** ......................................................................................... x

**Chapter 1. Basic concepts**  .............................................................................. 1  
1.1 Purpose of this redbook ................................................................................ 1  
1.2 Usage overview ............................................................................................. 2  
1.3 Terminology ................................................................................................... 4  
1.4 Basic EFS requirements ............................................................................... 4  
1.5 FLEX-ES ....................................................................................................... 5  
1.6 Linux ............................................................................................................... 5  
1.7 ThinkPad versus Netfinity ............................................................................ 6  
1.7.1 Hardware restrictions .............................................................................. 7  
1.8 ThinkPad hardware ....................................................................................... 7  
1.9 Netfinity hardware ....................................................................................... 8

**Chapter 2. Hardware and Linux installation** .................................................. 11  
2.1 FLEX-ES disk usage ..................................................................................... 11  
2.2 Network connections ................................................................................... 12  
2.3 ThinkPad installation .................................................................................... 13  
2.3.1 Disk planning .......................................................................................... 13  
2.3.2 Linux installation .................................................................................... 14  
2.3.3 Installation notes ..................................................................................... 18  
2.4 Netfinity installation .................................................................................... 20  
2.4.1 Download and install upgrades ............................................................... 21  
2.4.2 Linux installation .................................................................................... 23  
2.4.3 Multiprocessor servers .......................................................................... 27  
2.4.4 Minor problem ....................................................................................... 29

**Chapter 3. FLEX-ES installation** ................................................................. 31  
3.1 FLEX-ES overview ..................................................................................... 31  
3.2 FLEX-ES installation ................................................................................... 34  
3.2.1 X3270 .................................................................................................... 37  
3.2.2 Mount points for ThinkPad ................................................................. 37  
3.2.3 Mount points for Netfinity ................................................................. 37

**Chapter 4. S/390 Operating System** ............................................................. 39  
4.1 AD systems .................................................................................................. 39  
4.1.1 z/OS on CD-ROM .................................................................................. 40  
4.1.2 z/OS device configuration .................................................................... 41  
4.1.3 Future directions .................................................................................... 42  
4.1.4 An unzip program ................................................................................ 42  
4.2 Installation tasks ........................................................................................... 42  
4.2.1 Create a work volume .......................................................................... 44  
4.2.2 Loading the remaining volumes ............................................................ 44  
4.3 File ownership ............................................................................................... 45
## Chapter 5. FLEX-ES operation

5.0.1 Linux desktop windows ................................................. 48
5.1 FLEX-ES system and resource definitions .............................. 48
  5.1.1 Building a shell script .............................................. 50
  5.1.2 IPL z/OS ............................................................ 52
5.2 Operation and use .......................................................... 53
  5.2.1 User terminal connection .......................................... 53
  5.2.2 IODF requirements .................................................. 53
  5.2.3 System performance monitors ..................................... 54
  5.2.4 TCP/IP for z/OS .................................................... 54
5.3 Shutting down .................................................................. 55
5.4 More comments .................................................................. 55
5.5 Options ............................................................................ 56

## Chapter 6. Additional topics

6.1 Linux windows and FLEX-ES operation .................................. 57
  6.1.1 Change resource definitions ........................................ 58
6.2 Basic debugging .................................................................. 58
6.3 Operating Systems Messages console .................................... 60
6.4 Security ........................................................................... 60
6.5 Server memory ................................................................... 61
  6.5.1 The vmstat command .................................................. 61
  6.5.2 Importance of Linux swapping ...................................... 62
6.6 Ultrabay ........................................................................... 63
  6.6.1 Using a second Linux hard disk .................................... 63
  6.6.2 Disk layout (AD system with two hard disks) .................... 64
6.7 FLEX-ES FakeTape on z/OS ............................................... 65
6.8 Multi-system setup ............................................................ 67
6.9 x3270 client .................................................................... 67
6.10 Remote resources .............................................................. 69
6.11 Disk caches ..................................................................... 70
6.12 Tuning cachesize .............................................................. 72
6.13 Backup and restore considerations (ThinkPad) ...................... 72
  6.13.1 Using tar to back up S/390 volumes ............................... 73
  6.13.2 Using tar and ftp ...................................................... 74
  6.13.3 CD-RW drive .......................................................... 75
6.14 Backup and restore considerations (Netfinity) ...................... 75
6.15 S/390 identification ......................................................... 76
6.16 Display PSW and registers .................................................. 77
6.17 RAS discussion .................................................................. 77
6.18 Verify ckd disk ............................................................... 78
6.19 Installing FLEX-ES upgrades ............................................ 78
6.20 Remote operation ............................................................. 78
6.21 ThinkPad power control .................................................... 79
6.22 Using the 4mm tape drive .................................................. 79
6.23 Using an Overland T490E .................................................. 80
6.24 SCSI adapter for the ThinkPad ............................................ 81
6.25 Stand-alone z/OS restore .................................................... 81
6.26 FSI Channel Adapters ....................................................... 84
  6.26.1 Parallel Channel Adapter (PCA) ................................... 84
  6.26.2 ESCON Channel Adapter (PCA) .................................... 84
6.27 Positioning with other small S/390s ............................................. 84
6.28 Useful Linux commands ............................................................. 86

Chapter 7. Frequently Asked Questions ........................................... 89

Appendix A. Operational listings ..................................................... 97
Basic definitions for a single HDD ..................................................... 97
Shell script for a single HDD ............................................................. 99
Definitions for two HDDs ................................................................. 99
Shell script for two HDDs ................................................................. 100
Customized Offerings Driver definitions ........................................... 101

Appendix B. FLEX-ES definitions and commands .............................. 103
System definitions ............................................................................. 103
Resource definitions ......................................................................... 105
  Emulated control unit types ............................................................ 106
  Emulated device types .................................................................. 106
  Typical resource definitions ........................................................... 106
  Cloned devices .............................................................................. 111
  Compiled files ............................................................................... 111
Common rules .................................................................................. 112
The resadm command ...................................................................... 112
CLI commands ................................................................................. 113

Related publications ........................................................................ 115
IBM Redbooks .................................................................................. 115
  Other resources ............................................................................ 115
Referenced Web sites ....................................................................... 115
How to get IBM Redbooks ............................................................... 115
  IBM Redbooks collections .............................................................. 116

Index .................................................................................................. 117
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 disclaimers 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.
Trademarks

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

- Redbooks™
- AIX®
- CICS®
- DB2®
- DFS™
- DFSMSdss™
- ESCON®
- FICON™
- FlashCopy™
- IBM®
- IMS™
- Multiprise®
- MVS™
- Nettfinity®
- NUMA-Q®
- OS/2®
- OS/390®
- Parallel Sysplex®
- PartnerWorld®
- PS/2®
- RACF®
- S/370™
- S/390®
- ServeRAID™
- ServerGuide™
- ThinkPad®
- Ultrabay™
- VM/ESA®
- VSE/ESA™
- VTAM®
- WebSphere®
- xSeries™
- z/OS™
- z/VM™
- zSeries™

The following terms are trademarks of other companies:

- ActionMedia
- LANDesk
- MMX
- Pentium
- ProShare
- Microsoft
- Windows
- Windows NT
- Java
- Sun Microsystems
- C-bus
- Corollary
- UNIX
- The Open Group
- SET
- SET Secure Electronic Transaction
- WebSphere
- IBM
- OS/390

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

A ThinkPad Enabled for S/390 (ThinkPad/EFS) or a Netfinity Enabled for S/390 (Netfinity/EFS) are the smallest S/390-compatible systems currently available that have been tested and approved by IBM. They are based on an IBM ThinkPad or Netfinity running Linux and the S/390 emulation product FLEX-ES. FLEX-ES is a product of Fundamental Software, Incorporated (FSI) of Fremont, California. The resulting system can run current S/390 operating systems, such as z/OS.

Small systems such as these are attractive for education, development, and less demanding production operations.

This IBM Redbook describes the installation and use of these systems, based on usage of a particular z/OS package. This package is available only to members of the IBM S/390 Partners in Development (PID) organization who obtained systems through this organization. Consequently, this document is intended primarily for this group.

The information about Netfinity/EFS systems under Linux should be regarded as preliminary planning information. General support for Netfinity/EFS under Linux was not yet available at the time of writing.

The author

Bill Ogden is a retired IBM Senior Technical Staff Member, working part-time on favorite projects at the International Technical Support Organization, Poughkeepsie Center. He specializes in small S/390 and z/OS systems, writes extensively, and teaches ITSO workshops relating to these areas. Bill has been with the ITSO since 1978. Many of his projects for the last several years have been related to the IBM S/390 Partners in Development organization.

Thanks to the following people for their contributions to this project:

Jim Obrizok and Frank Yolton, of the S/390 Partners in Development program, who coordinated the PID and AD CD-ROM elements of the project.

Scott Carter, Martin Groen, Peter Ward, David MacMillan, Ray La Croix, Gary Eheman, Tom Valerio, and David Clare of Fundamental Software, Inc., who provided considerable help with the details of FLEX-ES planning, installation and usage.

Mary Beth Bradley, IBM Poughkeepsie, who helped organize the FLEX-ES projects.
Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:
  
  ibm.com/redbooks

- Send your comments in an Internet note to:
  
  redbook@us.ibm.com

- Mail your comments to the address on page ii.
Chapter 1. Basic concepts

Netfinity/EFS and ThinkPad/EFS systems provide small S/390 platforms. The EFS portion of the name is derived from "Enabled For S/390." EFS systems may be used to run z/OS (and OS/390), VM/ESA (and z/VM), VSE/ESA, and Linux for S/390. They may also be used for older versions of these operating systems. This document addresses only z/OS use, but the reader should not infer that the EFS is limited to z/OS.

We use z/OS (in 31-bit mode) for all the examples in this redbook. However, you could use any release of OS/390, any 31-bit version of VM, or any release of VSE. Future versions of FLEX-ES are intended to support 64-bit operation, but this function was not available when this document was being written.

EFS systems can serve many purposes. See “Positioning with other small S/390s” on page 84 for a discussion of the positioning relative to other small S/390 platforms.

1.1 Purpose of this redbook

This IBM Redbook describes only Linux-based EFS systems and describes the installation process for particular configurations of these systems. Earlier FLEX-ES production systems (emulating S/370, S/390, and zSeries) usually have been installed on a number of SCO UNIX bases. These are not discussed in this document.

FLEX-ES for Linux-based ThinkPads has been available since mid-2001. Support for Linux-based Netfinity systems was not quite complete at the time of writing, so some of the information in this document is preliminary. Netfinity support is more complex because a Netfinity/EFS system can include multiprocessor operation, hardware adapters for S/390 parallel and ESCON channels, and ICA ports.

The specific combinations described in this document, EFS systems running z/OS AD CD-ROM software, are available to members of the IBM S/390 Partners in Development organization who have obtained their systems through the PID program.

1 The Netfinity series is now known as the xSeries. Existing FLEX-ES documentation uses the Netfinity name, and the Netfinity/EFS name has been used for some time. For these reasons this redbook uses the term “Netfinity” instead of “xSeries.” IBM literature, as well as newer systems, use the name xSeries exclusively. You need to be flexible and regard the names as interchangeable for the purposes of this book.
This redbook is intended primarily for this group.

EFS systems for PID members are available only through IBM business partners, and are not available directly from IBM. In normal situations, the business partner performs the initial machine setup—including hardware installation, basic Linux operating system installation, initial disk configuration, and FLEX-ES installation. This redbook describes these same steps and, in a sense, describes steps that a PID member may not need to perform.

However, we believe this material will be useful for PID members for several reasons:

- Understanding how a system is set up, even though someone else may have done it for you, leads to a better understanding of the system.
- The system owner may accidently (or intentionally) delete or destroy part of the underlying elements of his system. If this happens, he might ask his IBM business partner for assistance, or he might prefer to rebuild the system himself.
- Significant upgrades to the underlying elements (hardware, Linux, FLEX-ES) may require a partial or complete rebuilding of the system. Again, the system owner has the option of asking his business partner to perform the upgrade, or of doing it himself.

You may see references to three other EFS redbooks:

- *Netfinity Enabled for S/390*, SG24-6501, describes a system based on Caldera’s Open UNIX base. (This is the most recent evolution of SCO UNIX.) This book is still current and relevant for these systems.
- *ThinkPad Enabled for S/390*, SG24-6507, is replaced by this redbook.
- *NUMA-Q Enabled for S/390*, SG24-6215, is no longer current.

### 1.2 Usage overview

An EFS system can be self-contained in a ThinkPad or Netfinity machine. That is, the system can provide the required z/Series emulation plus several 3270 terminal sessions for a z/OS operator console and VTAM (TSO, CICS, and so forth) terminals. It can also provide ASCII telnet sessions for direct logon to z/OS UNIX System Services. The number of terminal sessions (3270 and ASCII telnet) is limited by the screen space available for reasonable use on the display. For practical purposes, this is a single-user system.

We can easily make a multi-user system by connecting Linux to a LAN. This typically uses the integrated Ethernet interface available in both Netfinity and ThinkPad systems.

If we have sufficient memory in the Netfinity or ThinkPad, we can simultaneously emulate multiple z/Series machines. This is not quite the same as using LPARs in a real zSeries system, but it is similar in many ways. The multiple emulated systems can communicate with each other over the LAN or by emulated CTC connections.

---

2 This organization (usually known as PID) is part of the larger IBM PartnerWorld for Developers group (PWD) and both names are used somewhat interchangeably.
With the configuration shown in Figure 1-1, a typical client might use a Windows-based laptop connected to the LAN. PCOM\(^3\) can provide 3270 sessions that appear as locally-connected, non-SNA sessions to z/OS. These sessions can be used for z/OS operator consoles or VTAM applications such as TSO, CICS, and so forth. The telnet application provided by Windows can be used to connect to Linux on the ThinkPad/EFS system (if such access is enabled) or to z/OS UNIX System Services (if this access is enabled). If properly enabled, the client systems can ftp to Linux or z/OS, connect to a Web server under z/OS, and so forth. In configurations like this, Linux (on the ThinkPad/EFS machine) and z/OS (on the same machine) will have different IP addresses.

The EFS host system in this illustration is a ThinkPad, but exactly the same illustration could apply to a Netfinity/EFS host.

The performance of an emulated zSeries machine depends on the performance of the particular ThinkPad or Netfinity used, of course. Typical EFS performance (using a single processor for S/390 emulation) ranges from about 12 MIPS to perhaps 20 MIPS, depending on the exact nature of the workload. With multiple processors, Netfinity/EFS machines can be faster—ranging up to 50 MIPS or more.

There are a number of functions not available for EFS systems and details about these are mentioned, as appropriate, throughout this document. These include:

- Parallel Sysplex, Coupling Facility processors, and Coupling Facility links are not available.
- FICON channels are not available.
- Unique control unit hardware functions are generally not emulated, such as the cache and compression functions of 3990-6 control units.
- Advanced zSeries (and S/390) hardware recovery functions, such as multiple I-units in processors and the Application Preservation Feature, are not emulated.
- Cryptographic hardware functions are not available.
- OSA Express functions are not available. (Basic 3172 LCS and SNA emulation is available. Basic OSA functions over Ethernet and token ring are available starting with FLEX-ES release 6.1.14.)

---

\(^3\) PCOM is the common name for the IBM eNetwork Personal Communications product. It provides TN3270e client functions.
1.3 Terminology

EFS descriptions can become confusing if the terminology is not well defined. These are important terms:

- **Processor** means a PC processor in the ThinkPad or Netfinity system.
- **Processor(s) enabled for S/390** refers to the number of PC processors that the FLEX-ES license allows to be used simultaneously for S/390 emulation.
- **Server** means the underlying ThinkPad or Netfinity hardware.
- **Server operating system** means Linux in the server.
- **S/390 CPU** (or simply **CPU**) means an emulated single S/390 CPU engine.
- **S/390 system** means an emulated S/390 (in the EFS system) that might have more than one emulated S/390 CPU engine.
- **Instance** (or **FLEX-ES instance**) means an emulated S/390 system. We can run several FLEX-ES instances if we have sufficient resources.
- **z/OS** means the S/390 operating system. We could also use OS/390, VM/ESA or VSE/ESA, but we elected to work only with z/OS.
- **ThinkPad/EFS** and **Netfinity/EFS** are generic names. Business partners providing these products often have their own names for their specific implementation of the products.
- **Netfinity** is the name of an IBM product line. Newer products use the name **xSeries**.

For any FLEX-ES instance, there is a one-to-one correspondence between processors enabled for S/390 and S/390 CPUs. Running multiple instances is something like running multiple z/OSs in separate LPARs (but is not quite the same). We can run multiple instances of single-CPU S/390s in a ThinkPad/EFS system, for example.

1.4 Basic EFS requirements

Linux-based ThinkPad/EFS and Netfinity/EFS systems have many common details. To a large extent, the ThinkPad version is simply a subset of the Netfinity version. Details include:

- **No unique hardware is used for z/Series emulation.** There is no “S/390 adapter.” Everything is software, with these exceptions:
  - A **dongle** is required. This is a device (match-box size) with a cable that connects to the USB port of your Netfinity or ThinkPad. This provides license control for the FLEX-ES software. You must have a software license key file that matches the encoded serial number in your dongle.
  - **Optional S/390 channel adapters** are available.

- The base operating system of the hardware platform is Linux. In particular, it is Red Hat Linux 7.1 or 7.2. We use 7.2 for all our discussions, but 7.1 can also be used. **No other Linux version is supported.**

- A dual-boot system, probably with a version of Windows as the alternate operating system, is possible. However, we do not describe that option in this redbook. As described here, no Windows system is present on the machines.

- Sufficient memory must be present. An extremely rough guideline is 256 MB plus the size of the zSeries memory you want to emulate. For example, using a 512 MB system (ThinkPad or Netfinity), you could emulate a zSeries machine with 256 MB of real memory. Memory is discussed in more detail in “Server memory” on page 61.

- Sufficient disk space is required. We recommend about 5 GB for Linux, although the actual requirements are considerably less. The remaining disk space required is whatever is required for your S/390 volumes. A 3390-3 requires 2.8 GB, for example.
There is a one-to-one correspondence between the size of a 3390 volume and the amount of Netfinity or ThinkPad disk space required. In addition to space for Linux and space for S/390 volumes, we suggest that you reserve at least 6 GB additional disk as a work area. Disk layout is initially discussed in "Disk planning" on page 13.

- While not absolutely required, you probably want to connect your system to a LAN. With a Linux-based EFS system, using Ethernet, only a single LAN adapter is required. It can be used by both Linux and z/OS (and each has a separate IP address).
- SCSI-connected tape drives may be used. These are commonly used with Netfinity/EFS systems, but not with ThinkPad/EFS systems.
- Netfinity/EFS systems (but not ThinkPad/EFS systems) can use unique hardware adapters from Fundamental Software, Inc. These include:
  - S/390 parallel channel adapters. Two versions are available. One has a single parallel channel, and the other has three parallel channels.
  - ESCON channel adapters. Each adapter provides a single ESCON channel.
  - ICA (integrated communications adapter) cards, providing six communications interfaces. These can run in bisync or SDLC modes. Except for a few special instances, z/OS does not support these interfaces; they are often used by VSE/ESA systems.

See “FSI Channel Adapters” on page 84 for more specific information about these adapters.

### 1.5 FLEX-ES

FLEX-ES is a product of Fundamental Software, Incorporated (FSI) of Fremont, California. FLEX-ES runs under Linux or several UnixWare-based operating systems.

FLEX-ES is a licensed product. The licenses are normally arranged through FSI business partners. A FLEX-ES license specifies the number of PC processors (and their maximum speed) that can be used for S/390 emulation. A FLEX-ES license is keyed to a specific serial number in a hardware dongle that is connected to the USB port of the ThinkPad or Netfinity and will run only in a system that has this dongle connected.

FSI also produces five hardware options that may be used with FLEX-ES. These are:

- A parallel channel adapter, providing one S/390 parallel channel
- A parallel channel adapter, providing three S/390 parallel channels
- An ESCON channel adapter, providing one S/390 ESCON channel
- An ESCON channel adapter, providing four S/390 ESCON channels
- A communications adapter, providing six lines corresponding to S/370 ICA lines

The three-channel parallel adapter card is a full-length PCI adapter; the others are half-length PCI adapter cards. These adapters are not normally be used with a laptop machine.

### 1.6 Linux

In order to provide well-tested S/390 operation, a limited number of Linux distributions have been chosen as the ThinkPad base software. These are the Red Hat 7.1 and 7.2 releases. Other versions and distributions of Linux are not supported.

---

4 The ESCON adapters were not yet available at the time of writing.
Red Hat Linux 7.1 or 7.2 can be used “out of the box” for FLEX-ES systems on a ThinkPad T20 (and possibly on other ThinkPads) using Ethernet LAN connections. Patches to Linux may be needed to use token ring adapters (on either a ThinkPad or Netfinity system). Additional Linux patches are needed to use a multiprocessor server, even if only a single processor is used for S/390 emulation. The business partner that provided your EFS system will have more information about the patches. Additional drivers may be needed for newer video adapters (such as are used on T23 ThinkPads).

This may change in the future, of course. Linux is an ongoing process and the EFS base system is likely to migrate to future Linux kernels at appropriate times. FSI manages EFS development with a high priority for system stability. In general, this means the EFS base operating system will not “chase” small Linux enhancements and may delay movement to major new releases until their stability for the EFS environment has been thoroughly confirmed.

Linux is a large topic. This redbook addresses Linux only to the extent needed to install and use FLEX-ES. There are many clever things that can be done with Linux, and some of these can interact with FLEX-ES and files used with FLEX-ES. Except as needed to install a basic EFS system, we do not explore the things a user might implement using Linux facilities.

### 1.7 ThinkPad versus Netfinity

A ThinkPad/EFS system is appropriate for many tasks, such as:

- Education, for a single owner or as a server running a classroom of clients.
- Small development and testing, typically for a single person. Proof-of-concept work is often in this category.
- Demonstrations, in many different ways.

In our opinion, a ThinkPad/EFS system is not suitable as a production server. Key factors are:

- Non-parity memory is used. This simply lacks the reliability of ECC memory.
- RAID protection is not available for the disks. Furthermore, reliable backup arrangements (to tape, for example) are typically more difficult to provide in a ThinkPad environment.
- ThinkPads are not fully designed for continuous operation. Heat dissipation is a concern, although a docking station helps in this area.
- Disk operation is considerably slower than that of a fast RAID array.

A properly configured Netfinity/EFS system is much more robust and is suitable as a production server. Important factors include:

- ECC memory is used. The importance of this for long-term stable operation cannot be overstated.
- RAID disks should be used, and are assumed in all Netfinity/EFS discussions. RAID 5 is typically used and provides automatic protection against any single disk failure. A hot spare is not normally used in smaller configurations, but would be considered normal in larger configurations.
- Multiple, redundant power supplies are available for the approved Netfinity servers and provide N+1 protection.
- Tape connections are more likely to be available in this environment, making routine backups more practical.
- A Netfinity/EFS system normally has at least two processors. One might be enabled for S/390 emulation, leaving the other for all the Linux processes involved with S/390 I/O emulation, LAN operation, and so forth. Multiple processors can be enabled for S/390 emulation, making a much faster system.\(^5\)
The presence of ECC memory, RAID disk protection, and N+1 power supplies covers the vast majority of typical PC failures—providing a very stable hardware platform.

Each platform has its place. The outstanding characteristic of the ThinkPad is its portability, of course. A secondary factor is the ease of installation of a ThinkPad/EFS system (assuming use of an Ethernet Lan).

### 1.7.1 Hardware restrictions

FLEX-ES is supported only on specific Netfinity (or xSeries) models and specific ThinkPad models. FSI adds to the list of supported models over time.

### 1.8 ThinkPad hardware

ThinkPad/EFS systems for PID members are provided through IBM business partners. They cannot be ordered directly from IBM. Different business partners may start with slightly different base configurations and offer different options. The system we describe in this redbook should closely match most starting configurations offered by business partners.

The basic hardware requirements are:

- A Thinkpad in the A2x or T2x series, or one that is closely compatible, is the only supported base at the time of writing. An example is a T23 ThinkPad. A core requirement is that the processor be a Pentium II or Pentium III. Systems with basic Pentiums or Pentium Pro processors are not supported. Speed should be 800 Mhz or faster.
- A major compatibility requirement is that the X windows support in Red Hat Linux 7.2 (or 7.1) must operate correctly with the ThinkPad display.
- It must have a USB port (for the FSI license dongle).
- It should have 512 MB memory (or more, if available).
- It should have at least one 32 GB (or larger) disk drive.
- The option for a second 32 GB (or larger) drive (swappable in the slot used by the CD-ROM drive and the diskette drive) should be considered.
- It should probably have the integrated Ethernet adapter, and possibly a PCMCIA token ring adapter (depending on your requirements).
- A docking station is recommended, but not required.

ThinkPad models change frequently and the above requirements may need to be adjusted to match current offerings. We used both T20 (IBM type/model 2647-L1U) and T23 (IBM type/model 2647-9KU) ThinkPads while working on this redbook. It was necessary to obtain and install a new video driver during Linux installation on the T23. In fact, our most apparent problem in using various ThinkPad (and Netfinity) models is specifying the appropriate graphics adapter parameters to Linux (or finding additional video drivers).

While it is obvious to most readers, we note that a ThinkPad has a single processor. This processor is used for all processes running under Linux, including the FLEX-ES processes that emulate S/390 operation.

It is possible to place Linux and all the necessary emulated 3390 volumes on a single, large ThinkPad disk drive. The author normally works with two disk drives, one in the base ThinkPad and one in the Ultrabay. This provides more space, and provides an option for backing up emulated volumes by simply copying them to the other drive. Of course, when the second hard drive is in the Ultrabay, a CD-ROM drive or diskette drive cannot be placed in the Ultrabay.

---

5 This requires additional FLEX-ES license capabilities, with a corresponding increase in license fees.
With some ThinkPad models it is possible to boot from a hard disk in the Ultrabay. You might, for example, have Windows on the internal hard disk and Linux/FLEX-ES on a hard disk in the Ultrabay and boot from whichever is appropriate. (You must be a little more creative during initial EFS installation, since this involves reading CDs. One possibility is to temporarily install the Linux hard disk internally in the ThinkPad while installing Linux, FLEX-ES, and z/OS.)

1.9 Netfinity hardware

A number of IBM Netfinity systems may be used as a base machine for an EFS system. If you obtain your system through the PID program, the business partner involved will have a list of possible base systems. The exact list of systems qualified by FSI for FLEX-ES use is frequently updated; any list we might include in this redbook would be out of date by the time it is published.

A core requirement is that a Pentium II or Pentium III processor must be used. Basic Pentium or Pentium Pro processors are not supported. The system speed should be at least 800 Mhz. It must have a USB connector (for the FSI dongle).

We used a Netfinity 5100 (also known as an XSeries 230) simply because we had one and it is an FSI-qualified machine. Our Netfinity 5100 is a bit old and slow compared to more recent xSeries machines, but it was perfectly adequate for our purposes.

Our hardware consisted of the following:

<table>
<thead>
<tr>
<th>IBM PartNo</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8658-41Y</td>
<td>1</td>
<td>Netfinity 5100 Server (866 Mhz, 128MB)</td>
</tr>
<tr>
<td>33L3127</td>
<td>1</td>
<td>512 MB 133 Mhz ECC SDRAM DIMM Additional Memory</td>
</tr>
<tr>
<td>06P5736</td>
<td>1</td>
<td>IBM ServeRAID 4Mx SCSI Controller</td>
</tr>
<tr>
<td>37L7205</td>
<td>5</td>
<td>18.2 GB Ultra 160 Slim-High 10K RPM disk drive</td>
</tr>
<tr>
<td>19K4630</td>
<td>1</td>
<td>Intel Pentium III 866 Mhz Upgrade (Second Processor)</td>
</tr>
<tr>
<td>02K3454</td>
<td>1</td>
<td>PCI Wide Ultra SCSI Adapter</td>
</tr>
<tr>
<td>37L6881</td>
<td>1</td>
<td>xSeries HS Power Supply Expansion</td>
</tr>
<tr>
<td>33L3760</td>
<td>1</td>
<td>xSeries 250W Redundant HS Power Supply</td>
</tr>
<tr>
<td>06P3601</td>
<td>1</td>
<td>10/100 Ethernet Server Adapter</td>
</tr>
<tr>
<td>09N4041</td>
<td>1</td>
<td>12/24 GB DDS/3 4mm Internal SCSI Tape Drive</td>
</tr>
</tbody>
</table>

We did not have several options that a typical Netfinity/EFS customer might have. What we did not use includes:

- Larger memory. Customers wanting to run multiple z/OS instances or define larger emulated S/390 processors would probably order at least 1 GB memory. (We bought our Netfinity when 512 MB was considered very large, and were unable to obtain more memory at the time.)
- Additional disks. The Netfinity 5100 holds six slim-high drives. We arbitrarily ordered only five 18.2 GB drives. Larger drives are also available (and are recommended).
- FSI adapters for S/390 channels. Our particular environment had no S/390 channel devices.
- A display. We used an existing flat panel display.
- More LAN adapters, for connecting to multiple LANs.

These are all good elements. We did not use them simply because we wanted to implement a relatively small Netfinity/EFS system.
Netfinity 5100 Server
This server is a desk-side tower model. It is approximately 17 inches high, 9 inches wide, and 26 inches deep (43 cm x 22 cm x 66 cm). This is somewhat deeper than the average tower PC. Our base machine has a single 866 MHz Pentium III processor, with a 256 KB level 2 cache. The base unit includes 128 MB memory (133 MHz) in one memory slot and uses SDRAM 168-pin DIMM ECC memory. Three additional memory slots are available. The maximum memory size is 4096 MB and this maximum might be used in EFS configurations. Our configuration placed a 512 MB memory module in the second memory slot and retained the supplied 128 MB module in the first slot.

Six bays are provided for slim-high hot-swappable disks. These are accessible from the front panel. A CD-ROM drive (40x - 17x) and a diskette drive are included. Two more half-high fixed bays are available; we used one for a 4mm drive.

Five full-length PCI slots are available. Three are 64-bit slots and two are 32-bit slots. In our configuration, one 64-bit slot is used by the RAID adapter, one 32-bit slot is used by an Ethernet adapter, and another 32-bit slot is used by a SCSI adapter.

A SCSI adapter is built onto the planar board and is based on an Adaptec AIC-7899 dual-channel Ultra 160 controller. It is intended to drive internal disks. Since we are using a RAID adapter, this internal SCSI controller is not used.

The integrated graphics controller is based on the S3 Savage4 chip set. It has 8 MB SDRAM and provides up to 1600x1200 resolution with 65536 colors.

ServeRAID-4Mx Ultra160 SCSI Controller
The ServeRAID-4 family of Ultra160 SCSI controllers offers nine RAID levels and supports up to 2 terabytes of disk capacity. These 64-bit PCI controllers provide FlashCopy, Logical Drive Migration, and adapter/cluster failover. The 4Mx version is a double-channel controller with 64 MB ECC cache and supports up to 14 disk drives per channel. The large cache is automatically used by the controller to further enhance performance of the disk subsystem. The cache has integrated battery backup, and this is the primary reason this adapter was chosen.

The adapter is easy to install. A single cable is moved from the Server internal SCSI connector to the 68-pin SCSI connector on the RAID adapter.

A RAID adapter is considered a requirement for Netfinity/EFS systems. Neither PID nor FSI will support Netfinity/EFS systems that do not have and use RAID adapters.

Disk drives
The 18.2 GB 10000 rpm Ultra160 SCSI hot-swap slim-high drives we ordered are among IBM’s (and the industry’s) best performing drives. Transfer rates are up to 160 MB per second and the average seek time is 5.5 ms. We ordered five of these drives. Allowing for the parity space used by RAID-5, this provided us with approximately 72 GB of usable disk space. The server provides six bays for slim-high hot-swap drives. If we used all six bays (with these disks and RAID-5), we would have approximately 90 GB of useful space.

Our server will also accept 36 GB and larger drives. If we filled all six bays with 36 GB drives and used RAID-5, our effective disk capacity would be approximately 180 GB.
Second processor
The Intel Pentium III 866 MHz Upgrade provides a second processor for the system. It contains its own 256 KB ECC L2 cache, with the ECC functions providing enhanced protection against soft failures in the cache. A Netfinity/EFS system typically dedicates one processor to S/390 emulation and uses the second processor for all Linux and background tasks.

SCSI adapter
We elected to purchase an additional SCSI adapter, with an external connector, so we could attach external SCSI tape drives. We were forced to relocate this adapter from PCI slot 2 to slot 4 when using Red Hat 7.2 Linux.

4mm tape drive
A 4mm tape drive is included in many Netfinity/EFS configurations, primarily because many IBM software products are available on 4mm media. Experience with earlier entry-level S/390 systems has shown that 4mm tapes (and tape drives) are often not suitable for “mainframe-like” tape usage. We installed an older DDS2 4mm drive that we happened to have available, instead of the newer unit listed above.

The mechanical installation was a challenge until we understood the plastic guide rails cleverly shipped inside dummy “option slot” covers on the server. The server included an extra SCSI cable (packed behind the option slot covers). You will probably need a converter to fit the 50-pin connector on the tape drive, converting it to a 68-pin connector. (This is IBM FRU number 92F0324.) We connected one end of the SCSI cable to the B channel of the integrated SCSI adapter, and connected one of the connectors on the cable to the tape drive. We removed the address jumpers in the tape drive, making it SCSI address 0.

Redundant power supplies
The xSeries Hot-Swap Power Supply Expansion Kit replaces the original power backplane in the Netfinity. The new power backplane can accept up to three power supplies, and is required in order to install redundant power supplies.

The xSeries 250W Redundant Hot Swap Power Supplies can provide additional power for a Netfinity server, or function as a redundant power supply. Two of these may be installed, providing a total of three power supplies. Installing all three power supplies provides both additional power and a redundant unit.

Ethernet adapter
The base Netfinity unit contains one Ethernet connection. The additional Ethernet adapter provides a second LAN connection. This is not required, since a single LAN adapter can be shared between Linux and operating systems running under FLEX-ES. However, multiple adapters might be considered for performance or redundancy reasons.

Mouse
FSI recommends the use of a three-button mouse. We did not see a strong requirement for this and used the two-button mouse that came with the Netfinity system.

---

6 The integrated SCSI adapter does not have an external connector.
Chapter 2. Hardware and Linux installation

There are enough differences between Linux installation on a ThinkPad and Linux installation on a Netfinity that we will describe them separately.

ThinkPad/EFS installation is simpler than Netfinity/EFS installation for these reasons:

- The appropriate ThinkPad models can be used “out of the box.” We are unaware of any requirements for BIOS upgrades. This is not to say that you should not install BIOS upgrades. We are unaware of any that are required for using FLEX-ES.
- There are no hardware RAID functions to customize.
- A Netfinity/EFS system is assumed to have a RAID adapter, and appropriate drivers must be downloaded and added to the Linux system.
- However, Netfinity/EFS installation is easier in one respect: all the disks, the CD-ROM drive, and the diskette drive can be used at the same time, whereas it may be necessary to use a single Ultrabay for the CD-ROM drive, diskette drive, and additional hard disk on a ThinkPad.

2.1 FLEX-ES disk usage

Earlier FLEX-ES systems used raw disks (known as raw disk slices in SCO UNIX systems) to obtain better disk performance for emulated S/390 DASD. The Linux equivalent is the use of raw devices. FSI recommends using this approach for best performance and system stability. There are several factors behind this recommendation:

- The path length through the server operating system is shorter when raw devices are used instead of Linux (or UNIX) file systems. Furthermore, the disk buffering provided for Linux (or UNIX) file systems is not optimal for FLEX-ES operation.
- Fragmentation within a Linux or UNIX file system can impact overall system performance. The use of raw disks/devices prevents fragmentation of the files containing emulated S/390 volumes.
- Older UNIX systems did not support files larger than 2 GB, and this was a problem for emulating 3390-3 and 3390-9 drives.
- Using raw disk space prevents these disks from unexpectedly filling up. For example, circumstances might cause UNIX or Linux to write huge amounts of log or trace

---

7 This is not to say that you should not install BIOS upgrades. We are unaware of any that are required for using FLEX-ES.
8 A raw disk is a disk partition that does not contain a file system. UNIX sees it simply as a linear byte device.
information, such that the file system being used becomes full. In extreme cases, this could crash the system.

We do not describe the use of raw disks/devices in this redbook, for the following reasons:

- We believe the performance of Linux file systems is more than adequate for the user community addressed by this redbook, based on current processor speeds and Linux development.
- We describe the use of separate Linux file systems for emulated S/390 DASD as a way to minimize fragmentation and to avoid unexpected out-of-space conditions.
- We create another file system (named /holding in our examples) for general utility functions that might cause disk fragmentation. (We also use the ample free space in the root file system for work areas such as /tmp.)
- Linux file systems can handle files larger than 2 GB.
- Use of normal Linux file systems simplifies installation and use of the EFS system.

Disk fragmentation can be bad for performance. This observation is not unique to FLEX-ES, of course. For FLEX-ES operation, the performance-related disk files are the emulated S/390 volumes. There are relatively few of these (one per emulated 3390 volume, for example). By placing these in clean file systems (/s390 and /s391 in our examples) and never using these file systems for other purposes, we can avoid fragmentation of these large files.

Production Netfinity servers should use the raw disk interface recommended by FSI, and documented in their FLEX-ES manuals. Business partners (for PID members, or for their normal FLEX-ES customers) may prefer to use raw disks/devices as a method of obtaining the best possible FLEX-ES performance. In this case, their installation details will differ somewhat from the following descriptions.

### 2.2 Network connections

Your system can be connected to the Internet or any other LAN network. However, you should consider a few issues before installation. These include:

- You can use an external DHCP server (already on your LAN) to obtain an IP address for Linux. You cannot obtain an IP address for z/OS TCP/IP this way. (You are not required to use z/OS TCP/IP, of course. You need it if you want to ftp directly to z/OS, telnet to UNIX System Services, and so forth.) You will need a static IP address in order to use z/OS TCP/IP.

- You (or other users) can connect to your z/OS through the LAN by connecting to the FLEX-ES Terminal Solicitor (which is a process under Linux, and uses your Linux IP address). These connections appear as local 3270 terminals to z/OS, and do not use z/OS TCP/IP. This can be done if Linux obtained its address via DHCP or if you assigned a static IP address to Linux.

- If you configure your Linux to use a Domain Name Server (DNS), then you must be connected to the LAN (and DNS) in order to use FLEX-ES 6.1.14 or earlier releases.

- FLEX-ES releases after 6.1.14 are changed so that FLEX-ES will start whether or not you defined domain names and whether or not you are connected to a Domain Name Server.

---

9 This discussion applies to any S/390 operating system run under FLEX-ES, of course, and is not unique to z/OS.
Our step-by-step instructions assume you are connected only to a small private LAN, with no DHCP or DNS, or are not connected to any LAN. The examples use private LAN addresses in the 10 address range. If you will be connected to a larger LAN, we suggest you install our way first (with no LAN, or with a small temporary LAN\textsuperscript{10}). Once you have Linux, FLEX-ES, and z/OS working in this basic environment, you can then extend (or reinstall) your system for a larger environment.

### 2.3 ThinkPad installation

Before installing Linux, we installed all our ThinkPad options: additional memory, internal Ethernet port, the CD-ROM drive, and a 32 GB disk drive\textsuperscript{11} in the standard disk position. We did not install our Ethernet PCMCIA card at this time. (In fact, we never used the PCMCIA adapter because we could successfully share the integrated Ethernet adapter between Linux and z/OS TCP/IP.)

We started the system and selected F1 for BIOS setup. Working with BIOS functions, we verified that we had 512 MB of memory and that the internal Ethernet adapter was recognized. We set the time and date. In the Startup section, we verified that the Boot device List F12 Option was enabled; this lets us temporarily select the CD-ROM as a boot device. Our normal Boot List contained Removable Devices (that is, a diskette) followed by Hard Drive.

After exiting the BIOS setup function and performing power-off/power-on functions, we had an initial prompt to provide a temporary boot device list. Pressing F12 provided this list. The CD-ROM was the third element in the list, and this is needed to start Linux installation.

#### 2.3.1 Disk planning

We planned our T20 ThinkPad disk usage as shown in Figure 2-1 on page 14. We placed all of Linux in the root file system, for which we allocated about 3 GB.\textsuperscript{12} We allocated a small swap partition. The remainder of the internal hard disk (about 20 GB in our case) was used for the /390 file system.\textsuperscript{13} In this area we placed the four 3390-3 volumes needed to IPL z/OS (using the AD CD-ROM system). These four volumes require about 11.5 GB. Some of the remaining space should be kept free as a staging area for loading the second hard disk in the Ultrabay.

**Staging**

The ThinkPad Ultrabay can hold a second hard disk\textsuperscript{14} or a CD-ROM drive or another device. However, it cannot hold a hard disk and a CD-ROM drive at the same time. If we want to transfer files from a CD to the second hard disk, we need to stage this activity. That is, we install the CD-ROM drive in the Ultrabay and copy files from CD to the internal hard disk. We then remove the CD-ROM drive, install the second hard disk in the Ultrabay, and copy the files from the internal hard disk to the second hard disk. We then delete the files on the internal hard disk.

\textsuperscript{10} This can consist of a single Ethernet hub or token ring MAU.
\textsuperscript{11} Business partners providing ThinkPad/EFS systems were switching to 48 GB drives at the time of writing. This change has no effect on the comments here, except that it makes a single-disk system more attractive.
\textsuperscript{12} A small amount of space is also needed for a /boot partition, and so forth. The exact details are described later.
\textsuperscript{13} We use file system names of /s390 and /s391. These names are arbitrary; you could use any name.
\textsuperscript{14} We say “second” hard disk, but we could have any number of hard disks in Ultrabay carriers. Of course, only one can be mounted in the Ultrabay at any given time.
There are many variations of disk usage possible and we do not attempt to describe all the things you might do. For example, you might not use a second hard disk at all. Or, you may be able to connect an external CD-ROM drive through a USB port while a second hard disk is mounted in the Ultrabay. In any event, you should think about your disk layout. Linux and FLEX-ES require relatively little space and can be reinstalled quickly if you change your plans. Emulated S/390 volumes require large amounts of space and completely reloading these can take hours.

Our installation of the second hard disk is described in “Using a second Linux hard disk” on page 63.

2.3.2 Linux installation

We purchased a Red Hat Linux 7.2 standard package in a local store. (They also had a Deluxe version and a Server version; we selected the basic version.) This consists of seven CD-ROMs. The first two contain the system; the remainder contain source code, documentation, and various applications. You may have an alternate package that contains only two CDs containing the basic Linux system. Only these two CDs are needed for the installation described here.

We booted from the first Linux CD (using the ThinkPad F12 option to select a temporary boot device). This produced a Red Hat logo screen and offered the choice of graphics mode or text mode installation. We selected text mode, primarily because it was easier to document for this redbook. We then went through a number of installation prompts:

- Language: English
- Keyboard: US
- Mouse: Generic 3-button mouse (Use Tab key to change selection)
- Welcome to Red Hat Linux: OK
- System Type: Custom System (Do not select Laptop or Workstation)
- (Possible message about “Bad Partition Table”; if so, select Initialize)
- Partition: Manually Partition (If needed, of course)
- Select partition tool: Disk Druid

You can select fdisk instead of Disk Druid, and you should use whichever one you find most comfortable. We found that the Linux fdisk was just different enough from DOS fdisk to be confusing, so we used Disk Druid.
Our ThinkPad disk had no installed partitions and was 100% available for allocation. Using Disk Druid, we allocated four partitions. The Red Hat 7.2 resulting display was like this:

<table>
<thead>
<tr>
<th>Start</th>
<th>End</th>
<th>Size</th>
<th>type</th>
<th>mount point</th>
</tr>
</thead>
<tbody>
<tr>
<td>/dev</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>/dev/hda1</td>
<td>1</td>
<td>5</td>
<td>36M</td>
<td>ext2</td>
</tr>
<tr>
<td>/dev/hda2</td>
<td>6</td>
<td>4134</td>
<td>30483M</td>
<td>extended</td>
</tr>
<tr>
<td>/dev/hda5</td>
<td>6</td>
<td>412</td>
<td>3004M</td>
<td>ext2</td>
</tr>
<tr>
<td>/dev/hda6</td>
<td>413</td>
<td>433</td>
<td>155M</td>
<td>swap</td>
</tr>
<tr>
<td>/dev/hda7</td>
<td>434</td>
<td>4134</td>
<td>27323M</td>
<td>ext2</td>
</tr>
</tbody>
</table>

Your hda numbers may differ, depending on the order in which you created the partitions.

We selected an arbitrary size, 155 MB, for a Linux swap partition. In the light of later experience, this could probably be smaller. However, it is a small fraction of our available space and appeared to be a safe choice. The Red Hat documentation indicated that 32 MB was the maximum possible boot partition size, so we selected that. Again, it could probably be smaller.

It may take some experimentation with Disk Druid to obtain the desired configuration. We suggest simply working with the Disk Druid options (Add, Delete, Edit) until you succeed. It may be inconvenient to change the partitions later, so we suggest you take time to get it right now. Remember to scroll within some of the Disk Druid text windows.

The documentation did not provide much guidance for selecting the amount of space for Linux itself. We elected to put all of Linux into the root file system, and arbitrarily assigned 3000 MB for this. (If you elect to install "everything" in RH7.1, you will need about 2.5 GB. Our more minimal installation actually used about 1.1 GB.) If you have more Linux or UNIX experience, you might want to create several file systems for Linux instead of placing everything in a single root file system. Other than having about 25 MB available in /usr (for later FLEX-ES installation and working files), there are no special requirements for your Linux disk configuration.

We then assigned all the remaining space on this drive to a partition with mount point /s390. This name is arbitrary, but clearly indicates the intended purpose of the partition. Remember that the disk layout we describe is not required; it is merely one way to arrange a system.

At this point, we had only a single hard disk installed. Our second hard disk will eventually replace the CD-ROM drive; we cannot have both the CD-ROM drive and the second hard disk installed at the same time. We initialized the second drive later.

The installation prompts continued with fairly basic items:

- Select boot loader: LILO
- Boot Loader Configuration: /dev/hda Master Boot Record (MBR)
- Boot Loader Configuration: (Enter) (no special options required)
- Boot Loader Configuration: /dev/hda5 ext2 Linux (select for loading)

When time permits, we intend to update our recommendations to use grub as the standard loader. Many Linux users find grub easier to set up and easier to use.

---

15 Most Linux documentation recommends a swap partition the same size as your memory. That is, with 512 MB memory, the swap size should be 512 MB. We elected to make a smaller swap size, but you can create the recommended size if you wish.
16 We intentionally specified considerably more space than we needed. We assumed the extra space would be useful for a variety of purposes. Disk Druid expands allocations to even cylinder boundaries.
17 Separate file systems for /tmp and /home are the most common alternatives. However, for the purposes of an EFS system, placing everything in root works well. There is an exposure that logging or temporary files may fill the root file system, causing a system crash. We have never had this happen. Separate file systems for /tmp, /home, and possibly /var would lessen this exposure.
Next you must specify a network address for your system:

Network Configuration:
- [ ] use bootp/dhcp (deselect this)
- IP address: 10.20.30.30
- Netmask: 255.255.255.0
- Default gateway: (if you have one; we left this blank)
- Primary name server: (if you have one; we left this blank)
- Secondary name server: 
- Ternary name server: 

If you have multiple LAN adapters, the system may not assign your IP address to the adapter you want to use. You should verify your network setup after the basic Linux installation is completed.

Hostname Configuration:
- Hostname: t20
- Security level: No Firewall (use the space bar to change it)
- Time Zone: (as appropriate)
- Root password: xxxxxxxxx (must be at least 6 characters)
- Add user: aaaaaaa (use your name)
- User password: xxxxxxxxx (must be at least 6 characters)

The No Firewall security option was appropriate for our purposes, but may not match your needs. As far as we know, this has no particular FLEX-ES security implications and you can select the options appropriate for your needs. Select any user name you like for the Add User function, but do not select the name flexes. (flexes is a special name for FLEX-ES and will be automatically installed later.) The installation process will automatically create a group with the same name as the user name you specify, and add this user to the group. For example, if you specify username ogden, the installation process will automatically create group ogden and add user ogden to group ogden.

OK to Exit from the User Account routine

Authentication Configuration: (RH7.2 prompts slightly different)
- [*] Use shadow passwords
- [ ] Enable MD5 Passwords
- [ ] Enable NIS
- [ ] Enable LDAP
- [ ] Enable Kerberos

OK

Next you select the general packages you want installed. The list shown here is for Red Hat 7.2. The 7.1 list is similar, but has fewer options.

Package Group Selection: Customized (you need to scroll the following list)
- [*] Printing support
- [*] Classic X Window System
- [*] X Window System
- [*] Laptop Support
- [*] GNOME
- [ ] KDE
- [ ] Sound and Multimedia Support
- [*] Network Support
- [ ] Dialup Support
- [*] Messaging and Web Tools
- [ ] Graphics and Image Manipulation
- [ ] News Server

---

18 FLEX-ES needs to use TCP/IP port 24 on Linux for its Terminal Solicitor function. Almost all FLEX-ES installations will require this. FLEX-ES will also use port 555 if you have remote FLEX-ES resources installed. The typical ThinkPad/EFS user will probably not use this, especially for initial FLEX-ES use. If you install firewall functions, be certain that port 24 is not blocked.
Gnome and KDE are competing implementations of an X Windows-based desktop environment. You need at least one of them. Experienced Linux users often have strong preferences for one or the other package. We had no strong preferences and arbitrarily selected gnome.

You may select other options. We did not find documentation explaining exactly what each choice implied. As far as we know, additional selections have no impact on FLEX-ES operation.

You may receive the following prompts:

Some of the packages you...selected...require packages you have not selected. ...
(*) Install packages to satisfy dependencies

OK

The installation program then probes the video configuration. On our T20 ThinkPad (IBM 2647-L1U), Linux sensed the following:

Video card: S3 Savage/MX
Video RAM: 8192
OK

On a T23 ThinkPad (IBM 2647-9KU), Linux was unable to sense the graphics configuration. If this happens, you should select the option to Skip X Configuration at this time.

Continue with the installation:

Complete installation log in /tmp/install: OK
Now format file system(s) (took about 1 minute)

Transferring install image to hard disk
Package installation: 538 packages to install (RH 7.2)

Installation took 10 - 15 minutes. It called for the second CD-ROM when it was approximately half finished.

Create boot diskette: No (could not switch to diskette drive)

The next steps specify your system monitor. For RH 7.2, we selected the following for our T20 ThinkPad:
IBM 9513 T55A TFT Monitor
(Do not change the default synchronization rates)

x Customization:
  Color depth: 16 bits
  Resolution: 1024x768
  Default desktop: Gnome
  Default login: Graphic

Unfortunately, this Linux distribution does not have specific display parameters for ThinkPad displays. As a guess, we specified our ThinkPad display as an IBM T55A (which is a flat-panel LCD display) and this has worked well. As part of the monitor setup for 7.1 (although it is more concerned with video adapter parameters), we specified 4 MB video memory, no clock chip (as recommended by the 7.1 installation dialog) and 24 bit x 1024 x 768 resolution. While this may not be an optimum definition for ThinkPads, it works well and provides an excellent display.

After rebooting and logging into Linux as root, we tried the following:

```
# df -h           (display file system usage)
# cd /proc
# cat cpuinfo
# cat iomem
# cd /
# ls -al          (verify that normal root entries are present)
```

### 2.3.3 Installation notes

The **Custom System** installation option was required in order to have both X Windows and xinetd active. The **Laptop Installation** and **Workstation Installation** choices both omit xinetd support, and the **Server Installation** choice omits X windows. We require xinetd functions and X Windows functions and were forced to use the Custom System installation option.

After the installation outlined above, you should be able to log into the system and ping another host on your subnet.\(^{19}\) If the gateway IP system you specified (if any) is functional, you may be able to access remote hosts.

**Red Hat 7.1 network services**

For Red Hat 7.1, you may note that other systems cannot telnet to your ThinkPad Linux.\(^{20}\) If you want to permit telnet or ftp into your machine, you need to do the following:

```
# /sbin/chkconfig telnet on  (to permit incoming telnet)
# /sbin/chkconfig wu-ftpd on (to run an ftp server)
```

If this does not work for some reason, you can edit the relevant files as follows:

```
# cd /etc/xinetd.d
# vi telnet
      change “Disable = Yes”
      to “Disable = No”
      (save and exit from your editor)
Stop and restart your Linux system, OR force xinetd to restart:
# ps -ef | grep xinetd
# kill -s USR1 pidnumber  (the PID number for xinetd)
```

\(^{19}\) You may have problems if you ping systems on other subnets. The remote system must have a defined route back to you in order for the ping to work. Testing with another system on your local subnet avoids routing problems.

\(^{20}\) This is not required for FLEX-ES operation.
Similar editing of the wu-ftpd file in the same directory will enable the ftp server.

**Red Hat 7.2 network services**
The custom installation options we selected did not install telnet and ftp servers. We mounted the first Red Hat CD and used the following commands to install these services:

```
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom/RedHat/RPMS
# rpm -i telnet-server-0.17-20.i386.rpm
# rpm -i wu-ftpd-2.6.1-18.i386.rpm
# /sbin/chkconfig telnet on
# /sbin/chkconfig wu-ftpd on
```

We determined the names of the two packages by simply browsing the contents of the RPMS directory (on the CD), looking for likely names.

**X Windows configuration**
Linux is unable to sense the graphics adapter in some systems, including the T23 ThinkPad we used. This implies that Linux does not have a driver for the adapter. Our T23 ThinkPad has an S3 SuperSavage/IXC16 (16 MB) adapter with a 1400x1050 LCD display. We needed to download an appropriate driver from the www.s3graphics.com site. At the time of writing, the required driver was in the Savage_4.1.0_binary.tgz file. This file can be processed with the commands:

```
# cd /tmp
# tar zxvf Savage_4.1.0_binary.tgz
```

You might want to rename the original /usr/X11R6/lib/modules/drivers/savage_dvr.o module to something like savage_dvr.o.old before you replace it.

```
# cp savage_drv.o /usr/X11R6/lib/modules/drivers/savage_dvr.o.old
# cp s3switch /usr/local/bin/ (optional)
```

You might want to rename the original /usr/X11R6/lib/modules/drivers/savage_dvr.o module to something like savage_dvr.o.old before you replace it.

```
# cp savage_drv.o /usr/X11R6/lib/modules/drivers/savage_dvr.o.old
# cp s3switch /usr/local/bin/ (optional)
```

You might want to rename the original /usr/X11R6/lib/modules/drivers/savage_dvr.o module to something like savage_dvr.o.old before you replace it.

```
# Xconfigurator
    select the S3 Savage4 card
    select the Generic Laptop Display Panel 1400x1052
    don't probe for screen configuration
    Select 16MB memory
    Select No Clock Chip Setting
    select 24-bit color 1400x1050
    select the option to test the X Windows configuration
    quickly select Yes if you can read the graphics screen
    Select the option for a graphics logon
# shutdown -h now
```

Remember to disable the screen saver in gnome to avoid the problem mentioned in “System freeze” on page 20.

**Token ring support**
A patch may be needed to Linux in order to use token ring adapters. (This is for both ThinkPad and Netfinity systems.) The change is to net/802/tr.c in the Linux source tree. FSI provides a fix in a file named linux_tokenringpatch, and this may be applied by the Business Partner that supplies your system.

Exactly when is this patch needed? The best information we have is this:
A system lock-up MAY occur if the kernel needs to add routing information to the token ring route tables. This can occur when a frame is received in which the source routing bit is on AND the destination MAC address is neither a broadcast nor a multicast address.

We have used token ring interfaces with a number of Linux systems (some with FLEX-ES, and some without it) without problems (and without applying the patch).

**Verify networking setup**

We discovered that an unattached LAN port might cause generally sluggish operation of Linux. In one case we used a token ring PCMCIA card for LAN connectivity. Nothing was connected to the Ethernet port in the ThinkPad. System operation was sluggish. On investigation, it appeared that Linux was starting the Ethernet port and looking for a DHCP server. We disabled the port (by deselecting the “activate device when computer starts” option in the Network configurator).

Be aware that Linux may configure and start all your LAN ports automatically. Check that the IP address you specified during installation is associated with the correct LAN adapter. Check that DHCP is not being used (unless you want it, of course). Use the Network configuration tool to modify the setup. We found it necessary to reboot in order to make our changes effective.

**System freeze**

We had a minor problem with both Red Hat 7.1 and 7.2, when used on T20 and T23 ThinkPads. We do not know if the problem exists for other ThinkPads. By default, gnome invokes a screen saver function after a period of no keyboard activity. This function uses a random series of different patterns for the screen. At least one of these patterns causes the system to freeze. We do not know if a similar problem exists with kde.

One solution is to delete the screen saver function. For the 7.2 release, we did this by selecting the Start Here icon (on the gnome desktop). We then selected Preferences, Desktop, Screensaver, and No screensaver.

**CD-ROM drive access**

Several times we were unable to access the CD-ROM drive on the ThinkPad, using the normal /dev/cdrom name. When this happened, the corresponding entry in /etc/fstab also disappeared. We assume this problem was due to an interaction between changing Ultrabay devices (although we did this with the system turned off) and Linux’s automatic device recognition function.

When this happened, we used a dmesg | less command to scan the startup messages and looked for a reference to the CD-ROM drive. We noted that it was installed as /dev/hdc. (/dev/hdc was also used for the second hard disk, when it was installed in the Ultrabay.) The command mount /dev/hdc /mnt/cdrom let us access the CD-ROM drive in this situation.

### 2.4 Netfinity installation

The general installation sequence includes these basic topics:

- Downloading firmware upgrades for the Netfinity.
- Use of the ServerGuide CD to complete the hardware setup and configure the RAID disks.
- Step through the Red Hat 7.2 installation, including disk partitioning.
- Continue with FLEX-ES installation and AD CD-ROM installation, the same as for a ThinkPad system.
Each of these steps is described in some detail in the remainder of this chapter. We strongly suggest the use of Red Hat 7.2 for Netfinity systems with RAID adapters. In our case, at least, we found the default installation process detected the disks correctly and no special actions were needed.

2.4.1 Download and install upgrades

IBM and FSI strongly recommend that you install BIOS and firmware upgrades before installing FLEX-ES. (For PID members, your business partner may have done the work described in this section.) There are many ways to go about this, and it is not always clear exactly what needs to be done. Looking through the IBM support site, and having a little experience with earlier Netfinity upgrades, we decided that obtaining the latest copy of the ServerGuide CD was the most important step. We also noticed that ServeRAID BIOS and firmware, at a later level than contained on the latest ServerGuide, was available.

One process to obtain these could be the following:

- Find a PC system with at least 500 MB free space, a CD-R writer, and a fast Internet connection.
- Follow this path (site navigation changes frequently, so you may find differences):
  - Support & Downloads
  - Server Downloads
  - select xSeries, Netfinity, NUMA-Q
  - Get fixes
    - Download device drivers, BIOS, and updates
    - Device Drivers by Server
      - select ServerGuide CD
- This should provide a page for downloading a new ServerGuide CD. We followed instructions and downloaded an iso image of this CD. It is over 400 MB. (Be certain to read the instructions if you are using a Netscape browser.)
- We used Easy CD Creator to "burn" a copy of this CD.

We also want to upgrade the ServeRAID BIOS and firmware:

- We returned to the browser and followed this path:
  - Support & Downloads
  - Drivers
    - select Personal Computing
    - select ServeRAID Software
- Download the most recent ServeRAID upgrades. At the time of writing these were:
  - 49p3261.exe 1,009,082 IBM ServeRAID BIOS and Firmware Diskette 1 of 2 version 4.84
  - 49p3263.exe 1,107,074 IBM ServeRAID BIOS and Firmware Diskette 2 of 2 version 4.84
- You might read the README files associated with these diskettes. We did not bother to download these files.
- Do not download the Red Hat driver diskettes.
- Using a DOS window on your PC, execute the two files you just downloaded. You will need to read a license agreement and supply two diskettes for the output files. This creates the diskettes needed for the Netfinity.

Install the ServeRAID upgrade by booting the first diskette. (Our system never asked for the second diskette.) The diskette upgraded the ServeRAID firmware but not the ServeRAID BIOS. We do not know why. The system appeared to run correctly, so we did not investigate this function.
ServerGuide

Boot from the ServerGuide CD. This takes some time, so be patient. You should be presented with a series of screens similar to the following. There are different paths through the ServerGuide, so your operation may not be exactly the same as ours.

Main Menu
[x] Run Setup programs and configure hardware
[ ] Install device drivers and your network operating system (NOS)
[ ] Create system diskettes
[ ] Select another language

Select your operating system
Microsoft(R) Windows 2000
Microsoft Windows NT(R)
Novell Netware
[Red Hat Linux]
SCO UnixWare
Other network operating system

Select the two options shown here (Run Setup, and Red Hat Linux), and then select <next>.

[x] Express
[ ] Custom
[ ] Replication
<next>

Express Configuration Tasks:
[.] Set date and time
[ ] Clear hard disks
[ ] Perform system update
[ ] Configure ServeRAID adapter
[ ] Optimize server performance
[ ] Create system partition
<next> to perform the next indicated task in the list

You will step through each of these tasks, usually by selecting <next> at the appropriate place on the screens. We will not list all the screens involved, since most of the operation is obvious once you start it.

The system update function took some time and automatically rebooted the system. If you read the screens closely, it appears that it will downlevel the ServeRAID firmware you just upgraded. On our system, the ServerGuide upgraded the ServeRAID BIOS to level 4.80.26, but did not change the ServeRAID firmware because it was already at a higher level (due to our diskette upgrade).

RAID configuration

The ServerGuide CD allows you to customize the RAID configuration, using the ServeRAID Manager program on the CD. This is a GUI program and it is difficult to describe the operation in a few words. Our goal was to place all our disk drives in a single RAID 5 array, and then create a single logical drive using all the space in the array. We elected not to use a hot spare drive. Your goals may be different.

The ServeRAID Manager displayed our initial, default configuration as:

<table>
<thead>
<tr>
<th>Logical drv</th>
<th>Size MB</th>
<th>RAID Level</th>
<th>Array</th>
<th>Hot Spare</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>52071</td>
<td>5</td>
<td>A</td>
<td>Yes</td>
</tr>
</tbody>
</table>

After working with the parameters, our final configuration was:
Logical drv   Size MB   RAID Level   Array   Hot Spare
1 New        69428     5            A        No

To get to this point, we deleted the drives from the default array and then added all the drives to a new array. This left no drives for a hot spare.

**Completing the ServerGuide**

We exited the ServeRAID configurator and proceeded with the setup tasks. We selected Other for the Optimize Server Performance item and Keep Current Settings. The Create System Partition step takes some time and reboots the system. This loads Netfinity diagnostic programs in to a small partition. We did not create a replication diskette.

We exited from the ServerGuide at this point. We did not use it to assist in Red Hat installation. (We attempted to use it earlier, and found that something went wrong with the disk device drivers for Red Hat.)

### 2.4.2 Linux installation

Our Linux installation for the Netfinity was almost the same as for the ThinkPad. We booted from the first Linux CD. We were offered the choice of *graphics mode* or *text mode* installation. We selected text mode, primarily because it was easier to document for this redbook. We did not use any disk or RAID driver diskettes. The system automatically loaded several disk drivers, including aic7xxx and ips.\(^{21}\) This took a few seconds.

We then went through the normal Red Hat installation prompts:

- **Language**: English
- **Keyboard**: US
- **Mouse**: Generic 2-button mouse (Use Tab key to change selection)
- **Welcome to Red Hat Linux**: OK
- **System Type**: Custom System (Do not select Workstation)
- **Partition**: Manually Partition (If needed, of course)
- **Select partition tool**: Disk Druid

You can select *fdisk* instead of Disk Druid; use whichever one you find most comfortable.

We found that the Linux *fdisk* was just different enough from DOS *fdisk* to be confusing, so we used Disk Druid. Our plan for disk organization is roughly shown in Figure 2-2.

---

\(^{21}\) If something is wrong with the disk or disk driver environment, the process may hang at this point. Our system hung while installing the aic7xxx driver. We eventually resolved this by moving a SCSI adapter card from PCI slot 2 to slot 4.
Our disk had a single partition already installed (the System Partition, created by the ServerGuide CD) and was otherwise 100% available for allocation. Using Disk Druid, we allocated four partitions. The starting display (with RH7.2) was like this:

```
Start   End    Size   type   mount point
/dev/sda
/dev/sda4    1    7    54M   vfat
Free Space   8    8885   69368M Free Space
```

We selected arbitrary sizes for several partitions:
- 32 M for /boot
- 7000 M for root (considerably larger than needed)
- 3000 M for /holding (another work area)
- 500 M for swap
- All the rest for /s390 (for S/390 emulated volumes)

Our final configuration looked like this:

```
Start   End    Size   type   mount point
/dev/sda
/dev/sda4    1    7    54M   vfat
/dev/sda1   8    11   31M   ext2   /boot
/dev/sda2  12    751   5883M ext2   /s390
/dev/sda3  7542   8885  10501M Extended
/dev/sda4  7542   8437  7000M ext2   /
/dev/sda6  8438   8821  3000M ext2   /holding
/dev/sda7  8822   8855   500M swap
```

It may take some experimentation with Disk Druid to obtain the desired configuration. We suggest simply working with the Disk Druid options (Add, Delete, Edit) until you succeed; remember to scroll (up, down arrows) in some of the text windows. It may be inconvenient to change the partitions later, so we suggest you take time to get it right now. The system complained that our selected swap size might be too small, but we kept it at 500 MB.

If you have more Linux or UNIX experience, you might want to create several file systems for Linux instead of placing everything in a single root file system. Other than having about 25 MB available in /usr (for later FLEX-ES installation and working files), there are no special requirements for your Linux disk configuration.
We assigned all the remaining space on this drive to a partition with the mount point /s390. This name is arbitrary, but clearly indicates the intended purpose of the partition.

The installation prompts continued with fairly basic items:

- Select boot loader: LILO (you can use GRUB if you wish)
- Boot Loader Configuration: install in Master Boot Record (MBR)
- Boot Loader Configuration: (Enter) (no special options required)
- Boot Loader Configuration: /dev/sda5 ext2 Linux (select for loading)

When time permits, we intend to update our recommendations to use `grub` as the standard loader.

Next you must specify a network address for your system:

- Network Configuration:
  - [ ] use bootp/dhcp (deselect this)
  - IP address: 10.20.30.50
  - Netmask: 255.255.255.0
  - Default gateway: (if you have one; we left this blank)
  - Primary name server: (if you have one; we left this blank)
  - Secondary name server:
  - Ternary name server:
- Hostname Configuration:
  - Hostname: nf
  - Security level: No Firewall (use the space bar to change it)
  - Mouse: 3 button PS/2 (this prompt in 7.1 only)
  - Language support: English
  - Time Zone: (as appropriate)
  - Root password: xxxxxxxx (must be at least 6 characters)
  - Add user: aaaaaaa (use your name)
  - User password: xxxxxxxx (must be at least 6 characters)

The No Firewall security option was appropriate for our purposes, but may not match your needs. As far as we know, this has no particular FLEX-ES security implications and you can select the options appropriate for your needs. Select any user name you like for the Add User function, but do not select the name flexes. (flexes is a special name for FLEX-ES and will be automatically installed later.)

The installation process will automatically create a group with the same name as the user name you specify, and add this user to the group. For example, if you specify username ogden, the installation process will automatically create group ogden and add user ogden to group ogden.

- OK to Exit from the User Account routine
- Authentication Configuration: (RH7.2 prompts slightly different)
  - [*] Use shadow passwords
  - [ ] Enable MD5 Passwords
  - [ ] Enable NIS
  - [ ] Enable LDAP
  - [ ] Enable Kerberos
- OK

Next you select the general packages you want installed. The list shown here is for Red Hat 7.2. These are high-level names and each may cause several RPMs to be installed. Our selections represent a basic installation; you may want more options.

---

22 FLEX-ES needs to use TCP/IP port 24 on Linux for its Terminal Solicitor function. Almost all FLEX-ES installations will require this. FLEX-ES will also use port 555 if you have remote FLEX-ES resources installed. The typical ThinkPad/EFS user will probably not use this, especially for initial FLEX-ES use. If you install firewall functions, be certain that port 24 is not blocked.
Package Group Selection: Customized (you need to scroll the following list)

[*] Printing support
[*] Classic X Window System (not clear what is the difference here)
[*] X Window System (not clear what is the difference here)
[ ] Laptop Support
[*] GNOME
[ ] KDE (take your choice between Gnome and KDE)
[ ] Sound and Multimedia Support
[*] Network Support
[ ] Dialup Support
[ ] Messaging and Web Tools
[ ] Graphics and Image Manipulation
[ ] News Server
[ ] NFS File Server
[ ] Windows File Server
[ ] Anonymous FTP Server
[ ] SQL Database Server
[ ] Web Server
[ ] Router/Firewall
[ ] DNS Name Server
[ ] Network Managed Workstation
[ ] Authoring and Publishing
[ ] Emacs
[*] Utilities
[*] Legacy Application Support (not clear what this might be)
[*] Software Development
[*] Kernel Development
[*] Windows Compatible/Interoperation
[*] Games and Entertainment
[ ] Everything

OK

You may select other options. We did not find documentation explaining exactly what each choice implied. As far as we know, additional selections have no impact on FLEX-ES operation.

You may receive the following prompts:

Some of the packages you...selected...require packages you have not selected. ...

(*) Install packages to satisfy dependencies

OK

The installation program then probes the video configuration. Our particular Netfinity was sensed as:

Video card: S3 Savage 4
Video RAM: 8192

OK

Continue with the installation:

Complete installation log in /tmp/install: OK
Now format file system(s) (took about 1 minute)
Transferring install image to hard disk
Package installation: 526 packages to install (RH 7.2)

Installation took less than 10 minutes. It called for the second CD-ROM when it was approximately half finished.

Create boot diskette: Yes (You need a blank diskette)
The next steps specify your system monitor. For RH 7.2, we selected:

- **IBM 9513 T55A TFT Monitor** (This was our monitor on the Netfinity)
- (Do not change the default synchronization rates)

**Customization:**
- Color depth: 16 bits
- Resolution: 1024x768
- Default desktop: Gnome
- Default login: Graphic

The system then reboots. Remove the second Red Hat CD while it is booting. The BIOS will warn you that the boot record has changed. This is expected.

You will probably want to install the telnet and ftp servers, which our chosen installation options omitted. Follow the steps in “Red Hat 7.2 network services” on page 19.

While testing LAN operation of Linux, we noted that the installation process had chosen a PCI Ethernet adapter instead of the integrated Ethernet port as the eth0 device. This was not a problem, although it took a while to determine why we could not ping from a LAN connection to the integrated Ethernet port. Linux detected the integrated Ethernet port and addressed it as eth1. Your system will probably have a somewhat different configuration. You should check your LAN operation and determine what ports are being used. You can change these assignments by using the Red Hat network configuration program, which is available under the **systems** menu.

### 2.4.3 Multiprocessor servers

A server with multiple processors, regardless of how many processors will be used for S/390 emulation, requires two Linux kernel patches for the Linux 2.4.17 kernel. Installing the patches requires access to the Linux kernel source tree. This work is typically done by the business partner providing your EFS system, but we describe here the steps we used. This material was developed by Gary Eheman, of Fundamental Software, Inc.

1. Obtain a copy of the 2.4.17 Linux kernel source tree from FSI, with the proper patches installed. Your business partner should help you with this. This is typically in the form of a compressed tar file. We assume it is named linux-2.4.17.tar.gz. For our examples, we assume it is on CD. Alternately, you may have downloaded it.

2. Untar the source tree:

   ```
   # mount /dev/cdrom /mnt/cdrom
   # cd /usr/src
   # tar -zxvf /mnt/cdrom/linux-2.4.17.tar.gz
   ``

3. Verify that a directory named linux-2.4.17 now exists, and change the symbolic link linux to point to this directory:

   ```
   # cd /usr/src
   # ls
   # ls -ltr
   # rm linux
   # ln -s linux-2.4.17 linux
   # cd linux
   # ls
   ``

4. Do not change the .config file for your new Linux kernel unless you know what you are doing. The patches have already been applied, and the kernel is configured for FLEX-ES use. Save a copy of the .config file:

---

23 These are often known as SMP systems, for Symmetrical MultiProcessor.
# cd /usr/src/linux
# cp .config original.config
# cat README | less

- Begin the process of building a new kernel. Most of these steps produce a large amount of screen output that you are not expected to read:

  # make clean
  # make mrproper
  # make xconfig

- This last step should result in a GUI screen with many buttons.
  - Click the Load configuration from file button (lower right-hand part of screen)
  - Enter /usr/src/linux/original.config as the file name
  - Click the Store configuration to file button and select a new name. We used /usr/src/linux/original.config2. You will probably not need this file, but a backup might be useful sometime.
  - Click the Save and exit button.

- Run the actual kernel-building steps. These steps take some time and produce many screens of output.

  # make dep
  # make bzImage
  # make modules
  # make modules_install

- The make bzImage step should create two files in the root directory of your system, named vmlinuz and System.map. These files must be copied to /boot:

  # ls /verify the files are present
  # cp /vmlinuz /boot/vmlinuz-2.4.17
  # cp /System.map /boot/System.map-2.4.17

- There should be a System.map symbolic link in /boot. Make a new one that points to the new map file. If the current System.map is a file instead of a symbolic link, then rename the file to something like System.map.old.

  # cd /boot
  # ls -l System.map verify old one is present
  # rm System.map remove old symbolic link
  # ln -s System.map-2.4.17 System.map new symbolic link

- Make a ram disk image. This is necessary in order to have the necessary disk drivers available during the boot process. In particular, it is necessary for the RAID and SCSI drivers.

  # cd /boot
  # /sbin/mkinitrd -v initrd-2.4.17smp.img 2.4.17

  This command will look in various standard directories for modules related to 2.4.17 (the second operand) and build a ram disk using the name you specify in the first operand.

- Now you must update lilo to point to the new kernel. (If you use grub instead of lilo, you need to take equivalent actions.)

  # cd /etc
  # vi lilo.conf

  ----our original lilo.conf------
  prompt
  timeout=50
  default=linux
  boot=/dev/sda
  map=/boot/map

  -------our new lilo.conf-------
  prompt
  timeout=50
  default=linux
  boot=/dev/sda
  map=/boot/map
install=/boot/boot.b  install=/boot/boot.b
message=/boot/message  message=/boot/message
linear
linear
image=/boot/vmlinuz-2.4.7.10smp  image=/boot/vmlinuz-2.4.7.10smp
label=linux  label=linux10smp
initrd=/boot/initrd-2.4.7-10smp.img  initrd=/boot/initrd-2.4.7-10smp.img
read-only  read-only
root=/dev/sda5  root=/dev/sda5

image=/boot/vmlinuz-2.4.7-10  image=/boot/vmlinuz-2.4.17
label=linux-up  label=linux
initrd=/boot/initrd-2.4.7-10.img  initrd=/boot/initrd-2.4.17smp.img
read-only  read-only
root=/dev/sda5  root=/dev/sda5

Save and exit from vi. (You could use other text editors; using vi is not a requirement.)
The changes in lilo.conf make your new kernel the default boot kernel.

- Rebuild lilo. You must do this any time you change lilo.conf:
  
  ```
  # /sbin/lilo
  ```

- Shut down your Linux system cleanly and reboot. If the reboot fails, you can revert to your previous kernel (linux10smp) by using the keyboard arrow keys to move to that name in the initial Red Hat display screen, where kernel names are listed on the right-hand side. **Note:** You have only 5 seconds to make this choice, so be ready.

- Log into the system as `flexes` and rebuild the FLEX-ES msgmgr modules:

  ```
  $ cd /usr/flexes/modules
  $ make clean
  $ make all
  ```

- Reboot the system again and you should have a properly working SMP system, ready to run FLEX-ES.

**Alternative method**

Instead of the `make bzlilo` step, you can `make bzImage`. (The “I” is upper case.) Then continue with the `make modules` and so forth. This change will create a compressed kernel file in `/usr/src/linux/arch/i386/boot/bzImage`. Instead of copying vmlinuz to /boot, copy bzImage to /boot/bzImage. For example:

  ```
  # cp /usr/src/linux/arch/i386/boot/bzImage /boot/bzImage-2.4.17smp
  # cd /boot
  # ln -s bzImage-2.4.17smp bzImage
  ```

Continue with the System.map copying, as described above. The entry in lilo.conf or grub.conf should point to /boot/bzImage.

When time permits, we intend to update this procedure to use bzImage and grub as the recommended options.

### 2.4.4 Minor problem

We had a problem with the screen saver function. Our systems would sometimes freeze while a screen saver was running. By default, gnome invokes a screen saver function after a period of no keyboard activity. This function uses a random series of different patterns for the screen. At least one of these patterns causes the system to freeze. (Different patterns caused a similar problem with our ThinkPad.)
One solution is to delete the screen saver function. For the 7.2 release, we did this by selecting the **Start Here** icon (on the gnome desktop). We then selected **Preferences**, **Desktop**, **Screensaver**, and **No screensaver**.
FLEX-ES installation

This chapter provides a brief description of FLEX-ES and then describes the installation process for it. System definitions, resource definitions, and operation is described in the next chapter. The same installation process is used for ThinkPad and Netfinity systems.

FLEX-ES is very easy to install. However, we must stress the importance of having a good Linux installation before you start working with FLEX-ES. There are areas of special concern:

- Is X Windows working correctly? Can you log into the system with the graphics interface? Does gnome (or kde) appear to work correctly? Can you open and close terminal windows with proper responsiveness? Have you disabled the screen saver function? If not, you need to resolve these problems now.
- Is your network interface configured properly? (This includes disabling the interface, if that is your choice.) Is the system attempting to find a DHCP server? Do you want this? Have you defined domain names, but are not connected to a domain name server?
- If you are using an SMP Netfinity, have you applied the kernel patches and built a new kernel?

3.1 FLEX-ES overview

Conceptually, FLEX-ES can be viewed as the following components:

- A S/390 instruction emulator, which might be seen as the heart of the system. It examines each S/390 operation code and emulates that operation, using the instructions of its underlying PC processor.
- A resource manager that controls the interfaces between the emulated S/390 processors and emulated I/O devices and connections.
- Emulators for various S/390 I/O devices.
- A FLEX-ES console for controlling FLEX-ES startup and operation. (This is not related to the z/OS master console.)
- A Terminal Solicitor program that emulates local, channel-attached, non-SNA 3270 terminals. The actual terminals are TN3270 sessions that connect to this program through normal TCP/IP protocols.
- A number of utility programs to help set up and run the FLEX-ES environment.
Figure 3-1 provides a simplified view of FLEX-ES operation. FLEX-ES is simply a process\textsuperscript{24} under Linux. While this illustration should not be taken too literally, it can be used to make a number of basic points about FLEX-ES.

Key points include:

- FLEX-ES is a software product.\textsuperscript{25} All the hardware shown (in Figure 3-1) is standard ThinkPad or Netfinity hardware. (The hardware dongle required by FLEX-ES is not shown in the figure.)
- The FLEX-ES program, running under Linux on the ThinkPad or Netfinity processor(s), emulates a complete S/390 environment.
  - FLEX-ES obtains sufficient Linux virtual memory to emulate the “real memory” for the defined S/390 machine being emulated. While not detailed in the illustration, FLEX-ES can also emulate S/390 expanded memory.
  - FLEX-ES, as part of emulating a S/390, handles S/390 I/O instructions and emulates the S/390 I/O devices as required. In the illustration, various 3390 volumes (containing z/OS and so forth) are contained on the server’s\textsuperscript{26} disks.
- The FLEX-ES license specifies how many server processors may be used (at any one instant) for S/390 emulation. For a ThinkPad/EFS system, this will normally be one processor. For a Netfinity/EFS system, this can be up to three processors.
- The owner can elect to dedicate server processor(s) to FLEX-ES S/390 use. This provides a performance boost. You cannot dedicate all the PC processors to S/390, because other Linux functions need access to a processor. If processors are not

\textsuperscript{24} It is actually a number of interrelated processes and threads, but this detail can be ignored at the conceptual level.
\textsuperscript{25} The optional S/390 channel adapters available for use with FLEX-ES are ignored in this discussion.
\textsuperscript{26} Remember that we use the word server to mean either a ThinkPad or Netfinity system.
dedicated, then the normal Linux dispatching function will dictate which processors are being used by FLEX-ES at any instant. This number will never exceed the number permitted in the FLEX-ES license. A ThinkPad/EFS system has only a single processor, and it cannot be dedicated for S/390 emulation.

You can emulate a S/390 with a number of CPUs. This number cannot exceed the number of server processors licensed for S/390 emulation. With the ThinkPad/EFS system we are describing, this means that only a single-processor S/390 can be emulated. With a Netfinity system (and the proper hardware), a S/390 with up to three CPUs can be emulated.

The emulated S/390 can use LAN adapter(s) on the ThinkPad or Netfinity. z/OS TCP/IP can run this way. A single ThinkPad LAN adapter can be used both by z/OS TCP/IP and by Linux TCP/IP. This differs from FLEX-ES under UnixWare, where a LAN adapter may not be shared by a S/390 TCP/IP and UnixWare.27

The FLEX-ES Terminal Solicitor is a Linux process. It provides a TN3270 server function and listens (by default) on port 24 of Linux's TCP/IP. An external user can connect a TN3270 client to the Terminal Solicitor. The Terminal Solicitor and the FLEX-ES emulation process will then transform the TN3270 protocol so that it appears to be a local, non-SNA 3270 to the emulated S/390. This is the normal way to connect MVS consoles, TSO terminals, CICS terminals, and so forth. There is no particular limit to the number of terminals that can be connected this way.28 Each one must have appropriate VTAM definitions for a local 3270, of course.

If a LAN adapter is connected to z/OS TCP/IP, then TSO terminals may connect this way. If z/OS TCP/IP and UNIX System Services are configured for it, ASCII telnet sessions can also use this connection. The Terminal Solicitor is not involved in these connections. The two LAN interfaces (Linux and z/OS) would have separate IP addresses, of course, even though they may share a common LAN adapter.

The illustration shows a single emulated S/390. It is possible to emulate several S/390s at the same time.29 The server processor(s) licensed for S/390 emulation will be dispatched among the several emulated S/390s. Only one Terminal Solicitor would be used; it can be shared by all emulated S/390s.

The multiple emulated S/390s are known as multiple instances of FLEX-ES S/390 emulation. Multiple S/390 instances of z/OS are probably not very practical with a ThinkPad/EFS system, due to limited ThinkPad memory and due to there being a single ThinkPad processor that must be shared among all the S/390 instances plus Linux. However, multiple VSE/ESA or VM/ESA systems might be practical on a ThinkPad/EFS machine. Multiple S/390 instances are practical on larger Netfinity systems.

While it is not apparent in the illustration, the amount of real server memory available is a key performance factor. The performance of an emulated S/390 degrades quickly if Linux performs much paging (especially if the paging is triggered by FLEX-ES processes, or if the memory being paged is used by FLEX-ES to emulate S/390 real memory). The recommended paging level for Linux is zero, once steady-state S/390 emulation is running, and should seldom rise to more than one page per second. z/OS paging (in the emulated S/390) might be much higher than this, with no ill effects.

Do not confuse Linux paging with z/OS paging. The amount of real server memory available is a key factor in setting the size of the emulated S/390 machine memory and in deciding whether to use multiple S/390 instances.

27 It also differs from P/390-based machines and the MP3000, none of which permit sharing of LAN adapters by multiple TCP/IP stacks.
28 There is an overall FLEX-ES limitation of 2048 emulated devices, but typical EFS systems are unlikely to approach this limit.
29 A different FLEX-ES license is not required for this. The FLEX-ES license indicates how many server processors may be used for S/390 operation, but does not limit how many instances of S/390s you can run.
The terminal shown in the figure is usually the server display and keyboard. We assume you use the X Windows interface to Linux. In this mode, several windows can be open. Some of these might be x3270 sessions (TN3270 clients) connected to z/OS through the Terminal Solicitor; one of these sessions might be the MVS master console. Another window might have FLEX-ES command line interface (CLI) prompts used to control FLEX-ES operation. CLI commands are used to IPL a S/390 operating system, for example.

3.2 FLEX-ES installation

FLEX-ES can be delivered a number of ways:

- FTP over the Internet
- CD-ROM
- Diskettes
- Preinstalled by a business partner

The FLEX-ES package is not large (about 3.5 MB), and an FTP download is easy, even over a typical dial-up line. In addition to the FLEX-ES code, you need a FLEX-ES license key file and dongle. The license key file is a few hundred bytes and will normally be shipped with your system. For our examples, we assume that you have FLEX-ES on a CD.

If you download a new version using ftp, place the downloaded file in your /tmp directory and follow approximately the same instructions.

The FLEX-ES file on the CD (or the downloaded file) are usually in tar format. You want to untar the file and place the results in the /tmp directory:

```bash
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ls -al
flexes-6.1.13.i386.rpm.tar (or a name similar to this)
# cd /tmp
# tar xvf /mnt/cdrom/flexes-6.1.13.i386.rpm.tar (if working from CD)
```

If you are downloading a new FLEX-ES release, simply download the tar file into /tmp and the rest of the instructions here should apply. Your exact file names may be different, depending on the FLEX-ES release you are using. These commands should produce several files in /tmp:

```bash
# ls
flexes-6.1-13.i386.rpm (primary FLEX-ES modules)
msgmgr-6.1-13.i386.rpm (FLEX-ES kernel communication)
ftlib-6.1-13.i386.rpm (source for creating programs using FakeTape)
(other packages may be present for parallel channels, ESCON channels, ICA ports)
```

Again, your exact file names may differ but the general names will be as shown here.

```bash
# cd /tmp
# rpm -i flexes-6.1-13.i386.rpm
/var/tmp/rpm -tmp.66303: /usr/bin/expect: No such file.....
(This message means the installation process failed to set a password for user flexes)
(You can set the password manually)
# rpm -i msgmgr-6.1-13.i386.rpm
gcc .........
lc ..........
(ignore a few messages. Takes a few seconds.)
# rpm -i ftlib-6.1-13.i386.rpm
```
These three `rpm` commands perform the basic FLEX-ES installation.

A userid and a groupid named `flexes` are created automatically during FLEX-ES installation. All normal FLEX-ES operation is done under this userid. The FLEX-ES installation process attempts to assign the initial password “abcdef1” for userid `flexes`. If the installation options you selected while installing Linux caused `/usr/bin/expect` to be installed, then the FLEX-ES installation process will automatically set password `abcdef1` for userid `flexes`. Otherwise, you should set a password. (Of course, you do not need to use `abcdef1` as the password.)

```
# passwd flexes
abcdef1 (after prompt to enter password)
BAD PASSWORD: IT IS TOO SIMPLISTIC/SYSTEMATIC
abcdef1 (after second prompt to enter password)
```

The password is accepted, regardless of the message.

FLEX-ES installation creates `/usr/flexes` and makes this the home directory for userid `flexes`.

We then performed minor steps that we think makes operation easier:

```
# cd /usr/flexes
# mkdir rundir
# vi /root/.bash_profile
USERNAME="root"
PATH=$PATH:/usr/flexes/bin
export USERNAME BASH_ENV PATH
```

Adding `/usr/flexes/bin` (where all the FLEX-ES executables are located) to PATH makes operation easier. Redbooks dealing with FLEX-ES assume that you create directory `/usr/flexes/rundir` and store your FLEX-ES operational files in this directory. This is not required, but we assume this directory location in the remainder of this redbook.

### Installing the FLEX-ES license key and dongle

Connect the dongle to the USB port. (We have always done this before booting the system and we do not know how adaptable Linux may be if you connect or disconnect the dongle on a running system.) Issue the following command:

```
# cat /proc/bus/usb/usbd/devices
```

On our system, the results were as follows:

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#= 1 Spd=12 MxCh=2
B:  Alloc = 0/900 us ( %), #Int= 0, #Iso= 0
D:  Ver= 1.00 Cls=09( hub ) Sub=00 Prot=00 MxPS= 8 #Cfgs= 1
P:  Vendor=0000 ProdID=0000 Rev=0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=1860
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr= 0mA
I:  If##= 0 Alt= 0 #EPS= 1 Cls=09( hub ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03( Int. ) MxPS= 8 lv1=255ms
T:  Bus=01 Lev=01 Prnt=01 Port=00 Cnt=01 Dev#= 2 Spd=12 MxCh= 0
D:  Ver= 1.10 Cls=00 (> ifc ) Sub=00 Prot=00 MxPS=16 #Cfgs= 1
P:  Vendor=1404 ProdID=cccc Rev= 1.00
S:  Manufacturer=Fundamental Software Inc.
S:  Product=FSI Dongle
C:# #Ifs= 1 Cfg#= 1 Atr=80 MxPwr=100mA
I:  If##= 1 Alt= 0 #EPS= 2 Cls=ff(vend.) Sub=ff Prot=ff Driver=serial
E:  Ad=81(I) Atr=02(Bulk) MxPS= 16 lv1= 0ms
E:  Ad=01(0) Atr=02(Bulk) MxPS= 16 lv1= 0ms
```

30 The `/usr/flexes/.bash_profile` already contains `PATH=$PATH:$HOME/bin` where `$HOME` is `/usr/flexes`. 

Chapter 3. FLEX-ES installation  35
If your output is something like this and includes lines about the FSI dongle, you can assume
the dongle is being recognized. The following command is needed to make the dongle
available to FLEX-ES software:

```
# /sbin/modprobe usbserial vendor=0x1404 product=0xcddc
```

Your system may already include this command. Look in file /etc/init.d/flexes and see if this
command is present in the file. If it is not, add it. In our early system, we added it just after the
“loadable drivers” comments:

```
....
#
# Install our loadable drivers
#
/sbin/modprobe usbserial vendor=0x1404 product=0xcddc  **add this line**
#
if [ -f /usr/flexes/modules/msgmgr.o ]; then
....
....
```

You need a FLEX-ES license key, which is a Linux file of a few hundred bytes. The key must
match the dongle (via an encoded serial number in the key and dongle). Copy the license key
file to /var/adm/flexes/.flexeslicense. (Note that the target file name begins with a period.) If
the license key is provided on diskette, you will need to copy it from there. For example:

```
(shut down Linux)
(remove the CD-ROM drive from the Ultrabay of the ThinkPad)
(insert the diskette drive in the Ultrabay)
(boot Linux and log in as root)
# mdir a:
(if the diskette is in DOS format, you should see the file with your key)
# mcopy a:your.key /var/adm/flexes/.flexeslicense
(done)
(if the diskette is not in DOS format, try the following)
# mount /dev/fd0 /mnt/floppy
# ls -al /mnt/floppy
(if the key file is displayed, copy it with the following command)
# cp /mnt/floppy/yourlicensekeyname.key /var/adm/flexes/.flexeslicense
```

ThinkPad models are updated and changed frequently. The exact method of changing from
CD-ROM use to diskette use depends on your ThinkPad model. In some models, you can
have both a CD-ROM drive and a diskette drive active at the same time. In this case, there is
no need to shut down Linux to swap drives.

If you have multiple FLEX-ES licenses, be certain to identify the correct license with the
correct dongle. There is no visible way to associate a license file with a specific dongle. Each
dongle has a serial number on it, but you will not see this serial number in the license file. If
you have multiple EFS systems, you need to devise a way to keep track of which dongle goes
with which license file.

The Linux kernel provided with the Red Hat 7.2 system should recognize and use the USB
dongle, as described. However, if you build your own kernel, it will probably not use the
dongle correctly. If you build a kernel, configure the USB Serial Converter as a module and
say yes to the prompt about USB Generic Serial Driver support.

The license key file is an ASCII file, with the contents masked (scrambled). *Do not attempt to edit it.* Also, the current license key files are sensitive to the FLEX-ES release level. Do not
attempt to use your license key file with the wrong FLEX-ES release.
3.2.1 X3270

This step may not be necessary. If /usr/bin/X11/x3270 does not exist in your system, you can skip this step. We found it necessary for Red Hat 7.1 and an earlier release of FLEX-ES. It was not necessary for Red Hat 7.2 and the current release of FLEX-ES. We do not know if Red Hat removed the x3270 module or if the current FLEX-ES installation process deleted it.

A version of x3270 (a TN3270e client) is distributed with FLEX-ES. A different version of x3270 is distributed with Red Hat Linux. You should use the version distributed with FLEX-ES. Since we placed the FLEX-ES libraries after the Linux libraries in our PATH environment, the system will find the Linux x3270 first. You can correct this in a number of ways. We did this:

```
# cd /usr/bin/X11
# mv x3270 x3270xxx
```

(rename it to something obscure)

After doing this, the FLEX-ES distribution of x3270 will be found using our PATH. (You could accomplish the same thing by placing /usr/flexes/bin first in the PATH, or you could simply delete the X11 version of x3270.)

3.2.2 Mount points for ThinkPad

We created mount point /s390 when we installed Linux. It is for a large file system that occupies most of our first hard disk. We intend to use a second hard disk (in the Ultrabay slot of the ThinkPad) and we can create mount points for it now. (We will initialize and mount the second hard disk later.) All these mount points should be owned by userid flexes.

```
# mkdir /s391
# chmod flexes:flexes /s390
# chmod flexes:flexes /s391
# chmod flexes:flexes /holding
```

The names s390, s391, and holding are arbitrary, of course, but they express the purpose of the file systems.

3.2.3 Mount points for Netfinity

The Netfinity we describe has a single RAID array. We created several partitions in the array during Linux installation:

- / (root) for Linux
- /boot for Linux
- A swap file for Linux
- /s390 for all our emulated volumes
- /holding for a work area.

These were created when we first partitioned the RAID “disk” and no new mount points are needed. We should verify that /s390 and /holding are owned by flexes:

```
# chown flexes:flexes /s390
# chown flexes:flexes /holding
```
In principle, you can install z/OS on your ThinkPad/EFS or Netfinity/EFS system using any of the distribution media and logical packaging available from IBM. In practice, most of the IBM S/390 Partners in Development (PID) members use prepackaged systems distributed on CD-ROMs. The only recent package available is the z/OS AD CD-ROM system. This is available only to PID members who obtained systems through the PID organization.

Older versions of VM/ESA and VSE/ESA systems, on CD-ROMs, were produced by the PID organization. These are no longer available, and no replacements were apparent at the time of writing. EFS systems are not limited to z/OS, of course. If you can find a way to load it, you could use VM, VSE, and Linux systems. (The “other” current operating system, TPF, has not been tried on an EFS system, as far as we know.)

“Stand-alone z/OS restore” on page 81 briefly discusses z/OS installation using backup tapes created on another system. This is the most likely installation method for general EFS customers who are not PID members. The remainder of this chapter discusses the installation of AD CD-ROM systems.

### 4.1 AD systems

We chose a z/OS AD CD-ROM system (z/OS release 1.2) for this redbook. Not all readers are familiar with these “AD” systems; we briefly explain them here. A z/OS AD (Application Development) system is a prepackaged z/OS, with a number of priced features and additional program products included. Considerable customization has already been done, making the system immediately usable for many functions.

---

31 This statement assumes you have a tape drive on your ThinkPad/EFS that is compatible with the IBM-provided media, of course.

32 There are AD systems available for VM/ESA and VSE/ESA also. The discussion in this chapter is about z/OS, so we limit this discussion to the z/OS AD systems.
Why did we use it for our EFS projects? We used it primarily because it provides a very easy way to install a useful z/OS system. We could have built a system starting with a ServerPac, in the same way most z/OS customers build their systems. However, this requires considerably more time and effort and would have detracted from the time spent working with EFS elements. It would also require a hardware system with tape drives compatible with the ServerPac media, and our ThinkPad and Netfinity did not meet this requirement.

In general, a z/OS AD system is a rather straightforward implementation of z/OS and contains no magic components or clever setups. The experience and results of using it on an EFS system should be about the same as using any other straightforward z/OS implementation.

The AD CD-ROM systems, as the name implies, are distributed on CD-ROMs. This aspect is not common to other z/OS packaging, but does not affect the characteristics of the system once it is installed. The CD-ROMs are not seen by the S/390; they are processed by Linux programs.

### 4.1.1 z/OS on CD-ROM

The AD CD-ROM versions are based on the same versions available (on tape) through the standard IBM software distribution processes. There are no modifications involved due to the use of CD-ROMs for distribution. However, the AD versions distributed on CD-ROM are considerably more customized than the standard IBM software distribution versions.

In general, the AD versions (once installed on the S/390 system disks) are immediately ready for use. A set of userids is provided, for example. Minor additional customization (such as setting IP addresses) is required, but very little work is required compared to, for example, a ServerPac distribution of z/OS. The penalty for this immediate usability is that many configuration and customization decisions have been made by the group building the AD systems. In general, the resulting systems are quite suitable for a small development organization, but would probably not be suitable for a large, highly-structured production installation. Since the target for the AD CD-ROM systems is smaller development organizations and hands-on education, the AD CD-ROM systems have been very well received.

#### Basic CD-ROM formats

The fundamental format of the CD-ROMs is PC-compatible. That is, any DOS, OS/2, Windows, PC UNIX, and PC Linux can recognize the files and directories on the CD-ROMs. The CD-ROMs typically contain README files (in ASCII), P/390 DEVMAP files (binary), an OS/2 UNZIP program (binary), possibly an AIX UNZIP version (binary), and one or more files containing the z/OS volumes (binary). From a PC viewpoint, these z/OS volumes are binary files and are usually very large files.

The z/OS volumes (on the CD-ROM) are ZIPed files in AWSCKD format. AWSCKD is a P/390 device manager program that emulates 3380 and 3390 devices. In general, a complete 3390 volume is in one or two PC files. All the AD releases of OS/390 and z/OS are in this format.

The AWSCKD format is, in essence, a complete image of a S/390 disk volume. Within an AWSCKD file CKD tracks, cylinders, R0s, and so forth are defined. There will be a standard label, a VTOC, probably a VTOC index, and whatever data sets appear on that volume in an z/OS context. The data is in S/390 format. Text contained on a S/390 volume is EBCDIC and executables are S/390 binary files suitable for execution by z/OS. To a PC program, an AWSCKD file is simply a large binary file that is not useful to typical PC programs.
AWSCKD files on the CD-ROM are in ZIP format simply to save space. It is usually possible to ZIP an AWSCKD 3390-3 (2.8 GB) so that it fits on a CD-ROM (about 600 MB). There is no basic requirement that ZIP files be used and, in rare cases, disk images might not be zipped.

The AWSCKD format was developed for P/390s, where the underlying operating system used to emulate CKD drives is OS/2. OS/2 is a 32-bit operating system and has the usual restriction that a single file cannot be larger than 2 GB. An AWSCKD-emulated 3390-3 requires more than 2 GB, and is split into two AWSCKD files, the first is 2 GB and the second is about .8 GB. The two files are usually placed in a single ZIP file.

**FLEX-ES formats**

FLEX-ES can handle the CD-ROM format discussed above. FLEX-ES emulates CKD drives, but the internal format (in the Linux files used for emulation) is different than the AWSCKD format. FLEX-ES provides a utility (`ckdconvaws`) to convert AWSCKD format to the FLEX-ES format for emulated CKD drives.

### 4.1.2 z/OS device configuration

The general structure of recent OS/390 and z/OS AD CD-ROM releases is explained in considerable detail in other documents. PID Web sites have links to these documents. The contents of the various 3390 volumes and the addresses assumed by the systems (such as A80 for the IPL volume) are explained in these documents.

For the z/OS R1.2 AD system, the first four volumes (first four CD-ROMs) contain an IPLable system on 3390-3 volumes, and we initially installed only these volumes. We also created a 3390-1 volume for work space.

We decided our initial z/OS system (using the AD CD-ROM) would use the devices listed here. This represents a basic but usable system.

<table>
<thead>
<tr>
<th>Address</th>
<th>Device</th>
<th>VOLSER</th>
<th>Linux Name</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>A80</td>
<td>3390-3</td>
<td>Z2RES1</td>
<td>/s390/Z2RES1</td>
<td>IPL volume; z/OS libraries</td>
</tr>
<tr>
<td>A81</td>
<td>3390-3</td>
<td>Z2RES2</td>
<td>/s390/Z2RES2</td>
<td>More z/OS basic libraries</td>
</tr>
<tr>
<td>A82</td>
<td>3390-3</td>
<td>OS39M1</td>
<td>/s390/OS39M1</td>
<td>Paging, parmlib, catalogs, etc</td>
</tr>
<tr>
<td>A87</td>
<td>3390-3</td>
<td>Z2USS1</td>
<td>/s390/Z2USS1</td>
<td>System HFS data sets</td>
</tr>
<tr>
<td>A91</td>
<td>3390-1</td>
<td>WORK02</td>
<td>/s390/WORK02</td>
<td>Volume added locally</td>
</tr>
<tr>
<td>560</td>
<td>3480</td>
<td></td>
<td></td>
<td>Emulate a 3480, if needed</td>
</tr>
<tr>
<td>700</td>
<td>3270</td>
<td></td>
<td>mstcon</td>
<td>NIP &amp; z/OS master console</td>
</tr>
<tr>
<td>701-70F</td>
<td>3270</td>
<td></td>
<td>L701 - L70F</td>
<td>Local non-SNA VTAM terminals</td>
</tr>
<tr>
<td>E20, E21</td>
<td>CTC</td>
<td></td>
<td>eth0</td>
<td>Used for LAN TCP/IP interface</td>
</tr>
</tbody>
</table>

The AD system can use a much larger set of addresses and devices than shown here. Some of the addresses are arbitrarily chosen from the larger set provided with this AD system. The AD volumes containing the DLIBs, DB2, CICS, and IMS are not included in this list; they are not necessary for basic operation and we decided to not install them for our initial EFS step. We discuss their addition later.

---

33 The two files contain the characters _1 and _2 as the last two characters of each file name. P/390 utilities recognize that the second file is a continuation of the first. Unlike FLEX-ES, the P/390 does not emulate 3390-9 drives. If it did, it would use four 2 GB files plus a 1 GB file to equal the 9 GB contained on a 3390-9 drive.
The Linux file names shown in the table are not part of the AD system. Rather, they are the file names we decided to use when installing these emulated volumes on our EFS system. They are shown here for completeness. An important factor in our setup is to have an IPLable system on the primary ThinkPad hard disk. Although we will later install a second ThinkPad hard disk, we want to be able to IPL z/OS without the second disk installed. (For a Netfinity installation, this is not a concern, of course.)

The Linux file names for emulated S/390 volumes are simply Linux file names. These files can be renamed with the `mv` command, copied to other locations with `cp`, and treated as normal (but large) Linux files. There is no requirement to have the file name the same as the volser of the emulated S/390 volume contained in the file. Most users do assign a file name that matches the volser, but this is for convenience.

At the system level, we intend to define an emulated S/390 system with 256 MB central storage and 64 MB expanded storage. The expanded storage size is completely arbitrary. We wanted our total system definition to be small enough to avoid Linux paging. We could have defined larger central storage, or no expanded storage, or a different combination of storage sizes.

### 4.1.3 Future directions

The installation directions in this chapter assume that the CD contents (after unzipping) are in the AWSCKD formats, as described above. This may not be the case for future releases. The AWSCKD format is the native format used by P/390-based systems and by MP3000 systems (for emulated drives). These systems do not offer 64-bit operation. Software releases that can be used only through 64-bit operation have no requirement for native AWSCKD format. It could still be used, via conversion programs such as `ckdconvaws`, but other formats may be more appropriate. One option is a compressed tar format, provided the compression is at least as effective as the OS/2 `zip` program.

At the time of writing, all PID distributions continue to be in AWSCKD format. This discussion is simply to warn you that this format might not be used indefinitely.

Why not use DVDs instead of CDs for distribution? Fewer DVDs would be required. This is a possibility sometime in the future, but probably not soon. Few of the systems that use the AD CD-ROM distributions have DVD drives. This will probably change over time, as combined CD/DVD units become more common.

### 4.1.4 An unzip program

A PC-compatible unzip program is included in the Red Hat distributions. The command-line options are slightly different than those used with DOS-oriented unzip programs.

### 4.2 Installation tasks

We found that the easiest way to do the following process is to open two command-line windows on the Linux desktop:

- One window for the `unzip` and `ckdconvaws` commands, as userid `flexes`
- One window for `mount` commands and other utility commands, with an `su` to `root`

---

34 This means that the IODF distributed with the AD system contains a large number of defined devices and addresses. We selected a subset of these already-defined devices and addresses.
Having separate windows for the two major commands is convenient because root authority is needed for `mount` and `umount`, while the other functions should be under userid `flexes`. If you do not perform most of your FLEX-ES functions under userid `flexes`, you will have problems later. You can avoid later problems by logging into the system as `flexes` and performing all functions as `flexes`, except for the few cases where you need to operate as root.

Please go back and read the last paragraph again! One of the most common installation problems we have seen is due to a failure to log into the system as `flexes`. (Userid `flexes` must own most of the files used by FLEX-ES; this is the fundamental reason for working as `flexes`.)

**Mount CD**
Mount the first AD CD-ROM:

```
# mount /dev/cdrom /mnt/cdrom (issue from root)
# ls /mnt/cdrom/zos12
  z2res1.zip
```

(This `mount` command may be unnecessary because Linux may sense that a CD has been loaded and automatically performs the mount.\(^{35}\) If this happens, simply close the window that Linux opened as part of the automatic mount operation. You will still need to issue a `umount` command when finished with each CD.) The exact contents of the CD may vary from release to release, but you will see a large zip file. This is what will be installed. This file is normally in a second directory on the CD, and not in the root directory of the CD.

**Unzip and convert a volume**
The output of the `unzip` command can be piped directly to the `ckdconvaws` command:

```
$ unzip -p /mnt/cdrom/zos12/z2res1.zip | ckdconvaws -r - /s390/Z2RES1 3390-3
  FSIDC146 Max head=14, cyl=3343, blks=57
  FSIDC180 cylinder nnn Completed in mmm milliseconds
  FSIDC190 CKD conversion completed (3339 cyls copied, 0 cyl ignored)
```

The `ckdconvaws` command converts the AWSCKD format (used on the AD systems) to the FLEX-ES format, and writes to the Linux file indicated.\(^ {36}\) The volume being unzipped is a 3390-3 (and we must tell the `ckdconvaws` program this). The second operand, `/s390/Z2RES1`, is the target location for the Linux file that will emulate the S/390 volume.

Your disk layout may be different, and you need to adjust this path and name as required for your system. We used upper case names for the Linux files holding emulated S/390 volumes, but this is not required. We also made the Linux file names, such as `Z2RES1`, the same as the volser of the S/390 volume being emulated by that file. This naming convention is not required, but it helps us keep track of the purpose of each Linux file.

The `nnn` and `mmm` numbers shown in the output of the `ckdconvaws` command will increment as the program runs. When the volume is finished, use the root window to issue the commands:

```
# cd /
# umount /mnt/cdrom
```

(in the root window)

Take care when entering the `ckdconvaws` command. Be certain to leave blanks (spaces) before and after the `-r` and `-` operands.

Command speed may appear erratic, with fast cylinder copies followed by long pauses. This is due to Linux buffering large amounts of disk I/O in memory, and then flushing it to disk. (This effect is not seen if you are using raw disk interfaces.)

\(^ {35}\) Linux sensed the CD load automatically on our ThinkPad, but not on our Netfinity.

\(^ {36}\) The `-r` option indicates that `ckdconvaws` should create the output file, if it does not already exist.
We can pipe directly from the `unzip` program to the `ckdconvaws` program only if the emulated volume will be held in a single Linux file, as we are doing here. 3390-3 volumes are held in two files in the original AWSCKD format. Both these files are usually placed in the same zip file. The `ckdconvaws` program handles this transition automatically.

**Load the other CDs**

Repeat the same process (mount, unzip/convert, umount) for the other CDs that are to be loaded. For the z/OS 1.2 system, we used the following:

```
CD     CD File.................       Linux File      Disk Model
2     /mnt/cdrom/zos12/z2res2.zip    /s390/Z2RES2      3390-3
3     /mnt/cdrom/zos12/os39M1.zip    /s390/OS39M1      3390-3
4     /mnt/cdrom/zos12/z2uss1.zip    /s390/Z2USS1      3390-3
```

4.2.1 Create a work volume

This step is not required to use the AD system, but we like to have at least one work volume available. The FLEX-ES command:

```
$ ckdfmt -n -r /s390/WORK02 3390-1
```

will create and format an emulated 3390-1 volume. The -r flag in the command indicates that the file should be created if it does not already exist. The file name, WORK02, is arbitrary. This command does not format the volume for z/OS. After IPLing z/OS, we need to run an ICKDSF job to create a label and VTOC on the volume. We intended to use the volser WORK02, and we used the same name for the Linux file name. After IPLing, we also modified the VATLST00 member in the z/OS PARMLIB to mount WORK02 as a STORAGE volume.

4.2.2 Loading the remaining volumes

We suggest you skip this section when you are first installing your system, and return to it after you are comfortable running your emulated S/390 (as described in the next chapter).

There were 13 volumes distributed with the z/OS 1.2 AD system we used. Only four volumes are needed to IPL the system and use TSO. The remaining volumes are:

<table>
<thead>
<tr>
<th>CD#</th>
<th>volser</th>
<th>3390 model</th>
<th>assumed address</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Z20IS1</td>
<td>3390-3</td>
<td>A85</td>
<td>system DLIBs</td>
</tr>
<tr>
<td>6</td>
<td>Z20IS2</td>
<td>3390-3</td>
<td>A86</td>
<td>system DLIBs</td>
</tr>
<tr>
<td>7</td>
<td>Z20IS3</td>
<td>3390-3</td>
<td>A88</td>
<td>system DLIBs</td>
</tr>
<tr>
<td>8</td>
<td>Z2DB21</td>
<td>3390-3</td>
<td>A83</td>
<td>DB2 (including DLIBs)</td>
</tr>
<tr>
<td>9</td>
<td>Z2CIC1</td>
<td>3390-2</td>
<td>A84</td>
<td>CICS (including DLIBs)</td>
</tr>
<tr>
<td>10</td>
<td>Z2IMS1</td>
<td>3390-2</td>
<td>A89</td>
<td>IMS (including DLIBs)</td>
</tr>
<tr>
<td>11</td>
<td>Z2WAS1</td>
<td>3390-3</td>
<td>A8A</td>
<td>WebSphere</td>
</tr>
<tr>
<td>12</td>
<td>Z2WAS2</td>
<td>3390-3</td>
<td>A8B</td>
<td>WebSphere</td>
</tr>
<tr>
<td>13</td>
<td>SARES1</td>
<td>3390-3</td>
<td>A8C</td>
<td>Stand-alone system</td>
</tr>
</tbody>
</table>

You may elect to install any or all of these additional volumes. If you want to install system PTFs, for example, you will almost certainly need the three DLIB volumes. The SARES1 volume is also a z/OS 1.2 system, but in a very minimal form that requires only one volume. It is meant as an emergency recovery system or as a platform for customizing the main volumes while they are not active.

Later AD releases may have different volumes (and will certainly have different volsers involved), but the general principles should remain the same.
The process for loading the remaining volumes is, in principle, exactly the same as for loading the first four volumes. That is, you mount the CD, unzip and ckdconvaws it, and umount the CD. For a Netfinity system, or for a ThinkPad with an internal hard disk large enough to hold all the volumes you need, you would simply continue the previous loading process for whichever additional volumes you wish to load.

For a ThinkPad with two hard disks and where the Ultrabay must be shared between a CD-ROM drive and the second hard disk, you will need a few more staging steps:

- Prepare the second hard disk, as described in “Using a second Linux hard disk” on page 63. You may need to shut down Linux, insert the second hard disk into the Ultrabay, and reboot Linux.
- Install the CD-ROM drive. This may require shutting down and rebooting Linux. (This was required on our T20 ThinkPad.)
- Determine how much free space you have on the internal hard disk. A df -h command can be used. Look in the /s390 file system, if you used our suggested disk layout. Determine how many emulated volumes can be loaded in this space. Our T20 had 10 GB free in /s390 and this will hold three 3390-3 volumes.
- Install (mount, unzip/ckdconvaws, umount) volumes from CDs to fill the free space.
- Remove the CD-ROM drive and install the second hard disk. This may involve shutting down and rebooting Linux.
- Copy the volumes from the internal hard disk to the second hard disk; for example:
  
  $ cp /s390/Z2DIS1 /s391/Z2DIS1
  $ cp /s390/Z2DIS2 /s391/Z2DIS2

  This example assumes you are using our naming conventions. These copy commands will take several minutes for each file.
- Delete the volumes from the internal hard disk. For example,
  
  $ rm /s390/Z2DIS1
  $ rm /s390/Z2DIS2

- Repeat the process with more volumes from the CDs, if necessary.
- After all the additional volumes are on the second hard disk, use an ls -al /s391 command to verify that they are owned by userid flexes.

Creating more work volumes

We created two more work volumes and placed these on the second hard disk of the ThinkPad. (Netfinity users can place them with the other S/390 volumes, of course.) Your requirements for work volumes may be quite different. We used the commands:

  $ ckdfmt -n -r /s391/WORK01 3390-1
  $ ckdfmt -n -r /s391/WORK03 3390-1

Again, it will be necessary to run ICKDFS jobs to make these volumes ready for use by z/OS. You can do this after IPLing z/OS the next time. We placed these additional work volumes in VATLST00 as STORAGE volumes.

4.3 File ownership

All of the files containing emulated S/390 volumes must be owned by userid flexes and groupid flexes. You can verify this, as follows:

# ls -al /s390 (check the owner and group names)
# ls -al /s391 (if you have a second disk installed)
(if you need to change ownership, do the following)
# su (change to root, if not already root)
# chown flexes:flexes /s390/Z1RES1 (or whatever file needs to be changed)
The “flexes:flexes” parameter format causes both the owner and the group name to be changed.

4.4 Minor problems

While working with various AD releases, we found two unusual situations.

Files unzip in wrong order

Unzipping one of the 3390-3 volumes (Z2RES1) for the z/OS 1.2 AD system, for example, produces the two files in reverse order. In OS/2 terms, it unzips Z2RES1_2.A80 and then Z2RES1_1.A80. For an OS/2-based system this does not matter, since these are really two separate files (that together constitute a single 3390-3 volume). For older FLEX-ES releases, these files must be fed to `ckdconvaws in the correct order`; and this involves using intermediate files. In the current FLEX-ES releases (6.1 or later), `ckdconvaws` handles the reverse-order files automatically.

Volume in two separate zip files

Many of the AD volumes are 3390-3 volumes. An emulated 3390-3, on the OS/2-based emulation systems used to build the AD releases, requires two files. (This is due to the 2 GB maximum file size on OS/2.) The two files are normally placed in the same zip file on a CD-ROM. The `ckdconvaws` program, when used as documented in this redbook, automatically combines the two files (contained in the single zip file) and produces the single emulated 3390-3 file used by FLEX-ES.

In rare cases, the zip file containing the two OS/2 files is too large to fit on a CD-ROM. (This happened with the OS39DA volume for the OS/390 R10 AD system.) When this happens, the two OS/2 files are zipped separately and placed on different CDs. (For R10, the larger file was placed on CD #5, by itself, and the smaller file was placed on CD #3, which also held the OS39M1 volume.) For OS/2-based systems (such as P/390s and emulated I/O on the MP3000), this is not a problem because the two files are unzipped and restored as two separate files.

In the latest FLEX-ES releases (6.1 or later), the `ckdconvaws` command can handle the two zip files separately and produce the correct results. That is, you would run `ckdconvaws` twice (once for each zip file) and point to the same output file name for your S/390 emulated volume. This is a considerable improvement over earlier releases, where a moderately complex procedure was needed to handle this situation.
Chapter 5. FLEX-ES operation

Installing and running an EFS system requires a number of steps, some of which have already been described:

- Install the system hardware.
- Install Linux as the server operating system.
- Decide on your disk configuration.
- Install FLEX-ES.
- Load a S/390 operating system.

We have completed the steps listed above. The following steps are required to run a S/390 instance:

- Create a FLEX-ES definition file.\(^\text{37}\)
- Compile the definition file, creating rescf and syscf files.
- Start resadm, the resource manager, pointing to a rescf file.
- Start flexes, the S/390 emulator, pointing to a syscf file.
- Start the flexescli program, which provides a command line interface for controlling a FLEX-ES instance.
- Start at least one TN3270e session (for the MVS master console).
- Issue an ipl command through the flexescli interface.
- Operate z/OS as usual.
- Shut down the system when finished.

These steps are described in this chapter. In general, most of the setup described here should be done under the flexes userid. By convention, all the FLEX-ES configuration files, resource definitions, shell scripts, and so forth, are kept in /usr/flexes/rundir. This is not required, but it is common practice and the following examples use this directory.

\(^\text{37}\) This can be done in two separate files: a system definition file and a resource definition file. We elected to combine these into a single file.
5.0.1 Linux desktop windows

When using our EFS system, we routinely log into the system as flexes and open two terminal windows on the Linux desktop. We change to the /usr/flexes/rundir directory in both windows. In one window we su to root. We use this window for starting and stopping the FLEX-ES resource manager (resadm). The other window is used for running a shell script and is then used to enter flexes commands.

The shell script we use automatically starts two x3270 sessions. We then have a total of four terminal sessions on the desktop:

- One (su to root) for resadm commands,
- One for flexes commands.
- One x3270 session for the MVS console.
- One x3270 session for TSO.

There are many other ways you can arrange your desktop and FLEX-ES operation. However, we suggest you follow our pattern for your initial usage.

5.1 FLEX-ES system and resource definitions

Before the newly installed/restored z/OS system can be used, we must define the FLEX-ES system and resource elements required. “Basic definitions for a single HDD” on page 97 shows the file that we used for our initial system with four system volumes and a work volume. We placed this definition file in /usr/flexes/rundir/defR12A. (File name defR12A is completely arbitrary.) The content, format, and rules for creating these definition files are briefly explained in the Appendix. We specify the name of the definition file as an argument for the FLEX-ES configuration compiler:

```
$ cd /usr/flexes/rundir
$ cfcomp defR12A
```

Start FLEX-ES Configuration Utility
Configuration processing *SUCCEEDED* with no errors
Data Space Manager Terminated

We could issue the cfcomp command directly because we earlier placed /usr/flexes/bin in our Linux PATH. If you did not do this, you would need to use a full path name with the command:

```
/usr/flexes/bin/cfcomp defR12A
```

Our FLEX-ES definition file consists of two sections: the system section and the resources section. The compilation creates files S12A.syscf and R12A.rescf, based on the S12A and R12A names contained in our definitions.38 We can then invoke the FLEX-ES resource administrator to activate our resources:

```
$ su
Password: (<--- enter root password when this prompt is shown)
# cd /usr/flexes/rundir
# resadm -s R12A.rescf
# resadm -r
```

38 We could have used two definition files, one for the system definitions and one for the resource definitions. The use of separate system and resource definition files is most common when multiple instances of FLEX-ES are used (meaning multiple S/390s being emulated at the same time). This is unlikely to be the case with a ThinkPad/EFS system. If we did use separate system and resource definition files, we would need to compile both of them in order to obtain the syscf and rescf files we need to actually use a S/390 emulation instance. We found it more convenient to always combine the definitions in one source file.
Again, we could issue the `resadm` command directly because `/usr/flexes/bin` is in our PATH. You must be `root` to issue start or terminate `resadm` functions. This is the only `root` operation required for normal EFS use.

**Comments**

The steps just shown, to compile a system and resource definition file and to activate the resources (but not the system), require some explanations.

Our definition file source had been placed in `/usr/flexes/rundir`. We created the `rundir` directory just after we installed FLEX-ES; it was not created automatically. Two separate definitions are needed for a FLEX-ES emulated S/390. One defines the `system` being emulated:

- Central memory size available to this system
- Expanded memory size
- Number and type of CPUs
- Number and usage of channels
- Control units for all system devices

The other defines `resources` for a single system or multiple systems, such as:

- Interfaces for all the control units defined in the system section
- Devices for all the control units defined in the system section

These two definitions can be two separate files or combined into one file. We elected to combine them into one file, as listed in “Basic definitions for a single HDD” on page 97.

The FLEX-ES configuration file(s) (resources and systems) must be `compiled`, using the command `cfcomp`. Compilation produces two files. Parts of the output file names are taken from the names in the definition files, and part of the names (`syscf` and `rescf`) are fixed. In our case we will have `S12A.syscf` and `R12A.rescf`, based on the names coded in our definitions.

Once compiled, the resource portion (the `rescf` file) can be activated with the `resadm` command:

```
$ cd /usr/flexes/rundir
$ su                      (switch to root)
# resadm -s R12A.rescf    (must be root to use -s)
```

You can list all active resources with the command:

```
$ resadm -r
Resource: CPU Flags: READY Type: CPU Port: 32772 Pid: 17483
Resource: CU3990 Flags: READY Type: CU Port: 32778 Pid: 17485
Resource: CU3991 Flags: READY Type: CU Port: 32780 Pid: 17486
Resource: CU3174 Flags: READY Type: CU Port: 32782 Pid: 17487
Resource: CU3172 Flags: READY Type: CU Port: 32776 Pid: 17488
Resource: CU3480 Flags: READY Type: CU Port: 32784 Pid: 17489
Resource: NETCU Flags: READY Type: NETCU Port: 32786 Pid: 17492
Resource: TS3270 Flags: READY Type: TS3270 Port: 32774 Pid: 17493
```

If you try the `resadm -r` command very quickly after the `resadm -s` command, many of the resources may be `NOTREADY`. They should become `READY` within a few seconds. You normally have no use for the `port` and `pid` numbers, but these can be useful in debugging situations.

More options of the `resadm` command are discussed in “The resadm command” on page 112.

Once the resources are active, the `system` portion is ready to be started and IPLed. This is easier to do with a shell script.
5.1.1 Building a shell script

We used vi to create the following shell script file in /usr/flexes/rundir/sh12A. (The file name is arbitrary.) The contents are:

```
flexes S12A.syscf
xmodmap -e 'keysym Alt_L = Alt_L Meta_L'
xset fp+ /usr/flexes/fonts
xset fp rehash
echo ' Verify your master console session is started '
  ' and then ipl your system '
  '
x3270 -model 3 -keymap pc -port tn3270 localhost:mstcon &
x3270 -model 3 -keymap pc -port tn3270 localhost:L701 &
flexescli localhost S12A
```

The `flexes` command (in the shell script) starts an emulated S/390, using the indicated compiled system definition file.

The next three commands prepare X Windows parameters, and the `x3270` commands start 3270 client windows on the Linux screen. We elected to identify specific 3270 sessions (mstcon and L701) for these two 3270 client windows. These session names match names in our resources definition. There is no requirement to start these x3270 sessions here (in the shell script), but the terminal for the z/OS master console(s) should be started before IPLing z/OS. Using the shell script this way provides a convenient, automatic way to establish the z/OS console. We normally use the second 3270 session as a TSO terminal.

The last command of this shell script, `flexescli localhost S10A`, starts the FLEX-ES Command Line Interface (CLI) program in interactive mode, with a `flexes>` prompt replacing the default Linux prompt. You can enter `flexes` commands here or enter a `quit` command to return the session to a Linux prompt.

Assuming that we have already started the resource manager (with a `resadm -s S10A.rescf` command), we can now invoke the shell script to start S/390 operation:

```
$ sh sh12A
flexes> ipl a80 0a82cs
flexes>
```

Here we started the sh12A shell script. The last command in the shell script starts `flexescli` (the command line interpreter, or CLI) in interactive mode, and this program provides the `flexes>` prompt. At this point we can enter CLI commands, some of which are outlined in “CLI commands” on page 113. We entered an `ipl` command to start z/OS. We could have included the `ipl` command in the sh12A script, but we found it more convenient to enter it manually because we often change the IPL parameter values. (The “0a82cs” in the example is an IPL parameter value.)

If we wanted to include an `ipl` command in the shell script, it could look like this:

```
flexes S12A.syscf
xmodmap -e 'keysym Alt_L = Alt_L Meta_L'
xset fp+ /usr/flexes/fonts
xset fp rehash
x3270 -model 3 -keymap pc -port tn3270 localhost:mstcon &
x3270 -model 3 -keymap pc -port tn3270 localhost:L701 &
```

39 If we omitted the session identification, we would obtain the Terminal Solicitor selection menu on these sessions.
40 If you do not start the x3270 sessions this way, you would need access the Terminal Solicitor and start the TN3270 session that will be used for the z/OS console before starting an IPL.
41 For example, you could enter an IPL command at this point. Some users prefer to code an IPL command in their shell script, while others prefer to enter it at the `flexes` prompt.
echo 'ipl a80 0a82cs' | flexescli localhost S12A

This illustrates the two ways in which `flexescli` can be used. If a command is piped to it (with `echo`), then `flexescli` will execute that command and quit. If nothing is piped to it (as in the last line of the shell script), then `flexescli` starts in interactive mode and issue the `flexes` prompt. The two operands for `flexescli` are the IP name/address of the system running FLEX-ES (localhost) and the name of the FLEX-ES instance (S12A). The requirement for these two operands becomes more apparent if you are running multiple FLEX-ES images with networked channels.

Note that we hardcoded the name of our FLEX-ES instance (S12A) in the shell script. If we intended to use several different FLEX-ES defined systems, we would need multiple shell scripts. There are many ways you might decide to customize this operation once you are more familiar with it.

The default x3270 keyboard parameters may not be what you want. Read “x3270 client” on page 67 for more information about the `Enter`, `New Line`, and `Reset` key assignments.

The IPL parameters in this example (address A80 and parameter 0A82CS) are suitable for the first IPL of the z/OS AD 1.2 system. You should read the AD documentation for additional parameters.

**Terminal Solicitor**

Starting the resource manager, `resadm`, automatically starts the FLEX-ES Terminal Solicitor. In order to connect to the FLEX-ES Terminal Solicitor, we connect a TN3270e client to the Linux IP address using port 24. Port 24 is the default port for the FLEX-ES Terminal Solicitor. The client system can be connected to any LAN that is connected to our Linux. The Terminal Solicitor presents a panel to us with the names of available 3270 devices (these are often made available by the FLEX-ES CLI `mount` command, although we elected to hardcode them in the resource definitions):

```
Welcome to the FLEX-ES Terminal Solicitor (node: t20)

Please select (X) the desired service and press enter
(PAL to exit; CLEAR to refresh)

_ M702   _ M703
_ M704   _ M705   _ M706   _ M707
_ M708   _ M709   _ M70A   _ M70B
_ M70C   _ M70D   _ M70E   _ M70F
```

*Figure 5-1  Terminal Solicitor panel  (on a TN3270e screen)*

If we had not started the x3270 sessions for `mstcon` and `L701` in our shell script, then these names would also appear in the Terminal Solicitor list. We would need to select `mstcon` from the list to activate that session before IPLing z/OS, since it will be used for the MVS master console. When we select a terminal (by marking it with an X and pressing Enter), the Terminal Solicitor screen (on our client TN3270e session) is replaced by a 3270 connection to the indicated terminal address. After z/OS is up, this will normally result in the USSTAB logo screen.

---

42 It can also be another window on our Linux desktop.
43 Once a 3270 terminal device is selected by a user, it is removed from the Terminal Solicitor panel. When the device is freed by the user, it reappears on the Terminal Solicitor panel and can be selected by another user.
5.1.2 IPL z/OS

The z/OS IPL starts when you execute a *flexes* `ipl` command. The `ipl` command can be entered from the *flexes* window or in a startup script, as shown in 5.1.1, “Building a shell script” on page 50. Once our z/OS was operational, we tried a few typical z/OS operator commands, as shown in the following (the operator commands are shown in bold type):

- 11.06.30  
  11.06.32  
  *d m=stor*

  
  
  RESTART STORAGE STATUS  
  ONLINE-NOT RECONFIGURABLE  
  0M-256M  
  ONLINE-RECONFIGURABLE  
  NONE  
  PENDING OFFLINE  
  NONE

- 11.42.47  
  *d m=cpu*

  
  PROCESSOR STATUS  
  ID CPU SERIAL  
  0 + 00F02B1245  
  CPC SI = 1245.L01.FSI.FREM.0000000000000000  
  + ONLINE - OFFLINE . DOES NOT EXIST W WLM-MANAGED

- 11.43.44  
  *d a,l*

  
  ACTIVITY 665  
  JOBS M/S TS USERS SYSSAS INITS ACTIVE/MAX VTAM OAS  
  00002 00011 00000 00027 00012 00001/00010 00007  
  JES2 JES2 IEFPROC NSW S VTAM VTAM VTAM NSW S  
  DLF DLF DLF NSW S RACF RACF RACF NSW S  
  VLF VLF VLF NSW S INETD4 STEP1 BPXINIT OWT AO  
  LLA LLA LLA NSW S TSO TSO STEP1 OWT S  
  SDSF SDSF SDSF NSW S TCPIP TCPIP TCPIP NSW SO  
  PORTMAP PORTMAP PMAP OWT SO FTPD1 STEP1 FTPD OWT AO  
  NFSS NFSS GFSAMAIN NSW SO  
  P390 OWT

- 11.41.55  
  *d u,dasd,online*

  
  UNIT STATUS  
  VOLSER VOLSTATE  
  0A80 3390 S Z2RES1 PRIV/RSNT  
  0A81 3390 A Z2RES2 PRIV/RSNT  
  0A82 3390 A 0S39M1 STRG/RSNT  
  0A87 3390 A Z2USI1 PRIV/RSNT  
  0A91 3390 A WORK02 STRG/RSNT

---

44 The AD system expects a 3270 terminal at address 700 for its master console. The name *mstcon* is arbitrary, and matches the name chosen for a 3270 defined at address 700 when the resource definition file was created.

45 Many 3270 emulator users set up the right-hand Ctrl key as the logical 3270 Enter key, since this most closely matches a "real" 3270 keyboard. We did this and, using the IBM PCOM emulator, we pressed the right-hand Ctrl key.
The processor type (in the `d m=cpu` response) is 1245. This is one of the IBM-assigned processor types for an EFS system. Otherwise, the displayed responses are exactly the same (except, perhaps, for storage sizes) as would be found when running on any S/390 hardware.

### 5.2 Operation and use

Once you have z/OS IPLed, operation is the same as for any other z/OS installation.

#### 5.2.1 User terminal connection

A TSO user, for example, would normally connect to the system by connecting a TN3270e session to the Linux IP address (or name, if you have a Domain Name Server service), port 24. This should produce the Terminal Solicitor display, as shown in Figure 5-1 on page 51. The user would then select one of the available sessions (in the Terminal Solicitor display) and would be connected to this S/390 address. When he eventually drops the TN3270e session, the S/390 3270 address will be restored to the Terminal Solicitor display for others to use.

If you start z/OS TCP/IP, a user could connect directly to it by using TN3270 and the IP address you assigned to z/OS TCP/IP. In this case, the user would not be connected to the Terminal Solicitor, but would be connected directly to z/OS TCP/IP. See the AD documentation for a brief discussion of customizing z/OS TCP/IP.

#### 5.2.2 IODF requirements

z/OS requires an IODF data set that defines the I/O configuration seen by the software. This normally matches the IOCDS defined for the S/390 I/O hardware configuration. An EFS platform does not have an IOCDS. All resources are defined in FLEX-ES system and resource files, compiled with the FLEX-ES resource compiler `cfcomp`, and then activated by the FLEX-ES resource manager `resadm` command.

An IODF is still required within a z/OS system, but the HCD input to generate it does not need to define control unit details. That is, a simple device definition (device number, type, optional features) is all that is required. The z/OS dynamic I/O redefinition capability is not available.

In response to a `d ios,config(all)` command, we received the following:

```
- 11.38.06    d ios,config(all)
11.38.06     I05506I 11.38.06 I/O CONFIG DATA 656
ACTIVE IODF DATA SET = SYS1.IODF99
CONFIGURATION ID = OS390          EDT ID = 00
HARDWARE SYSTEM AREA DATA COULD NOT BE OBTAINED
ELIGIBLE DEVICE TABLE LATCH COUNTS
0 OUTSTANDING BINDS ON PRIMARY EDT
```

You can, however, perform software dynamic configuration changes via z/OS Hardware Configuration Definition (HCD) dialogs or the z/OS ACTIVATE command, provided the affected devices are included in the FLEX-ES definition files.
5.2.3 System performance monitors

Because I/O sub-channel blocks are not maintained by FLEX-ES software emulation, the z/OS Resource Measurement Facility (RMF) is not fully supported. You can run RMF, or other system performance monitors, but some of the reporting (especially when it comes to I/O activity) will not be complete.

When we started RMF, it reported the absence of an IOCDS in this environment and automatically terminated I/O queuing activity reporting:

```
  11.22.34  s rmf
  11.22.35  STC00439 $HASP373 RMF STARTED
  11.22.36  STC00439 ERB100I RMF: ACTIVE
  11.22.36  STC00439 ERB2651 RMF: IOCDS INFORMATION UNAVAILABLE TO RMF.
             RESPONSE CODE 01F0
  11.22.37  STC00439 ERB2601 ZZ : I/O QUEUING ACTIVITY RMF REPORT TERMINATED
  11.22.38  STC00439 ERB1001 ZZ : ACTIVE
```

SYS1.LOGREC may not have hardware error information. Consequently, any report produced by the Environmental Recording and Editing Program (EREP) will have limited value when it comes to hardware detected errors. We have not investigated this area in more detail.

5.2.4 TCP/IP for z/OS

It is possible to share a single Ethernet or token ring LAN adapter between Linux (including the FLEX-ES Terminal Solicitor) and z/OS TCP/IP. This requires some coordination between the FLEX-ES definitions and the z/OS TCP/IP profile data set parameters. Our goal was to assign IP address 10.20.30.31 to z/OS TCP/IP, using the z/OS device addresses E20 and E21 for the interface. (Our Linux IP address was 10.20.30.30.)

We included the following FLEX-ES definitions:

```
System definition:
  channel (3) local
  cu devad(0xE20,2) path(3) resource(R10A3172)

Resource definitions:
  R10A3172: cu 3172          (for token ring: cu 3272TR)
             interface local(1)
             options ‘ipaddress=10.20.30.31’
             device(00) 3172 eth0    (for token ring: tr0)
             device(01) OFFLINE
             end R10A3172
```

We changed the following lines in the TCPIP.PROFILE.TCPIP data set of z/OS:

```
... DEVICE LCS1 LCS E20
... LINK ETH1 ETHERNET 0 LCS1  (LINK TR1 IBMTR 0 LCS1 for token ring)
... HOME 10.20.30.31 ETH1      (HOME 10.20.30.31 TR1)
... GATEWAY
  10   = ETH1  1492  0.255.255.0  0.20.30.0  (or TR1)
... START LCS1
```
The fourth element in the LINK statement specifies the use of LAN adapter 0 (sometimes known as the MPTS adapter number, from the OS/2 MPTS program). Adapter 0 is the default LAN adapter number for FLEX-ES. This number is not related to any Linux LAN number. In particular, it is not related to the ethn, or trn numbers used in Linux. The z/OS TCP/IP PROFILE statement can use various forms of static routing statements; the one shown in the example is the oldest of these forms.

In order to share a LAN adapter with Linux, we must specify, in the FLEX-ES resources file, the IP address to be used by z/OS. It must also be specified in the z/OS TCP/IP profile data set, as shown here. With the specifications described here and using the z/OS 1.2 AD system, we have these connection options from another host on the network:

- **TN3270** to 10.20.30.30, port 24 -- this connects to the Terminal Solicitor (assuming *resadm* is started). We can then select a session and should receive the USSTAB logo screen. The Terminal Solicitor is a Linux program, provided with FLEX-ES.
- **TN3270** to 10.20.30.31, port 23 -- this connects directly to z/OS TCP/IP. In recent AD systems (where a USSTAB function is defined for TCP/IP) this should produce the logo screen.
- **telnet** to 10.20.30.30, port 23 -- this connects to Linux (if telnet connections are enabled).
- **telnet** to 10.20.30.31, port 1023 -- this connects to z/OS UNIX System Services, as an ASCII terminal (assuming you are using a recent AD system).
- **telnet** to 10.20.30.31, port 23 -- this produces a line-mode connection (via VTAM) to TSO and is not very useful.

### 5.3 Shutting down

Shutting down the system involves the following steps:

- Shut down z/OS. (With the AD systems we would enter `s shutdown` on the MVS console to start a VTAMAPPL script that issues various commands.)
- **Using the flexes window:**
  ```
  flexes> shutdown                      (terminates S/390 emulation)
  $  
  ```
- **Using the root window:**
  ```
  # resadm -T                           (terminate all resource managers)
  # resadm -k                           (end resadm)
  #  
  ```

To start the S/390 system again, we would use the `resadm -s xxx.rescf` command (using the name of our compiled resource file), start the shell script, and then issue an `ipl` command from the `flexes` prompt. When you shut down Linux, resadm will also be stopped. You are not required to use the `resadm -k` command.

### 5.4 More comments

Some users (and Business Partners) prefer to start *resadm* automatically. This is done by placing a *resadm* command in the `/etc/init.d/flexes` file.\(^{46}\) We have not done this (in the setup described in this redbook) for the following reasons:

- The *resadm* command must be hardcoded if it is included in the `/etc/init.d/flexes` file; that is, it must specify the name of the resources files to be started. In our case, we had several

---

\(^{46}\) Starting with FLEX-ES release 6.1.15, this is the `/etc/init.d/FLEXES` file. You can use the symbolic name `S90FLEXES` instead because this name remains constant.
different resources files that we used at different times. (These include R12A.rescf and R12B.rescf that are used in our examples.) We preferred to enter our resadm -s command manually, and select which rescf file to use at that time.

- We sometimes used our Linux system for other purposes, and did not want any of the FLEX-ES functions active.

While customizing new FLEX-ES definitions, we often need to stop the current resources definitions and start a new set. There are two ways to do this:

```bash
# resadm -T          stop all the current FLEX-ES resources
# resadm -k          kill resadm
# resadm -s new.rescf start a new resources file
# resadm -r          verify that it looks OK
```

----- Better method -------

```bash
# resadm -T          stop all the current FLEX-ES resources
# resadm -x new.rescf start a new resource file
# resadm -r          verify that it looks OK
```

In both examples, we assume you used the rescf command to compile a new set of resources, creating new.rescf. The point is that it is better not to completely stop resadm (with the -k option). See “The resadm command” on page 112 for a listing of the resadm options.

### 5.5 Options

The next chapter discusses a few of the options you have available to further customize your operation. Our most common enhancement is to create at least two definition files (one with a minimal IPLable system, and one with all our emulated volumes) and two shell scripts to start these different configurations. Another variation is to have a single definition and use flexes mount commands to add additional emulated volumes.

There are many variations possible. However, we strongly suggest that you operate with a minimum system until you feel comfortable using it and until you understand every single character in the FLEX-ES definition files you use.

**Tip:** We must note that FLEX-ES is very unforgiving about errors. Any error, whatsoever, in the definition files, the startup script, or the emulated volume files will prevent FLEX-ES emulation from starting. Once it starts, it is very stable. However, you must have everything exactly right in order to get it started.
Additional topics

Most of this redbook is concerned with the installation of a basic, entry-level ThinkPad/EFS system. Many additional options and techniques can be used to enhance the entry-level system, and some of these and other topics are briefly discussed here.

6.1  Linux windows and FLEX-ES operation

We found that we often had four windows (terminal emulation programs) open on the Linux desktop:

- A Linux window running `flexescli` (with the `flexes` prompt)
- A Linux window with `su` to `root`
- An x3270 session for the z/OS master console
- An x3270 session for TSO

The root window was useful for issuing `resadm` commands. The window with the `flexes` prompt is generally required for FLEX-ES operation. We found that having two 3270 sessions was our most useful arrangement.

Our typical startup process went like this:

Log into Linux with userid `flexes` (working directory is `/usr/flexes`)
Start two terminal windows on the Linux desktop

Window 1

```
$ su  
# cd rundir  
# resadm -s R12A.rescf  
# resadm -r
```

(Window 1 continues)

Window 2

```
$ cd rundir
$ sh sh12A
```

Wait for 3270 sessions to start

```
flexes> ipl a80 0a8200
```

(Note that prompt changes to `flexes`)

Operate z/OS as normal, using 3270 sessions. Shut down z/OS when finished.

Window 2
This process can be automated or abbreviated in a number of different ways. We wanted to keep a straightforward, manual control over FLEX-ES startup and used the steps shown here.

Another window (not root) was sometimes useful for running `vmstat` commands and similar utility functions. (This could be done from the root window if you ignore the usual advice to avoid root usage except when specifically required. Or, you could `exit` from root in Window 1 and work from there.)

### 6.1.1 Change resource definitions

Changing resource definitions for an emulated S/390 requires several steps:

- Edit the FLEX-ES definition file to make the changes. You can edit the same file you used to create the resources that are currently running. For example:
  
  ```
  $ vi defR12B
  ```

- Use the `cfcomp` command to compile the new definitions:
  
  ```
  $ cfcomp defR12B
  ```

- Bring down z/OS, if it is running.

- Stop the emulated S/390:
  
  ```
  flexes> shutdown
  ```

- Stop the current FLEX-ES resources that are running:
  
  ```
  # resadm -T
  ```

- Start the newly compiled resources:
  
  ```
  # resadm -x R12B.rescf
  ```

- Start your shell script (which starts FLEX-ES emulation) and IPL your system:
  
  ```
  $ sh sh12B
  flexes> ipl a80 0a8200
  ```

Note that we did not need to completely stop `resadm` with the `-k` option. The `-T` option ended all the currently running resources and the `-x` option started a new set of resource definitions.

### 6.2 Basic debugging

We present here a few common debugging steps that may be useful if you have a problem starting FLEX-ES. We must note that FLEX-ES typically does not produce error messages oriented to a S/390 person. An error message indicates that an error occurred, but you may need to ignore the specific text of the message and attempt to deduce the original cause of the problem.

A number of common errors occur if userid `flexes` and groupid `flexes` do not own the emulated I/O files and most of the files in `/usr/flexes/rundir`. (We assume your `rescf` and `syscf` files, along with any shell scripts you might use, are in this directory.) You can use a command such as `ls -al /usr/flexes/rundir` to verify ownership, and a command such as `chown flexes:flexes /usr/flexes/rundir/sh12A` to change ownership. (Of course, if you `cd` to the relevant directory first, you will not need to enter full path names to files in that directory.)
Do all your resources start when you use the `resadm -s xxx.rescf` command? You can check this with a `resadm -r` command to list the currently active resources. You might verify that the resource names (listed by the `resadm -r` command) exactly match the resource names you specified in your system definition. **Immediately** after starting the resources, a `resadm -r` command may show them as NOT READY. A few seconds after starting the resources, a `resadm -r` command should show them as READY.

Do your `rescf` and `syscf` files match your current definitions (in your definition source file)? Changing your definition source file(s) has no effect until you recompile them and start (or restart) `resadm` with the newly compiled files. If you are not certain, you can simply compile your source file again. In our examples, this would be:

```
$ cd /usr/flexes/rundir
$ cfcomp defR12A
```

(Using your correct source file name)

Be certain you have a clean compile. This is indicated by the messages:

```
Configuration processing *SUCCEEDED* with no errors
Data Space Manager Terminated
```

Are you using the correct files? The `resadm` command uses the `rescf` file and the `flexes` command uses the `syscf` file. It is easy to reverse the two file suffixes, and the resulting error message does not identify the exact problem.

Once the resource manager (`resadm`), the emulator (`flexes`), and the command line interface (`flexescli`) are started, you may find the following CLI commands useful:

```
flexes> d devstate 700  (display state of emulated device)
flexes> d ckdcachestats A80  (verify I/O occurs to a disk)
flexes> d psw
```

If your emulated 3270 terminal operation is erratic, be certain you know where the 3270 Enter key is located, and where the 3270 New Line key is located. You may be creating a 3270 Enter operation when you intended to use the New Line key. See “x3270 client” on page 67 for suggestions in this area.

Be certain that your disk model numbers are consistent. For example, if you created a 3390-2 (using `ckdfmt` or `ckdconvaws`) but you specify a 3390-3 in your resource definitions, the system may not start.

**Files that may grow**

FLEX-ES writes trace and debugging information in `/var/adm/flexes`. It manages these files; they should not grow to unreasonable numbers. The `/var/adm/flexes` directory will have additional directories corresponding to the system names and some of the control unit names you defined in your FLEX-ES definitions. Many of the trace files written in these directories have names beginning with a period. Use an `ls -al` command to see all the files. For example:

```
# cd /var/adm/flexes
# ls
CU3990  CU3991  resmanagererr  S12A  S12B
# cd S12A
# ls -al
-rw-rw-r--   1  flexes   flexes    944 Apr 15 09:03 devtr.1425.0
-rw-rw-r--   1  flexes   flexes    800 Apr 15 09:03 devtr.1429.0
-rw-rw-rw-   1  flexes   flexes    944 Apr 15 09:03 .stderr.2.2.3637
-rw-rw-rw-   1  flexes   flexes      0 Apr 15 09:04 .stderr.2.2.4136
# dtprint devtr.1425.0
..........
```
The FLEX-ES dtprint command may be used to print these files. The information produced is intended for debugging use by FSI developers and is not documented. Many of the files may have zero length.

6.3 Operating Systems Messages console

Current S/390 and zSeries machines have an Operating Systems Messages console function that is provided through the Support Element (SE) or a Hardware Management Console (HMC). This console function is sometimes known as the “system console” or the “hardware system console.” z/OS attempts to use it if all other MVS consoles fail.

FLEX-ES emulates this console through the CLI window—the window with the flexes prompt. Messages written from the S/390 to the Operating Systems Messages console appear after the flexes prompt. You need to press Enter (with the desktop focus in this window) to restore the flexes prompt. You can reply or enter commands through the Operating System Messages console by using the CLI command hwc:

    flexes> SPECIFY SYSTEM PARAMETERS
    (message from z/OS)
    (press Enter to get flexes prompt)
    flexes> hwc R 00,CLPA
    flexes>

6.4 Security

As explained earlier, FLEX-ES is a layer of software that resides and operates between a S/390 operating system and an underlying Linux system. All the security features and functions that come with a z/OS system work as on any other S/390 platform. However, it is possible for a Linux user with sufficient privilege to gain access to the contents of an emulated DASD or central storage associated with an emulated CPU, and so forth.

An EFS owner must plan and manage traditional Linux security functions for the underlying Linux system, as well as traditional S/390 security management. If the EFS platform is used only for S/390 operation, this can be fairly simple.

A unique concern involves the z/OS master console(s). These can be implemented through the Terminal Solicitor. This is convenient, but offers an opportunity for an unwanted person to connect as a master console. We suggest that you have a master console on the server display and direct any other master consoles47 to specific IP addresses and not through the Terminal Solicitor. This is done by specifying an IP address in the FLEX-ES resources file, instead of a terminal name. This means, of course, that the client systems connecting to the emulated 3270 interfaces must have static IP addresses.

The FLEX-ES resource manager uses TCP/IP port 555 to talk with other instances of FLEX-ES resource managers. There may be a potential for problems if someone hacks this port. We are not aware of any instances of this, but it is a potential concern if you use an open network to link multiple FLEX-ES systems at this level. (Would you use an open network for shared DASD data flow on a “real” S/390? You need to look at this potential exposure from this viewpoint and not from a PC networking viewpoint.)

47 We are using the term “master console” loosely here to mean any z/OS operator console.
6.5 Server memory

The \textit{memsize+essize+cachesize*11+DASD cache} total value (described in “System definitions” on page 103 ), when translated to bytes of storage, approximates the amount of Linux virtual storage needed to run an instance of S/390 emulation. If you emulate two S/390 systems (at the same time), you will need to add the values for each of the two emulated systems.\footnote{Some of the DASD cache memory may be shared among multiple emulated S/390s.} You can emulate more S/390 instances, but each one will require more memory.

The key principle is that the Linux system \textbf{should not be forced to page} when running S/390 emulation. Linux might perform considerable paging when getting started, but once a S/390 instance is started and the system \textit{working set} is established, the average Linux paging rate should be close to zero. (S/390 operating systems running in the emulated S/390 might have much higher paging rates; this is not the point at issue here.) This almost always means that the server (ThinkPad or Netfinity) \textbf{real memory must be considerably larger than the emulated real memory of all the S/390 instances that are active at any one time.}

In principle, the Linux memory needed to emulate a S/390 is “just” virtual memory to Linux and might exist in a much smaller real memory used by Linux. In practice, this does not work well. Your server should have enough memory so that all the S/390 requirements (\textit{memsize + essize + cachesize*11 + DASD cache}) fit in your \textbf{real server memory}, without requiring Linux paging. If you assume 100 MB for Linux, FLEX-ES programs, TCP/IP operation, and so forth, then the 512 MB ThinkPad in our examples might be used as follows:

\begin{verbatim}
Rough guess for Linux, FLEX-ES, etc 100 MB
S/390 memsize 256 MB
S/390 essize 64 MB
FLEX-ES instruction cache * 11 22 MB
default DASD cache for 10 volumes (approx) 10 MB
--------
462 MB
\end{verbatim}

This leaves a reasonable margin in a 512 MB system. The 100 MB for Linux is just a guess. The \textbf{working sets} (as seen by Linux) of the emulated S/390 memory (central and expanded), the FLEX-ES instruction cache, and most of the DASD cache will normally be their full sizes. The working set of Linux itself and its many system processes are much harder to determine.

6.5.1 The \texttt{vmstat} command

The traditional UNIX command for monitoring swapping\footnote{In S/390 terms, we would say \textit{paging rates}. In older contexts, \textit{swapping} has a different meaning than \textit{paging}. In modern Linux systems, \textit{swapping} appears to have exactly the same meaning as \textit{paging}.} rates, \texttt{sar}, is not available in base Linux distributions. The \texttt{vmstat} command can be used instead. For example, the command \texttt{vmstat 10 2} means to run \texttt{vmstat} with 10 seconds between reports and quit after 2 reports.

\begin{verbatim}
# vmstat 10 2
proc memory swap io system cpu
 r  b  w  swpd  free  buff  cache  si  so  bi  bo  in  cs  us  sy  id
 0  0  0  0329156 28768  70156  0  0  4  1  105 196  1  0  99
 0  0  0  0329156 28768  70156  0  0  0  0  105 196  1  0  99
\end{verbatim}

You should refer to your Linux documentation or \texttt{man} listings for more complete information about \texttt{vmstat}. Very briefly, the key fields are documented as:

\begin{itemize}
  \item Procs
    \begin{itemize}
      \item r: number of processes waiting for CPU time
      \item b: number of processes in uninterruptible sleep
      \item w: number of processes swapped out but otherwise runnable
    \end{itemize}
\end{itemize}
Memory
- swpd: the amount of virtual memory used (in the swap file) (kB)
- free: the amount of idle memory (kB)
- buff: the amount of memory used as buffers (kB)

Swap
- si: amount of memory swapped in from disk (kB/s)
- so: amount of memory swapped out to disk (kB/s)

IO
- bi: blocks received from a block device (blocks/s)
- bo: blocks sent to a block device (blocks/s)

System
- in: the number of interrupts per second (including clock interrupts)
- cs: number of context switches per second

CPU
- us: user time (percent of CPU time)
- sy: system time (percent of CPU time)
- id: idle time (percent of CPU time)

We are particularly interested in swap (paging) rates. Notice that the unit of measurement is kilobytes per second.

The Linux swapping rate is meaningful (for our discussion) only in a steady-state condition with a typical S/390 workload. Linux swapping while booting, or while starting FLEX-ES, is not relevant. Unusual S/390 work, such as CLPA processing or very unusual disk access patterns (affecting disk caches) might temporarily drive Linux into swapping. This is not good, but can probably be tolerated for short periods.

6.5.2 Importance of Linux swapping

Why is the Linux swapping rate so important? A reasonable analogy is CICS paging in an z/OS system. A system with many TSO users might have sustained paging rates of hundreds of pages per second (on a larger S/390) with no ill effects, but CICS on the same system would require a paging rate close to zero. The problem is that the whole address space (CICS, for example) is placed in wait when a page fault occurs. Placing CICS\textsuperscript{50} in wait causes all the CICS users serviced by that address space to wait while the page fault is resolved. A page fault in a TSO user address space causes only that one user to wait.

FLEX-ES operation is close to the CICS analogy. A Linux page fault in a key FLEX-ES process may cause the whole emulated S/390 instance to wait until the Linux page fault is resolved.

Remember that a page fault in a FLEX-ES S/390 instance has a very different effect than a page fault in Linux. For one thing, the S/390 page fault is seen only by the FLEX-ES emulation program—it is an emulated page fault. It is handled, by z/OS, as a S/390 page fault. If it occurs in a TSO user address space or a batch address space, it affects only that address space. If it occurs in a CICS address space, it affects all the users of that CICS. This is business as usual for z/OS.

\textsuperscript{50} Modern CICS systems ameliorate this situation in various ways; the description here should be regarded as conceptual.
The key message is that you should adjust your FLEX-ES system parameters (emulated S/390 memory, disk caches, instruction cache) to avoid Linux swapping. Defining a smaller emulated S/390 memory size may increase z/OS paging. Of course, it would be nice to avoid any paging, but z/OS paging is much less damaging than Linux swapping and your tradeoffs should always be in this direction. You can juggle disk cache versus instruction cache versus S/390 memory allocations for your best performance. Simply be careful not to push Linux into swapping.51

6.6 Ultrabay

Recent ThinkPad units have a “bay” that can hold a CD-ROM drive, or a diskette drive, or a second hard disk drive—but only one at a time. Current Windows operating systems permit the user to change these devices while the system is running. We were unable to make clean (without side effects) changes under Linux, while Linux was running. We found it was easier to shut down Linux, let the ThinkPad power down, and then swap devices in the Ultrabay. Linux detected the new device correctly when it was restarted.

6.6.1 Using a second Linux hard disk

We obtained a second 32 GB hard disk drive (IBM part number 08K9511), along with the mounting tray (IBM part number 08K6068) needed to use it in the Ultrabay. We removed the CD-ROM drive and installed the second hard disk while the ThinkPad was turned off. We turned power on and let Linux boot. We then determined the Linux identity of the second drive:

```
# cat /proc/partitions
major minor #blocks name   (... more data .....)
 3    0      nnnnn    hda
 3    1      nnnnn    hda1
 3    2      nnnnn     hda2
...  ...     ....      ....
22    0    31253040    hdc  0 0 0 0 0 0 0 0 0 0
```

The first hard disk (internal in the ThinkPad) is /dev/hda; looking at the above listing, we see a second hard disk as /dev/hdc. This drive (as expected for a new disk) had no partitions. (We do not know why Linux selected hdc instead of hdb, which would logically be next in sequence.)

As discussed in earlier chapters, we want to create two file systems on the second hard disk. One would be about 2 GB and use mount point /holding.52 The other would use all the remaining space and have mount point /s391. We intended to use /holding for compressed tar images (created as backups) and other temporary files and we might have considerable creation/deletion activity in this file system. The /s391 file system would be solely for emulated S/390 volumes. Using two file systems seemed to reduce the chances for significant fragmentation of emulated volume files.53

51 Again, we stress that “Linux paging” refers to steady-state operation after S/390 emulation is started. Linux booting or FLEX-ES startup may cause Linux paging and we are not concerned with this temporary effect.
52 The size of this partition is quite arbitrary.
53 We may have been too sensitive about fragmentation. Experienced Linux users say there are practically no fragmentation effects in normal Linux file systems. However, they admit that FLEX-ES operation (57 KB reads or writes when emulating a 3390) is not typical Linux I/O. We felt that disk I/O may be the weakest point of ThinkPad/EFS performance and felt that avoiding fragmentation might help performance. We welcomes any measurement data (on an EFS system) that may help resolve this issue.
We also noted that IBM, like most other manufacturers, uses decimal numbers to describe disk capacity (1M = 1,000,000) while Linux utilities typically use power-of-two numbers (1M = 1,048,576). Using Linux numbers, the capacity of the disk was about 29 GB.

We partitioned the disk with `fdisk` and formatted it:

```
# fdisk /dev/hdc
n  (fdisk option to add a partition)
Command action: p  (a primary partition)
Partition number: 1
First cylinder (1-66144): 1  (default = 1)
Last cylinder (1-66144): 4140  (this makes a partition of about 2 GB)
n  (fdisk option to add a partition)
Command action: p  (another primary partition)
First cylinder (4141-66144): 4141
Last cylinder (4141-66144): 66144
p  (print the partition table)
Device  Boot  Start   End    Blocks   ID   System
/dev/hdc1   1     4140    1956118+   83   Linux native
/dev/hdc2  1414  66144    2929680   83   Linux native
```

(if the ID is not 83 = Linux native, use the `t` option to change it)

```
t  (fdisk option to change partition ID)
Partition number: 1
Hex code: 83  (83 is Linux native)
t
Partition number: 2
Hex code: 83
p  (fdisk option to list partitions)
w  (fdisk option to write partition table and end)
```

```
# mkdir /s391
# mkdir /holding
# mke2fs /dev/hdc1
# mke2fs /dev/hdc2
# mount /dev/hdc1 /holding
# mount /dev/hdc2 /s391
# chown flexes:flexes /holding
# chown flexes:flexes /s391
```

We used the new disk space to confirm that it worked, and then edited `/etc/fstab` to cause the new file systems to be automatically mounted at boot time. (There are no unwanted effects if the second hard disk is not present when Linux is booted; the file systems on that disk are simply not mounted.) We added two lines at the end of `fstab`:

```
LABEL=/ / ext2 defaults 1 1
LABEL=/boot /boot ext2 defaults 1 2
/dev/fd0 /mnt/floppy auto noauto,owner 0 0
LABEL=/s390 /s390 ext2 defaults 0 0
none /proc proc defaults 0 0
none /dev/pts devpts gid=5,mode=620 0 0
/dev/hda7 swap swap defaults 0 0
/dev/hdc1 /holding ext2 defaults 0 0
/dev/hdc2 /s391 ext2 defaults 0 0
```

### 6.6.2 Disk layout (AD system with two hard disks)

We arranged our ThinkPad hard disks this way:

```
Internal Hard Disk
  Linux root
  Linux swap partition
  /s390 partition
```
Z2RES1 3390-3 IPL volume
Z2RES2 3390-3 more system libraries
OS39M1 3390-3 z/OS VSAM data sets, spool, paging, and so forth
Z2USS1 3390-3 UNIX System Services HFS files
WORK02 3390-1 a local work volume
SARES1 3390-3 an emergency recovery system

Second Hard Disk /
/s391 partition
  WORK01 3390-1 another local work volume
  WORK03 3390-1 another local work volume
  Z2DIS1 3390-3 system DLIBs
  Z2DIS2 3390-3 system DLIBs
  Z2DIS3 3390-3 system DLIBs
  Z2DB21 3390-3 DB2
  Z2CIC1 3390-2 CICS
  Z2IMS1 3390-2 IMS
  Z2WAS1 3390-3 WebSphere
  Z2WAS2 3390-3 more WebSphere

This provides us with an IPLable system (and an emergency recovery system) even if the second disk is not installed.

We created two FLEX-ES definition files and two FLEX-ES startup shell scripts: one for operation with only the internal hard disk and one for operation with both hard disks. These files are listed in “Operational listings” on page 97.

6.7 FLEX-ES FakeTape on z/OS

FakeTape\(^{54}\) emulates tape devices using Linux disk files instead of tape drives. Provided the appropriate tape devices are defined in the z/OS IODF configuration data set, FakeTape will emulate any type of tape drive from 3420 to 3490-E. Because FakeTape always writes and reads the same format to/from Linux, it operates at the same speed for all different emulated tape device types. We ran several tape jobs using IEBGENER, IEBCOPY and DFDSS and they all performed well. Our definitions included a 3480 tape drive at address 560.

This is one of the jobs we executed:

```
//P390T  JOB 1,OGDEN,MSGCLASS=X
//BACKUP  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//SYSIN  DD DUMMY
//SYSUT1 DD DISP=SHR,DSN=SYS1.PROCLIB(JES2)
//SYSUT2 DD DSN=PROCLIB,DISP=(NEW,KEEP), 
//UNIT=560,LABEL=(1,SL),VOL=SER=222222
```

We submitted the job and the IEF233A mount request was issued by z/OS. The console interaction went like this:

```
$HASP373  P390T STARTED - INIT 1
IEF244I  P390T - UNABLE TO ALLOCATE 1 UNIT(S)
IEF877E  P390T NEEDS 1 UNIT(S) FOR SYSUT2
FOR SYSTUT2
FOR VOLUME 222222
OFFLINE
0560
02 IEF238D  P390T - REPLY DEVICE NAME OR ‘CANCEL
```

\(^{54}\) FakeTape is a trademark of Fundamental Software, Incorporated.
As seen in this interaction, we submitted our job. The tape drive was offline and z/OS requested that we provide a device. We answered with the device address, 560. We next received a mount request for volume 222222 (taken from our JCL) on unit 560. At this point, we switched to the CLI window (with the flexes prompt) and entered `mount 560 /tmp/222222`. This file name was quite arbitrary; if you plan to use many emulated tape volumes, you will probably want to create a directory for them (and perhaps a whole file system for them).

z/OS attempted to read the tape label (using an empty /tmp/222222 file) and received an error. This causes normal recovery processing for tape labels. We entered a volser (222222) in response to message IEC704A, and the job ran correctly. z/OS rewound and unloaded the “tape” at the end of the job. This logically unmounted the file and made the tape drive logically not ready.

We later read the tape and printed (to SYSPRINT) the tape contents. The process was about the same, except our tape file (/tmp/222222) now had a correct label and was immediately accepted after we entered another flexes> `mount 560 /tmp/222222` command.

Server files used by FakeTape are normal files in the server’s file systems. With a little planning—for space and naming—a whole tape library can be emulated using inexpensive server disks. A file (in the server file system) is equivalent to a tape volume. You can compress these files, ftp them, write them on CD-ROMs, and so forth, provided they are restored to their normal form when used again by FakeTape.

FakeTape can automatically recognize and handle data in AWSTAPE and AWSOMA formats, as produced by P/390-related systems. You should be able to exchange such files via ftp, although we did not try this. You can use the FLEX-ES utility `initawstape` to initialize a file in AWSTAPE format and then write output data to it.

Several special options are available for FakeTape resource definitions:

```plaintext
c3480: cu 3480
  options 'maxwritesize=200,allowdisconnects,allowmountccws,autoloader'
device(00) 3480 OFFLINE
end c3480
```

The maxwritesize option is important. It specifies (in megabytes) the maximum size of an emulated output tape. If your output approaches this size, FakeTape will signal an end-of-tape reflective marker. The S/390 program would usually write trailer labels at this point. In any case, FakeTape will write additional records but signal end-of-tape for every additional write request. If you do not specify a maxwritesize value, two exposures occur:

- You might consume all the free space in the server file system.
- You will never have end-of-volume processing (triggered by an emulated end-of-reel reflective marker).
In our opinion, you should always have a maxwritesize specification when using FakeTape. FLEX-ES 6.1.13 added the ability to modify device options using the FLEX-ES `mount` command. For example:

```
flexes> mount 560 /holding/tape/222222 'maxwritesize=50'
```

(The maxwritesize parameter also applies to DAT tapes. DAT drives do not signal that they are near the end of the tape (to permit trailer records to be written). They signal end-of-tape and nothing more can be written. A well-selected maxwritesize can overcome this limitation.)

The other options are more specialized and you may never need them with FakeTape.

### 6.8 Multi-system setup

You can define multiple S/390s by defining multiple FLEX-ES system definition files and compiling them. You can run multiple S/390 systems, one at a time, by simply creating multiple shell scripts (see “Building a shell script” on page 50) that name the appropriate *syscf* file in the *flexes* command. You can run multiple S/390 systems at the same time (*multiple instances*) if you have enough server memory and have defined your FLEX-ES resources appropriately.

Only one resource definition (*rescf* file) can be active on a server, so it must define all the resources needed by the S/390 instances. Each S/390 instance must be started with a unique system definition (*syscf* file). In this case you need to build and compile separate files for system and resource definitions. You would normally start a separate CLI window, with its *flexes* prompt, for each instance. Some care is needed to use the right *flexes* windows when controlling the S/390 instances. The CLI command `set prompt` can be used to change the *flexes* prompt to something more meaningful for each instance.

The redbook *NUMA-Q Enabled for S/390: Technical Introduction*, SG24-6215, describes the simultaneous operation of z/OS, VM/ESA, and VSE/ESA *instances*. A NUMA-Q system (now named x/Series 430 EFS) is larger than a typical EFS system, but the details for running multiple FLEX-ES instances are the same. (One exception is that, with more server processors, it is possible to use dedicated processors for *multiple instances* on the NUMA-Q machine, whereas this is not possible on a ThinkPad or small Netfinity.)

### 6.9 x3270 client

The FLEX-ES product distribution includes a copy of x3270. This is based on the freely distributable x3270 package, but with a few modifications. In general, it is intended only for use on the EFS system itself, under Linux. Users connecting from other platforms will probably use whatever TN3270e clients are available for their platforms. For this redbook project, we used IBM’s PCOM package—usually from OS/2 machines.

An x3270 window (on the Linux console) is started with the command:

```
$ x3270 -model 3 -keymap pc -port tn3270 localhost &
```

---

55 You can change all device options except DASD trackcachesize.
You can optionally identify the FLEX-ES terminal session you want to connect to like this:

```bash
$ x3270 -model 3 -keymap pc -port tn3270 localhost:t91f &
```

In this case, your resources definition (or a prior FLEX-ES `mount` command) must have defined a 3270 device named `t91f`. If you do not identify a specific terminal connection, the `x3270` session is connected to the Terminal Solicitor. In a FLEX-ES system, `port tn3270` normally resolves (via `/etc/services`) to port 24, the default port for the Terminal Solicitor.

The `x3270` command must be in the current search path, of course. It is in the `/usr/flexes/bin` directory, and we suggested placing this in the current PATH. The `model` parameter refers to 3270 models:

<table>
<thead>
<tr>
<th>Model</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>24 lines x 80 columns</td>
</tr>
<tr>
<td>3</td>
<td>32 lines x 80 columns</td>
</tr>
<tr>
<td>4</td>
<td>43 lines x 80 columns</td>
</tr>
<tr>
<td>5</td>
<td>27 lines x 132 columns</td>
</tr>
</tbody>
</table>

These are standard 3270 family model designations. Model 3 is perhaps the most common and that is what we used.

Important keyboard mapping for `x3270` includes:

<table>
<thead>
<tr>
<th>Function</th>
<th>Keys Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA1</td>
<td>Left-Alt + 1 (hold down the left Alt key and press 1)</td>
</tr>
<tr>
<td>PA2</td>
<td>Left-Alt + 2</td>
</tr>
<tr>
<td>Clear</td>
<td>Left-Alt + c or Left-Alt + Home</td>
</tr>
<tr>
<td>Enter</td>
<td>Enter</td>
</tr>
<tr>
<td>NewLine</td>
<td>Left-Alt + Enter</td>
</tr>
<tr>
<td>SysRq</td>
<td>Left-Alt + s</td>
</tr>
<tr>
<td>F13</td>
<td>Left-Alt + F1</td>
</tr>
<tr>
<td>EraseEOF</td>
<td>Left-Alt + f</td>
</tr>
<tr>
<td>Insert</td>
<td>Left-Alt + i</td>
</tr>
<tr>
<td>DelTab</td>
<td>Delete</td>
</tr>
<tr>
<td>BackTab</td>
<td>Shift + Tab</td>
</tr>
</tbody>
</table>

This key mapping may be disconcerting to experienced 3270 users (especially ISPF users) because it uses the large Enter key (on the PC keyboard) as the 3270 Enter key. On a “real” 3270, the large key in this position performs a New Line function, and a smaller key (where the Right Ctrl key is positioned on a PC keyboard) performs the 3270 Enter function. You can change `x3270` key assignments by the following:

```bash
# cd /usr/lib/X11/app-defaults
# cp X3270 X3270old               (keep copy of original, just in case)
# vi X3270                          
X3270.keymap.pc: \            (scroll until you find this line)
Meta<Key>d: Redraw()\n\nAlt<Key>d: Redraw()\n\nMeta<Key>Return: newline()\n\nAlt<Key>Return: newline()\n\n<Key>Return: newline()\n\n<Shift<Key>Tab: BackTab()\n
......
...... (near end of 'X3270.keymap.pc' section)
Alt<Key>Home: Clear()\n\n<Key>KP_Enter: Enter()\n\n<Key>Control_R: Enter()\n```

---

56 You must make certain that another `x3270` program does not appear first in the search path, as discussed in “X3270” on page 37. Current FLEX-ES releases do this automatically during installation.
<Key>Control_L:  \n\n(\text{insert this line})
:<Key>:  Default()
X3270.keymap.pc84:  

.....

You will need to use \texttt{:w!} to save the file in \texttt{vi}, because it is a read-only file. Make these changes carefully; there is one altered line and three new lines. Be careful not to make any other changes unless you understand the format and functions of this file.

We changed the large Enter key on the PC keyboard to perform the 3270 New Line function, the left Ctrl key to perform the 3270 Reset function, and the right Ctrl key to perform the 3270 Enter function. This is the most familiar keyboard arrangement for most 3270 users.

\section*{6.10 Remote resources}

FLEX-ES can use TCP/IP links to other instances of FLEX-ES on remote servers. You could, for example, run only the FLEX-ES resource manager on a remote server that has a large disk configuration. With the proper system definition (locally and resources definitions (locally and remotely), a FLEX-ES instance could access the remote disks (as emulated S/390 devices) as readily as local disks. When used with a fast LAN, the performance is not much slower than local device access. Usage is not limited to emulated disks; for example, a SCSI tape drive on a remote system could be used, or emulated channel-to-channel connections. A FLEX-ES license for each system is required.

Setup for this type of operation can be recognized by the keyword \texttt{network} in channel definitions and in \texttt{cu} definitions. FLEX-ES automatically uses TCP/IP port 555 for communication between multiple resource managers.

We did not use remote resources for our redbook projects. Our projects have concentrated on setting up and operating basic EFS systems, and we did not need remote resources for this. However, FLEX-ES remote resources is potentially a very powerful and useful function in appropriate situations.

A conceptual network is shown in Figure 6-1 on page 70. In this example, a single instance of FLEX-ES is executing (in system B in the center of the figure). It is using remote resources on the other two systems; disk resources on system A and a tape drive on system C. The FLEX-ES resource managers in all three systems communicate with each other. The emulated S/390 is unaware that some of the emulated devices are on remote systems.

System and device definitions are required on all three systems. These would include the following lines:

\textbf{System A:}
\begin{verbatim}
  adisk: cu 3990
  interface network(1)
\end{verbatim}

\textbf{System B:}
\begin{verbatim}
  channel(3) network
  channel(4) network
  cu devad... path(3) resource(adisk)
  cu devad... path(4) resource(atape)
\end{verbatim}

\textbf{System C:}
\begin{verbatim}
  atape: cu 3480
  interface network(1)
\end{verbatim}
6.11 Disk caches

FLEX-ES automatically caches 15 tracks of data for each emulated 3390 or 3380. You can adjust this in three ways:

- Specify a different number of tracks to cache for a particular emulated drive.
- Specify a different number of tracks to cache at the control unit level; excess tracks (above those needed for the specified or default cache for each device on the control unit) will float, as needed, among all the devices on the control unit.
- Use the `writethroughcache` parameter to force a different operation of the cache (on a device level). The default operation uses a writeback cache technique.

Here is an example that uses all three options:

```
(resources definitions)
....
c3990A: cu 3390
  interface local(1)
  options 'trackcachesize=150'
  device(00) 3390-3 /usr/flexes/links/A3s1
device(01) 3390-3 /usr/flexes/links/B3s1 devopt 'trackcachesize=5'
device(02) 3390-3 /usr/flexes/links/C3s1 devopt 'trackcachesize=45'
device(03) 3390-3 /usr/flexes/links/D3s1 devopt 'trackcachesize=30,writethroughcache'
device(04) 3390-1 OFFLINE devopt 'trackcachesize=3'
end c3990A
```

This is a bit complex. The five devices defined will ask for (15 + 5 + 45 + 30 + 3 =) 98 tracks of cache. (Device (00) does not specify a cache size and defaults to 15 tracks.) The control unit definition specifies 150 tracks of cache. This is (150 - 98 =) 52 more tracks than needed by individual device caches, and the 52 tracks will be a floating cache. The floating cache is managed by internal FLEX-ES logic. Each 3390 track is about 57 KB, so the 150 tracks of cache will require about 8.3 MB of server storage.
Cache is normally allocated for an offline device, since you might perform a FLEX-ES `mount` command to use the device. If you are certain you will not use the device you can specify a cache of three tracks. Three tracks is the minimum allowed cache size. (The minimum track cache size was stated as zero in earlier redbooks and earlier FLEX-ES documentation; this was incorrect.)

If you specify a control unit cache size of less than the sum of the individual device caches, the specified control unit cache size is ignored.

FLEX-ES defaults to `writeback` cache operation. This allows the S/390 disk write channel operation to complete when the data is in the FLEX-ES cache. The data will be flushed to the server disk at an indeterminate time in the future.\(^{57}\) An exposure exists if the system fails after a S/390 channel program thinks it completed a disk write, but the cache buffer has not really been written to disk. In this case, the S/390 program may have wrong state information. Such failures are extremely rare and the performance advantage of `writeback` is so great that this default operation is almost always used.

A \textit{writethrough} operation means that the S/390 channel operation for a disk write is not complete until the data is actually written to the server disk.\(^ {58}\) A copy of the data is retained in the FLEX-ES cache for possible future use. A \textit{writethrough} cache provides considerably lower performance than a \textit{writeback} cache, but it provides higher integrity.\(^ {59}\) It might be considered for a S/390 volume containing DB2 log data, for example.

If you have enough server memory, you can specify large disk caches for better overall system performance. There is obviously room for considerable tuning here, by manipulating cache sizes at the device and control unit level. You can use the \texttt{d ckd cachestats cuu} command to monitor cache effectiveness:

```
flexes> d ckd cachestats A80
ADDRESS  READS  WRITES  CACHE HITS  DEDICATED LINES  LINES USED
A80    2880   182      1811 (97%)          15         15 (25%)
A81    2880   182      1811 (97%)          15         15 (90%)
A82    2880   182      1811 (97%)          15         15 (45%)
```

The command can specify any device address on the control unit. The \textit{dedicated lines} column indicates the number of dedicated (non-floating) cache tracks. A \texttt{clear ckd cachestats cuu} command can be used to reset the statistics.

You can display the definition of an emulated disk with the \texttt{d devstate cuu} command:

```
flexes> d devstate A87
Filename: /s390/OS39HA State: OPEN, READY
Options: trackcachesize=30
```

\(^{57}\) This should be a familiar concept. The typical UNIX operation involves writeback caches, where the disk cache buffers are synched (flushed) to disk every 10 seconds or so.

\(^{58}\) This is not quite correct, because the RAID adapter also has a cache and the individual disk drives often have a buffer that performs a temporary cache function. We ignore these points in the current discussion.

\(^{59}\) In this case it is important to configure your system so that FLEX-ES uses a raw disk interface for emulated disk I/O. This permits FLEX-ES to completely control (except for hardware adapter buffering, if any) actual disk I/O. If normal Linux files are used for emulated volumes, data may still be in unwritten Linux buffers after FLEX-ES thinks it has written all the data.
6.12 Tuning cachesize

The FLEX-ES *cachesize* parameter (in the system definition section) specifies the number of S/390 bytes that should be reflected in the FLEX-ES instruction cache. FLEX-ES, in effect, compiles S/390 instructions into Pentium instructions in order to execute them. This involves overhead that can be reduced by saving the compiled instructions. This is the purpose of the instruction cache. The underlying structure is complex and proprietary to the FLEX-ES product. The instruction cache requires an average of 11 bytes for each S/390 instruction byte cached. A S/390 LA instruction, for example, requires 44 bytes in the cache.

You can monitor the effectiveness of the instruction cache with the `d cachestats` command:

```
flexes> d cachestats
Cache hits (ml): 1234567/6543321  55%
Cache hits (fb): 66554433/3344556678 20%
Cache hits (fbt): 98744/4567890 33%  
Cache misses: 5023567432/103543345876 5%        <== monitor this number
Cache compiles: 2345245/34567346 80%
```

The number of *cache misses* in this report is the critical information. If this number is above about 4%, you should increase your *cachesize* parameter. (But never increase it to the point where Linux starts paging!) If the number is considerably less than 4%, you might make better use of your server memory by increasing the defined S/390 memory size, defining more S/390 expanded memory, or increasing disk cache sizes. The CLI command `clear cachestats` will reset the statistics. The counters overflow, and you will probably need to reset the statistics before monitoring them.

The `d cachestats` command, by default, displays information for processor zero. If you have more than one processor enabled for S/390 emulation and want to display statistics for more than processor zero, you need to first issue a `set cpu` command:

```
flexes> set cpu 0 1  (display information for two processors)
flexes> d cachestats
```

FSI has not documented the meanings of the other statistics in this report.

6.13 Backup and restore considerations (ThinkPad)

Since a typical ThinkPad/EFS cannot directly connect to tape drives, the backup options are more limited than for a Netfinity/EFS system. Still, backing up and restoring S/390 data has interesting variations. The most basic element involved is where to store your backup data. There are several options:

- You can store S/390 backups on the ThinkPad disks, as Linux files. This provides logical backup of S/390 data, but does not provide protection against physical failure of disk drives.
- You can *ftp* back up data (in a variety of formats) to another system. This provides protection against physical system failures.

There are two general formats for backed up data:

- A file at the Linux level

---

60 A ThinkPad/EFS system might use remote FLEX-ES resources to connect to a machine that does have attached tape drives. This should work (although we did not try it), but cannot be considered a typical ThinkPad/EFS environment.
These files contain a complete emulated S/390 volume. It is easy to restore the complete volume, but quite messy to restore a single S/390 data set.

- A S/390 backup, using the z/OS program ADRDSSU or equivalent

This backup might be written to a FakeTape file. Once written, this file can be treated as a normal Linux file. It could, for example, be ftped elsewhere for safety. This type of backup requires the use of z/OS jobs and may be slower than purely Linux backup functions. It has the strong advantage that single S/390 data sets can be restored easily.

### 6.13.1 Using tar to back up S/390 volumes

An emulated S/390 volume, such as a 3390 volume, is typically a single Linux file. You can back up this volume simply by copying the Linux file somewhere else in your Linux file system. It would be advisable to copy it to another physical disk drive, if possible, for obvious reasons. Emulated volumes are large Linux files and a ThinkPad has limited disk space. Using a compression program as part of the backup process is attractive.

We performed a small test, as follows:

```bash
$ tar -cvzf /holding/OS39RA.tarz /s390/OS39RA
  note 1
$ ls -al /holding
  note 2
$ cd /s390
$ mkdir untar
  note 3
$ cd untar
$ tar -xzkwvf /holding/OS39RA.tarz
  note 4
extract 's390/OS39RA'? y
$ ls -al
$ cmp -s /s390/OS39RA /s390/untar/OS39RA & echo 'OK'
  note 5
```

1. We used a `tar` command to archive the file into a compressed tar file, which we named `OS39RA.tarz`. The `z` option in the `tar` command requests compression. The tar/compression step took about 12.6 minutes. The file was saved as a relative name (without the leading `/`). (The OS39RA volume is a 3390-3 volume from OS/390 V2R10.)

2. The `ls -al` command showed the resulting tar file was 468,091,705 bytes. This represents a 6.25:1 compression. Based on PC experience when building AD CD-ROM systems, we typically have approximately a 5:1 compression ratio when compressing files with the zip product for PCs. We have seen ratios from 3:1 to as high as 10:1.

3. We created a new directory, `untar`, so we could recover the file without destroying the original.

4. We changed to `/s390/untar` and issued another `tar` command to extract the file. The additional flags (k w) provided reassurance that we would not overlay our original file. The tar extraction placed the extracted file in the current directory (`/s390/untar`), but with the additional s390 directory name, and an `ls -al` command confirmed it was the correct size. The tar extraction took about 4.75 minutes.

5. We intended to use the `cmp` command to verify that the restored file matched the original file, but it appears that this command does not work with files larger than 2 GB.

The complete operation took place on a single HDD in the ThinkPad. Linux disk buffering was very effective and HDD seeking was not excessive during these operations, judging both from the sound of the HDD and observed performance.
The disadvantage of this backup method is that single S/390 data sets cannot be restored from the backup file. (Of course, you could restore the file containing the S/390 volume, rename the file, add it to your FLEX-ES definitions, IPL, clip the S/390 volume to a different volser, vary it online, and copy a single file from it. This is an extended process, but it does work for simple data sets. Attempting to deal with VSAM data sets, for example, could be more complex.)

We did not try this, but we could have first backed up the S/390 volume (with ADRDSSU) to create a FakeTape file and then processed this file with tar. This would require more processing and more Linux disk space, but this process could be used to more easily restore individual z/OS data sets.

6.13.2 Using tar and ftp

One way to back up S/390 volumes on a ThinkPad/EFS system is to ftp the Linux file containing the volume to another system. Using a 100 Mpbs LAN connection, especially on a private or local LAN, performance can be quite good. These files are large, by any standard, and ftp'ing compressed files probably makes more sense. We created the following shell scripts:

Shell script /usr/flexes/rundir/buOS39RAc

```
tar -cvzf /holding/OS39RA.tarz /s390/OS39RA
ftp -niv 0=buOS39RAp
```

Shell script /usr/flexes/rundir/buOS39RAp

```
open 10.20.30.50
user ogden xxxxxxx
bin
put /holding/OS39RA.tarz /s390/TP/OS39RA.tarz
```

We then executed these functions:

```
$ cd /usr/flexes/rundir
$ sh buOS39RAc
```

```
tar: Removing ‘/’ from member names
s390/OS39RA
Connected to 10.20.30.50
220 nfefs1 FTP server (Version wu-2.6.1-16) ready
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
331 Password required for ogden.
230 User ogden logged in/
200 Type set to I.
local: /holding/OS39RA.tarz remote: /s390/TP/OS39RA.tarz
226 Transfer complete
468091178 bytes sent in 57 seconds (8e+03 Kbytes/s)
221-You have transferred 468091178 bytes in 1 files.
221-Total traffic for this session was 468091705 bytes in 1 transfers.
221 Thank you for using the FTP server on nfefs1
```

The tar function again took 12.6 minutes and the ftp function took less than one minute. We used a private 100 Mpbs LAN connection, with no other active systems on it. The target system (our Netfinity/EFS machine) was otherwise idle during this time. It would have been faster to ftp /s390/OS39RA than to tar it, but we wanted the disk compression offered by tar. We created the /s390/TP directory on the target system before we ran the script.
We elected to use a separate tar file for each volume. A single, combined tar file would be very large and too unwieldy. We simply made more shell scripts; two for each S/390 volume we wanted to back up.

We could, of course, take ADRDSSU backups (to FakeTape files) and then tar and ftp these files. This might be the best solution, although it would require the most processing and ThinkPad disk space.

There are clearly several ways to approach backup and restore for a ThinkPad/EFS system. We cannot recommend any as the right way, and you will probably use a combination of techniques. We strongly suggest that you take time to understand the options available, try whatever is appropriate for you, and establish a routine for taking backups.

This discussion has not mentioned backing up the server operating system. Ideally, we should be able to capture the complete server state in a form that can be restored by a standalone boot program. The options are somewhat limited, since our ThinkPad/EFS systems did not have tape drives or CD-RW drives. One option would be to completely duplicate the primary ThinkPad hard disk contents onto another hard disk in the UltraBay. We did not have time to explore this problem during this redbook project, but hope to address it in a future project.

### 6.13.3 CD-RW drive

At least one of the base ThinkPads used for EFS systems has an optional CD-RW drive. This obviously can be used for backup purposes, working from Linux and taking full-volume backups. It would be necessary to compress the file containing the S/390 volume first, to fit on a CD. Also, some disk staging may be necessary if two hard disks are used for EFS operation and one of these must be removed to mount the CD-RW drive in the UltraBay.

### 6.14 Backup and restore considerations (Netfinity)

Back ing up and restoring S/390 data has more variations on an EFS machine than on a “real” S/390. You can, for example:

- Dump a S/390 disk volume to a channel-attached 3480/3490 drive, using a S/390 program. This has the advantage of using a universal medium (3480 or 3490) and being able to restore by data set. The disadvantages are that the media capacity is relatively small, and the drives are expensive (if you do not already have them).
- Dump a disk volume to an SCSI tape drive that emulates a 3480/3490, using a S/390 program. An advantage is that you can restore by data set. Within this category are:
  - 3480/3490-compatible tape drives. The advantages of these are that they are a universal media. The disadvantage is limited capacity of the media. See “Stand-alone z/OS restore” on page 81 for an example of such usage.
  - DLT drives. The advantage of these is that they are fast and have a large capacity. The disadvantage is that they are not a universal S/390 format.
  - 4mm drives. These have a large capacity, but lack the reliability of the DLT and 3480/3490 drives.
- Dump a disk volume to a FakeTape volume; that is, dump to a server file. If this is done using a S/390 program, you can restore by data set. An advantage is that it is fast. Disadvantages are heavy use of server disk space and the media cannot be carried off site. (You could consider FLEX-ES networking to overcome this last restriction.)
- Dump the server file (Linux file) that contains the S/390 volume, using a server program such as tar or dd. The advantage is that only simple server commands are needed; also, dump performance may be faster than anything under an emulated S/390 instance and the final result can be compressed. The disadvantages are that only a full S/390 volume
restore is possible; in addition, some staging space may be needed in the Server file system.

- A compressed `tar` file (tar.Z) is probably the most common way to do this.
- The resulting file can be left on a server disk, written to tape (any variety of SCSI drive), ftp’ed to another machine, and so forth.

The `ckdbbackup` and `ckdrestore` commands provided with FLEX-ES are most useful when using raw disks/devices (and especially if multiple raw disk slices are used for a single S/390 volume). The implementations discussed in this redbook do not use raw disks/devices, and these commands are not really required.

There are clearly many ways to approach backup and restore for a Netfinity/EFS system. We cannot recommend any as the right way, and you will probably use a combination of techniques. We strongly suggest that you take time to understand the options available, try whatever is appropriate for you, and establish a routine for taking backups.

This discussion has not mentioned backing up the server operating system. Ideally, we should be able to capture the complete server state in a form that can be restored by a standalone boot program. We did not have time to explore this problem during this redbook project, but hope to address it in a future project.

### 6.15 S/390 identification

The S/390 instruction STIDP stores an 8-byte field:

```
0APLSSSSIIII0000
```

```
||| | +------ the four zeros are constants
||| | +-------- machine type (see notes below)
||| +----------- arbitrary number set by manufacturer (serial number)
|+------------- LPAR number
+------------- processor number
```

The processor type field for an EFS system has one of these values:

- 1245 is the processor type for a system based on xSeries 430 or certain other uses
- 1247 is the processor type for a system obtained through PID
- 1246 is for all other EFS systems

The SSSS field is normally set to a unique value for each license and functions as a CPU serial number. The LPAR number is usually zero, but you can set it by a statement in the system definitions. The processor number corresponds to one of the processor numbers in your system definitions. For a single processor, this is normally zero.

In most cases, these fields are set by the FLEX-ES license key and you cannot change them. There is an option to make these fields changeable; this option must be specified when the system is ordered. If this option is used, the machine type cannot be set to 1245, 1246, or 1247. Two CLI instructions are associated with this area:

```
flexes> display cpuid
flexes> set cpuid             (only if permitted by the license key)
```

In general, the option to change the CPUID is limited to FSI Business Partners.
6.16 Display PSW and registers

You may sometimes need to display the S/390 PSW. You can do this from the *flexes* prompt provided by the CLI program:

```
flexes> d psw  (displays psw and next instruction, if possible)
flexes> d g    (displays general purpose registers)
```

6.17 RAS discussion

Is a ThinkPad/EFS or Netfinity/EFS system as reliable as a “real” S/390? No. A reliability discussion can quickly become extended and open-ended. We can summarize our views this way:

- A Netfinity/EFS or a ThinkPad/EFS system is not a S/390 (or a zSeries) system in terms of reliability. It does not have dual instruction units in each processor, or an Application Preservation function, for example.

- A Netfinity/EFS system is based on a very high-quality server and is expected to exhibit much better reliability than a typical desktop PC. Contributing factors are:
  - ECC memory
    This is extremely important in preventing the random lockups and crashes that many PCs experience.
  - Use of a RAID adapter
  - ECC memory in the RAID adapter, for the same reasons
  - Battery operated cache in the RAID adapter
  - Multiple, redundant power supplies
  - Avoidance of random applications (of uncertain trustworthiness) on the server
    (That is, games are not downloaded from random sources on the Web, we hope. The system should be used only for S/390 emulation.)
  - Thorough physical inspection of the Netfinity hardware by the business partner involved
    While this sounds trivial, it contributes to stability.
  - Using a UPS system “in front” of the server
  - Better cooling
    This is a significant factor for a system that is intended to run continuously.

- A ThinkPad/EFS is based on a quality laptop system, but this does not compete with a Netfinity/EFS server in many RAS characteristics.
  - The ThinkPad does not have RAID protection for its disk(s). With a single disk drive, or, at the most, two drives, there is no practical method to add RAID protection.
  - Multiple power supplies are not available.
  - ECC memory is not used, for a number of reasons.
  - Interfaces (such as SCSI or S/390 channels) are not typically available to ThinkPad/EFS users, making routine system backup functions more difficult.
  - Cooling effectiveness, especially when used without a docking station, can be greatly influenced by the exact location and positioning of the ThinkPad.

If you *must* have S/390 RAS, you need to buy a S/390. We believe a Netfinity/EFS system will provide more than adequate reliability within its place in the hierarchy of servers. A ThinkPad/EFS system is not intended as a production system and should not be compared with the RAS characteristics of production systems.
6.18 Verify ckd disk

The `chkckd` command can be used to verify the internal format of an emulated S/390 ckd disk:

```
# chkck -a /s391/WORK01  (Use your correct file name, of course)
```

The `-a` flag indicates that a full range of checks should be performed. Error messages from this utility should be taken seriously. However, the following message might be expected from volumes containing VM minidisks:

```
$ chkck -a /s390/OS39M1
FSIDU166 [cyl = 99 head = 14 rec = 1] Record cylinder number on “OS39M1”
does not match home address cylinder number (hacyl: 0x0063 reccyl: 0x0d70)
```

We also get these messages for some of the z/OS AD volumes, due to the use of VM systems during a step used when building new AD releases. FLEX-ES has been enhanced, starting with release 6.1-13, to better handle disks with improper formatting.

6.19 Installing FLEX-ES upgrades

Installing a new FLEX-ES release under Linux involves a few `rpm` commands:

```
# rpm -e ftlib  (remove existing FLEX-ES installation)
# rpm -e msgmgr
# rpm -e flexes
# cd /tmp
# rpm -i flexes-6.1-13.i386.rpm  (assumed location of new FLEX-ES rpms)
(a few messages appear)
# rpm -i msgmgr-6.1-13.i386.rpm
(a few messages appear)
# rpm -i ftlib-6.1-13.i386.rpm
```

(More rpm packages may be involved for Netfinity systems that use FLEX-ES channel adapters or ICA hardware. These are processed in the same way as the basic three rpm packages.)

The old FLEX-ES packages should be removed (with the `-e` option) in the order shown, and the new packages installed in the order shown. Your package names may be different (to reflect new release numbers). When removing the old packages, you can ignore a message about not removing the `/usr/flexes` directory.

This example does not indicate how the new FLEX-ES packages arrived in your `/tmp` directory. See “FLEX-ES installation” on page 34 for a discussion of CD and ftp delivery, and the use of `tar` to extract rpm packages.

Starting with FLEX-ES 6.1.15, you can use the `rpm -U` option for upgrades, instead of the separate `rpm -e` and `rpm -i` commands, provided you are upgrading from at least 6.1.13.

6.20 Remote operation

You can operate FLEX-ES and z/OS remotely. It is very easy to do and no special setup is required except for establishing a remote z/OS operator console. The key elements are these:

- Obtaining Linux command-line sessions. If you use the operational techniques described in this redbook, you need two windows: one for starting `resadm` (as root) and one for starting a shell script and then using `flexescli`.
Chapter 6. Additional topics

- Obtaining 3270 sessions. This is easiest to do using the Terminal Solicitor that is part of FLEX-ES.
- Obtaining access to a 3270 session that is defined as a z/OS operator console. Perhaps the easiest way to do this is to leave the normal z/OS console as is (this is address 700 in the AD system), and define another console at an address that appears in the Terminal Solicitor. (You need to change the CONSOLxx member in the z/OS PARMLIB to do this.)

There are many variations possible, but operation might be something like this:

- From a remote system, `telnet` to your server and log in as `flexes`. Then `su` to `root`. Enter your normal resadm startup command, such as `resadm -s R12A.rescf`.
- Start a TN3270 client session, directed to the IP address of your Linux server. It should display the Terminal Solicitor menu. Select the terminal session in the Terminal Solicitor that you have defined as a z/OS operator console. (We assume that you leave your normal z/OS operator session on the server screen, so you will have two z/OS operator terminals when you IPL. This is not required, of course.) Be certain to start the operator TN3270 session before IPLing z/OS.
- Make another `telnet` session to your server and log in as `flexes`. Execute your normal FLEX-ES startup shell script, for example, `sh sh12A`. This should return the normal `flexes` prompt.
- Enter your normal `ipl` command at the `flexes` prompt.

Again, there are many variations on this theme. You can easily operate with a single `telnet` session, for example.

### 6.21 ThinkPad power control

After installing Linux on our ThinkPad, we found that the `suspend` function (that is, closing the “lid” of the ThinkPad) worked correctly when Linux and FLEX-ES were running. However, if we left the system alone while running, with no keyboard or mouse movement for several hours, it automatically removed power and we needed to reboot Linux. We were unable to find any power controls in Linux that managed this timeout function.

We went into the ThinkPad BIOS (F1 after starting the ThinkPad) and changed all the power-related functions to “Disable.” This solved the problem and we could leave z/OS running overnight (or for weeks) with no problems. We did not attempt to analyze the exact use of each BIOS control. We needed to scroll in at least one of the BIOS screens to access all the controls.\(^{61}\)

### 6.22 Using the 4mm tape drive

We ran the same job we used for FakeTape (“FLEX-ES FakeTape on z/OS” on page 65) using a 4mm drive in our Netfinity/EFS system. Before submitting the job, we used this command:

```
flexes> mount 560 /dev/sg0
```

---

\(^{61}\) When we first installed the ThinkPad, we quickly checked the BIOS controls and thought we disabled any automatic power-off function. However, we failed to scroll through all the options and we had the automatic shutdown problem just mentioned.
in the flexes window. This assigns the 4mm drive to the FLEX-ES resource at S/390 address 560. (This address is appropriate in the AD system, where it is generated as a 3480 tape drive.) When the job was submitted, we received a mount request on the z/OS console, inserted a scratch 4mm tape, went through the normal standard label creation messages, and watched the job run. The 4mm tape was unloaded at the end of the job. (We later ran another job to read the tape, just to verify that it was correct.)

After creating a standard label 4mm tape on our EFS system running z/OS, we read the tape (without problems) on a 4mm drive attached to a P/390 system running z/OS.

6.23 Using an Overland T490E

We attached an Overland T490E SCSI tape drive to a SCSI PCI adapter in our Netfinity/EFS system. This is an older drive that was often used with P/390 systems. It can read 3480 and 3480 IDRC tapes (18 tracks) and 3490 tapes (36 tracks); it can also write all of these modes. Our Netfinity/EFS system also had a 4mm SCSI drive. Linux sensed the 4mm drive first and addressed it as /dev/sg0. The Overland drive was /dev/sg1.

We could use FLEX-ES mount commands to associate a tape drive with a FLEX-ES instance, as discussed in “Using the 4mm tape drive” on page 79, or we could hardcode the associations in the FLEX-ES definition files. We decided to hardcode the definitions. The z/OS system we used had an appropriate 3490E drive defined at address 500, and a 3480 drive at address 560.62 Our associated FLEX-ES definitions included:

```plaintext
system S12A:
    ......
cu devad(0x560,1) path(2) resource(CU3480)
cu devad(0x500,1) path(2) resource(CU3490)
    ......
end S12A

resources R12A:
    ......
CU3480: cu 3480
    interface local(1)
device(00) 3480 /dev/sg0                     (the 4mm drive)
end CU3480

    ......
CU3490: cu 3490
    interface local(1)
device(00) 3490-E /dev/sg1                   (the Overland drive)
end CU3490
    ......
end R12A
```

Both drives were online after we IPLed z/OS, and we had no problems using them.

The Overland drive is similar to the IBM 3490-F01 drive. Both have an extensive list of firmware options that can be set from the panel of the drive. We used the following options with the Overland drive:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Our setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>Power-up restart</td>
<td>01 = Unload cartridge</td>
</tr>
<tr>
<td>43</td>
<td>Format Display</td>
<td>01 = Enable</td>
</tr>
<tr>
<td>44</td>
<td>Write Synchronization</td>
<td>02 = Sync on 2nd consecutive file mark</td>
</tr>
</tbody>
</table>

62 We could use HCD to define or change these addresses, but we found it easier to adapt the device addresses to the existing IODF definitions.
6.24 SCSI adapter for the ThinkPad

Potentially, one might provide a SCSI interface to the ThinkPad and connect external SCSI tape drives. There are two ways this might be done:

- Use a third-party PCMCIA SCSI adapter
- Use a half-size PCI SCSI adapter in a ThinkPad docking station.

We did not try the PCMCIA solution, although some users have reported successful operation with these.

6.25 Stand-alone z/OS restore

We installed the Customized Offers Driver system (IBM product 5665-343, release 1.17). This is often known as the COD system and it replaces what was formerly known as the Starter System. It is a working z/OS system on two 3390-3 volumes. Only limited functions are available. The COD system is a sufficient base for installing a z/OS ServerPac. That is, if you have no other z/OS (or OS/390) system installed, you could first install the COD (using a stand-alone restore) and then use the COD to customize and install a ServerPac. This process is independent of any AD systems and is one of the startup processes that might be followed by new z/OS customers who are not using AD CD-ROM packages.

Although this section describes the installation of the COD system, the same general steps apply to installing backup tapes of any system. In essence, the COD package is simply a small z/OS system that has been backed up to tape (plus an additional tape containing two stand-alone utility programs).

Our copy of the COD included the following:

- Four tapes containing 3390-3 volume T9ESY1
- Five tapes containing 3390-3 volume T9ECAT
- One tape with stand-alone utilities

We first studied the COD documentation (IBM order number GI10-0615-04, or later) to determine what device addresses we could use. We found the following should work:
3390 disk drives at addresses 320, 321 (up through 33F, if needed)
3490 tape drive at address 390
3270 console for NIP and z/OS at address 0A1
3270 terminals for TSO at addresses 0F00-0F03

The COD has a large number of devices and addresses (“device numbers”) in its IODF and our choices were somewhat arbitrary. Based on our choices, we created appropriate FLEX-ES definitions and a startup shell script, shown in “Customized Offerings Driver definitions” on page 101.

Included with the COD system is a tape with two stand-alone utilities. A stand-alone utility is one that can be IPLed directly—without any operating system involved. The first utility on the tape is Device Support Facilities (DSF) that is used to initialize a disk volume. The second utility on the tape is DFSMSdss, which is used to restore a volume from tape to disk.

The general process we used is this:

- Use a FLEX-ES utility to create two 3390 volumes on our Netfinity/EFS system
- Compile the FLEX-ES definitions and start a FLEX-ES instance
- IPL and use DSF (from the stand-alone utility tape) to initialize the volumes
- IPL and use DFSMSdss (from the tape) to restore the COD tapes
- IPL the COD system

The detail steps were:

```
$ ckdfmt -n -r /s390/TEMP01 3390-3
$ ckdfmt -n -r /s390/TEMP02 3390-3
```

These steps create two new 3390 volumes for FLEX-ES. The file names and locations are arbitrary. Each file is about 2.8 GB.

```
$ cfcomp defCOD
... compile the FLEX-ES definitions
# resadm -T deactivate FLEX-ES resources that are running
# resadm -x RCOD activate the new resources
$ sh shCOD start a new FLEX-ES instance
```

The FLEX-ES definitions in file defCOD use the resource name RCOD and the system name SCOD. File shCOD is a shell script for starting an instance of FLEX-ES. The definitions and shell script are listed in “Customized Offerings Driver definitions” on page 101. The example here assumes that resadm was already running; the -T option deletes whatever FLEX-ES resources are currently operational. The -x option activates new FLEX-ES resources. (You could also completely kill resadm and restart it with the new resources; however, the technique shown here is the preferred method.)

At this point, we mounted the stand-alone utility tape in our Overland SCSI tape drive (see “Using an Overland T490E” on page 80 for details).

```
flexes> ipl 390
```

(After tape activity stops, press Enter on any of the 3270 sessions started by the shell script. Do not touch the tape drive. Do not rewind it.)

(The following messages are on the 3270 screen.)

```
CLEAR SCREEN WHEN READY (should appear on the 3270 screen)
```

63 If you are restoring some other system (other than the COD), you must determine the device numbers used by that system. That is, you must know the devices (and device numbers) included in the IODF that will be used when you IPL the new system. You must find this information from external sources; there is no easy way to determine it by examining the tapes.
(Alt-c is the default X3270 combination for clearing the screen)
(You should receive a number of title messages from DSF)

ICK006E DEFINE INPUT DEVICE, REPLY ‘DDDD,CCUU’ or ‘CONSOLE’
(simply press Enter)

ICK006E DEFINE OUTPUT DEVICE, REPLY ‘DDDD,CCUU’ or ‘CONSOLE’
(simply press Enter)

ENTER INPUT/COMMAND:
init unit(320) nvfy devtyp(3390) volid(D9ESY1) nomap
U
(reply U to initialize the volume)

ENTER INPUT/COMMAND:
init unit(321) nvfy devtyp(3390) volid(D9ECAT) nomap
U
(reply U to initialize the volume)

The two init commands you entered perform the volume initialization for the 3390 volumes.
At this point, switch to the window with the flexes prompt and IPL the tape again. If you did not alter it after IPLing the first time, the tape should already be positioned such that the second utility will be loaded.

flexes> ipl 390
This will produce startup messages from DFSMSdss (on the 3270 session) that are very similar to the startup messages from DSF. Respond to them in the same way. Remove the tape from the drive and load the first tape to be restored. This would be tape 1 or 4 for volume D9ESY1 in our example.

ENTER INPUT/COMMAND:
restore fromdev(tape) fromaddr(390) toaddr(320) verify(D9ESY1)
(the tape should be read and you should see disk activity as the volume is restored.)
(The tape will be unloaded automatically after it is read, and a message to load the next tape will appear on the screen.)
(Load tapes when requested. Also, clear the screen when requested because the utility waits until you clear the screen before continuing operation.)
(After the last tape you should see the message FUNCTION COMPLETE HIGHEST CONDITION CODE WAS 0.)
(You can now restore the second volume. Load the first tape of the second disk volume.)

restore fromdev(tape) fromaddr(390) toaddr(321) verify(D9ECAT)
(Follow the same process for all the tapes of this volume.)

The COD system is now ready to use.

flexes> ipl 320
(no IPL parameter is required)

You should see the SPECIFY SYSTEM PARAMETERS message on the 3270 session at address 0A1. The COD documentation provides the specific startup steps for its version of z/OS.

---

64 By Enter, we mean the 3270 Enter function. You may have customized your X3270 sessions to move this function to the right-hand Ctrl key. By default, it is the large Enter key on the PC keyboard.
The same general process (and stand-alone utilities) can be used to restore dumps of any z/OS or OS/390 system. The stand-alone utilities will adapt themselves to any device addresses. However, the devices in your FLEX-ES definitions should match the addresses known in the z/OS IOCDS in order to IPL a system after it is restored.

6.26 FSI Channel Adapters

The following FSI hardware adapters are available for Netfinity/EFS systems, but are not supported for ThinkPad/EFS systems:

- Parallel Channel Adapter (single channel and three channel versions)
- ESCON Channel Adapter (single channel and four channel versions)
- ICA Adapter

The required FLEX-ES packages (modules) to use these adapters are not provided with ThinkPad/EFS systems. There are a number of reasons for this, including the following:

- ThinkPad/EFS systems are not intended for production use. Different users will define production in different ways, and we will not attempt to make a rigorous definition here. Supporting these adapters could place ThinkPad/EFS in a potential production environment, and this has too many exposures (especially in the RAS area).
- These adapters would require docking stations that could accept full-length PCI adapters. None of the current docking stations intended for the supported ThinkPad models have this capability.
- Extensive testing would be necessary to verify PCI bus functions with the FSI adapter cards in this environment. This has not been done.

6.26.1 Parallel Channel Adapter (PCA)

This information was not available at the time of writing. (FSI has supplied and supported parallel channel adapters for FLEX-ES systems based on SCO UNIX for several years. They will be supported for Linux-based systems, but this work was not complete at the time of writing.)

6.26.2 ESCON Channel Adapter (PCA)

This information was not available at the time of writing. (FSI has announced their intention of selling and supporting ESCON channel adapters, but this work was not complete at the time of writing.)

6.27 Positioning with other small S/390s

Recent entry-level S/390 machines used by PID members have included the following:

- P/390, R/390, and S/390 Integrated Server (IS) systems (all based on P/390 adapters). These systems are no longer manufactured or marketed, but are still widely used by members of the IBM Partners in Development (PID) organization.
- Application StarterPak systems. These are no longer manufactured or marketed, but a number are in use by PID members.
- Several models of Multiprise 3000 systems (MP3000). These are currently manufactured and marketed and widely used.
The following table offers a few initial comparisons:

<table>
<thead>
<tr>
<th>S/390 CPUs</th>
<th>Rel CPU Perf</th>
<th>Typical S/390 Storage in MB</th>
<th>Internal Disk Perform</th>
<th>Typical Internal Disk Capacity</th>
<th>S/390 Chans</th>
<th>Significant Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThinkPad/EFS</td>
<td>&lt; 1</td>
<td>256</td>
<td>Good</td>
<td>32 GB +</td>
<td>None</td>
<td>Very portable!</td>
</tr>
<tr>
<td>Netfinity/EFS (small)</td>
<td>1</td>
<td>1.8</td>
<td>512+</td>
<td>Very good to excellent</td>
<td>160 GB</td>
<td>Possible</td>
</tr>
<tr>
<td>Netfinity/EFS (large)</td>
<td>3</td>
<td>5</td>
<td>2000</td>
<td>Very good to excellent</td>
<td>360 GB+</td>
<td>Several possible</td>
</tr>
<tr>
<td>P/390 R/390 Int. Serv.</td>
<td>1</td>
<td>1</td>
<td>256 fixed</td>
<td>Limited</td>
<td>18-36 GB</td>
<td>Very limited</td>
</tr>
<tr>
<td>MP3000-P30</td>
<td>1</td>
<td>8</td>
<td>1024 fixed</td>
<td>Very good</td>
<td>72 GB</td>
<td>Excellent</td>
</tr>
<tr>
<td>StarterPak model A10</td>
<td>1</td>
<td>4+</td>
<td>1024 fixed</td>
<td>Good</td>
<td>90 GB</td>
<td>4 ESCON</td>
</tr>
</tbody>
</table>

This table is intended only as a starting point for positioning EFS systems and needs a number of qualifications:

- The relative CPU performance numbers shown are rough indications of processor performance. Total system performance is dependent on many other factors as well, including memory size and disk performance. Also, processor performance depends on the nature of the workload and may vary considerably from the indicated ratios. Do not attempt to use this table for anything other than positioning the EFS models listed relative to the other specific machines shown.
- ThinkPad/EFS is shown as less than 1 CPU. This indicates that the single PC processor in the ThinkPad must be shared between Linux processes (such as S/390 emulated I/O) and S/390 instruction emulation.
- Many of these machines can have a wide range of memory installed. The numbers shown are intended to represent typical systems.
- All the machines except the StarterPak can have a wide range of disk storage installed. Again, the numbers shown are intended to represent typical installations. The numbers represent effective disk capacity, after RAID or mirroring overhead.
- The S/390 Chans column refers to S/390 channel connectivity, using ESCON and parallel channels. All of the systems offer some channel connectivity, but all except the MP3000 are relatively limited in this area.
- This table ignores RAS characteristics. The PC-based machines (Netfinity/EFS, ThinkPad/EFS and all the P/390-based machines) are designed to different RAS standards than the more “industrial strength” MP3000 and StarterPak systems. This may be an important factor when selecting a system for essential 24x7 production requirements.

A key element, not shown in the table, is the availability of 64-bit operation. Of the systems shown here, only EFS systems (both ThinkPad and Netfinity) will provide 64-bit operation. MP3000, P/390, and StarterPak systems do not (and will not) provide this capability.

---

65 64-bit operation for FLEX-ES was not available at the time of writing. FSI has commitments to provide such support.
6.28 Useful Linux commands

Not all EFS owners have Linux or UNIX backgrounds, and a few brief notes may help these users.

Common commands
New users may find Linux commands a little confusing. The following is intended as a brief reminder of the syntax for a number of commands often used while working with Linux in an EFS environment.

- `chown flexes /s390/volA` make user id “flexes” owner of this file
- `chgrp flexes /s390/volA` make group “flexes” the group owner of this file
- `chown aaa:bbb /home/myfile` make aaa the owner userid and bbb the owner groupid
- `chmod 600 /s390/volA` allow only file owner to read & write this file
- `cat filename` list all of a file on the terminal
- `more filename` list a file, page by page (space bar to advance page) (Ctrl-z to exit from the program)
- `less filename` a newer version of `more`
- `cp /s390/volA /s391/volX` copy first file to second file
- `find / -name unzip` start search in root and find file named “unzip”
- `updatedb` update the database used by the `locate` command (root)
- `locate unzip` locate all files with “unzip” in their name
- `file /usr/bin/unzip` describe nature of named file
- `mkdir /home/ogden/source` create a new directory
- `ls -al` list current directory, in some detail
- `ls -lrt` list current directory, most recently updated files last
- `mv oldfile newfile` rename or move a file
- `rm filename` delete a file
- `head filename` list first 10 lines of file
- `tail filename` list last 10 lines of file
- `tar -cvxf mytar.tar /flexes/rundir` create tar file from files in /flexes/rundir
- `tar -tvxf mytar.tar` list files contained in “my.tar”
- `tar -xvxf mytar.tar` extract all the files in the tar file
- `unzip /dev/cdrom/os390/os39ra.zip -d/s390` unzip a file on CD-ROM
- `mdir a:` display DOS diskette
- `mcopy /flexes/x/y a:x.txt` copy Linux file to a DOS diskette file
- `mdel a:afile.txt` delete a file on diskette
- `mformat a:` format a DOS diskette
- `shutdown -h now` shut down the Linux system
- `last` who has logged into Linux?
- `w` who is logged into Linux now?
- `whoami` displays your userid
- `ifconfig` display current LAN adapter status
- `netstat -r` display current routing state
- `route add default gw 9.12.17.150 eth0` create default gateway
- `route add -net 10.10.0.0 255.255.255.0 eth0` add network route
- `top` monitor activity. “q” to quit
- `vmstat 10 5` 5 activity reports. 10 seconds between reports.
mount /dev/cdrom /mnt/cdrom  mount a CD-ROM
mount /dev/fd0 /mnt/floppy  mount a diskette (but not a DOS diskette)
mount /dev/hdc2 /s391  mount file system on second HDD at mount point /s391
mount  display the mount table

Multiple consoles, sessions, screens
Linux offers multiple consoles on the PC (or ThinkPad) display. The keys Ctrl-Alt-Fn are used to select a console. Ctrl-Alt-F1 through Ctrl-Alt-F6 produce traditional command lines, and Ctrl-Alt-F7 produces the X windows GUI desktop. Each of these consoles is a separate Linux session, requiring a separate login. You can log in with different userids in each session, or use the same userid in several sessions. You can switch between sessions at any time, using Ctrl-Alt-Fn keys.

You can also have multiple command line windows open on the desktop. These do not require separate logins.

Text editors
Traditional UNIX users and many Linux users use vi as their primary text editor. Traditional mainframe users often avoid vi, if possible. The kde desktop offers the Advanced Editor, which can be started from the fountain pen nib icon in the toolbar. Gnome offers gedit. These are both much like a simple PC text editor and very unlike vi. They are suitable for almost all the text editing we required while installing and customizing FLEX-ES. Anyone comfortable using any version of Windows or OS/2 should have no trouble using these editors. The Red Hat Linux distribution includes a number of editors, and you can select the ones you like best.
Frequently Asked Questions

Q: I sometimes need to connect a USB device to my ThinkPad, but the FSI dongle is using the USB port. Can I disconnect it? When?
A: You can disconnect it when FLEX-ES is not running. Linux appears to sense the insertion or removal of the dongle and reacts appropriately.

Q: Can I disconnect the dongle after FLEX-ES is started?
A: We tried this. FLEX-ES continued to run for about a minute and then terminated. It appears that the dongle is sensed at irregular intervals.

Q: If I install a new release of FLEX-ES (perhaps by downloading it), do I need a new license key? A new dongle?
A: You need a new license key file, but can use your existing dongle.

Q: Can I install a boot manager and have Linux (with FLEX-ES) and Windows on the same ThinkPad?
A: Yes, but you will need to do the installation yourself. It is likely to involve a complete reinstallation of Windows. As an alternative, you might consider completely removing the primary hard disk and replacing it with a disk containing whatever version of Windows you prefer. However, the primary hard disk (as opposed to a second hard disk on a ThinkPad) is not intended for frequent removal.

Q: Does each S/390 user need a Linux userid?
A: No. A typical z/OS TSO user, for example, never “sees” Linux.

Q: Must I use vi to administer the Linux system?
A: No. Linux has a number of other editors, some of which are very “PC-like.” However, you may find it easier to use vi to edit files whose names begin with a period.

Q: Why do I need to define emulated control units and channels? I did not need to do this with emulated I/O on P/390 and MP3000 systems. For these machines, I could simply define emulated devices.
A: There are two major reasons. The first is that working through (emulated) channels and devices provides a more accurate emulation of S/390 hardware. In most cases this has little practical effect. The second reason involves emulating multiple S/390 instances, and sharing
emulated I/O--for example, shared DASD. This is a more complex situation that does not exist on a P/390 and is not permitted (via LPARs) on an MP3000. The emulated control units provide the logical interlock and queueing points for sharing devices according to S/390 architectural specifications.

Q: How real is the channel emulation? Can I make my system faster by defining lots of emulated channels and spreading out my I/O devices?
A: As best we could tell, the emulated channels are not very real in this sense. Many system definitions we examined define only two or three channels of each channel type. You may want to define separate channels for emulated 3174 or 3274 control units.

Q: I understand that a LAN MAC address is used to control FLEX-ES licenses for Netfinity Servers. Can I use the same method for ThinkPads?
A: No. FLEX-ES licenses for ThinkPad/EFS and Netfinity/EFS (using Linux) require the use of a dongle. Any FLEX-ES version that uses Linux as the base operating system will require a dongle.

Q: Can I run a ThinkPad in 24x7 operation? (24 hours/day, 7 days/week)
A: There is no rule against it, and some ThinkPads are left running in docking stations. However, in our opinion, this is not a reasonable solution for an EFS system. The fully-loaded ThinkPad used for EFS develops considerable heat (on the bottom) and may require additional air flow. In addition, a system intended for 24x7 S/390 operation should have RAID disks.

Q: Can I use token ring LAN adapters? You mention only Ethernet.
A: Yes, token ring can be used. We usually discuss Ethernet because (1) it is what we used, and (2) the majority of EFS users will probably have Ethernet LANs.

Q: How many Ethernet adapters should I have for the ThinkPad or Netfinity?
A: One should be sufficient, and will probably be the mini-PCI adapter for the ThinkPad. You can share this adapter (using multiple IP addresses) between Linux and S/390 operating systems.

Q: Can I use the FSI channel adapters or ICA adapter with a ThinkPad? I could put them in a docking station.
A: No. This is not supported.

Q: Can I have “odd size” 3390 drives, such as a 100 cylinder 3390 on a P/390?
A: No. Only standard size drives are emulated at this time.

Q: An emulated 3390-3 was 3339 cylinders on my P/390. Why is it 3343 cylinders under FLEX-ES?
A: A standard 3390-3 has 3339 cylinders of directly usable space. The additional cylinders include the “CE cylinder” (for testing) and alternate tracks. P/390 emulation did not include these additional cylinders, while FLEX-ES emulation does include them. In normal use, you will not see any difference between the two methods.

Q: Are 4mm tapes compatible between my MP3000 and my Netfinity/EFS system?
A: Yes, they should be, for those EFS systems that support 4mm drives.

Q: How many emulated S/390 devices can I have?
A: There is a FLEX-ES maximum of 2048 emulated devices per S/390 instance. If you emulated two S/390s, each will have this maximum number of devices.
Q: I have a fixed amount of real memory in my system. You say that I should not overcommit this for S/390 emulation. Should I make my emulated S/390 central storage as large as possible (without triggering Linux swapping) or should I define some expanded storage at the expense of central storage?
A: Provided you do not exceed the maximum 2 GB permitted for central storage, we think that server memory is better used as S/390 central storage than as S/390 expanded storage. However, you might have specific applications that require expanded storage, so we cannot provide an absolute answer. You face the same quandary when dividing up real memory on an MP3000, for example, between central storage, expanded storage, and a disk cache.

Q: Several ThinkPad processor speeds are available. The model you used for this redbook (750 Mhz) is one of the slowest. Can I use a faster processor?
A: Yes.

Q: Can I write FakeTape files from my Linux applications?
A: Yes. The format is well documented. APIs are provided if you work under Linux.

Q: How many x3270 sessions or TN3270 sessions can I have on the Linux desktop?
A: There is no particular limit. Reasonable screen usage might indicate a practical maximum of something like three or four.

Q: Should I use Terminal Solicitor sessions (via Linux TCP/IP) or direct z/OS TCP/IP connections for my TSO and CICS users?
A: Assuming z/OS TCP/IP is configured and working, it does not appear to make much difference. z/OS TCP/IP connects TN3270 sessions to VTAM and whether VTAM is using a local, non-SNA 3270 (as emulated by the Terminal Solicitor) or working through z/OS TCP/IP has no observable effect at the end-user level. Systems programmers will probably prefer to work through the Terminal Solicitor because it is operational even if z/OS TCP/IP is not working.

Q: When I connect via z/OS TCP/IP, I do not get the terminal logo I always receive if I connect via the Terminal Solicitor. Why not?
A: This may be due to your z/OS TCP/IP setup. The z/OS TCP/IP PROFILE needs to point to a USSTAB module that generates the logo. Recent AD releases have this set up.

Q: Can I connect to z/OS with an ASCII telnet session? I want to use vi.
A: Yes, although this is not really an EFS question. Recent AD releases have port 1023 (of z/OS TCP/IP) configured for ASCII telnet sessions. If you are not using an AD release, you will need to configure z/OS TCP/IP.

Q: I have the impression there are frequent FLEX-ES releases. Do I receive these automatically? When should I upgrade? Where is there a concise list of what is new in each release?
A: In general, releases are obtained through the IBM business partner from which you purchased the system (for PID members). We suggest you should upgrade only when a later release contains a feature you need. Remember that you will need a new license key file for a new release.

Q: Where can I order a printed copy of the FLEX-ES manuals?
A: For PID members, FLEX-ES manuals can be obtained from the IBM business partner from which you purchased the system.

Q: Can I use the system for other purposes when FLEX-ES is not running? Can I run Linux server functions on it when FLEX-ES is running?
A: When FLEX-ES is not running, you can run anything you want on your system—taking care not to disturb the FLEX-ES customization. When FLEX-ES is running, you could still run other Linux workload. However, we generally recommend against this. The external
workload may force Linux swapping and this can have negative effects on FLEX-ES performance. Having gone to considerable effort to obtain a small S/390 (in the form of your EFS system), we suggest you use it only as a S/390. If you need to also run a Linux server, we suggest you obtain another machine.

Q: Should I purchase an Uninterruptible Power Supply (UPS) for use with my EFS machine?  
A: The ThinkPad battery provides, in effect, a UPS function. You might consider an external surge protector. For a Netfinity/EFS system, we believe a UPS is a good idea.

Q: How do I order an EFS system? What if I am not a PID member?  
A: You do not need to be a PID member to order an EFS system (although you will not have access to AD CD-ROM systems). The FSI Web site has a list of their business partners.

Q: Where can I find documentation about the various FLEX-ES traces?  
A: There are several FLEX-ES traces:

- The processor event trace is always active and uses a permanently defined circular buffer. It is intended only for FLEX-ES debugging and is not documented.
- The processor instruction trace uses the tracesize buffer you define in your system definition. It is intended for FLEX-ES debugging but might be useful to more sophisticated customers. It is minimally documented at present, but FSI intends to provide better documentation.
- Device traces, which are usually obtained with the devtrsnap command. These are not documented at present, but a user familiar with channel programming techniques might find these traces useful.

The traces are not well documented at this time.

Q: Are any modifications, PTFs, or SPEs required for z/OS on EFS?  
A: No.

Q: Are printer FCBs emulated?  
A: Yes. FLEX-ES printer emulation supports the Load Forms Control Buffer channel command. The default FCB provides 66 lines at 6 lines/inch, with a “channel 1 punch” at line 3 and a “channel 12 punch” at line 64.

Q: Is a default 1403 carriage tape emulated?  
A: Yes. It is set for 66 lines at 6 lines/inch, and has a channel 1 “punch” at line 3. There is no way to alter these settings.

Q: Are Token Ring and Ethernet equally supported?  
A: Yes, but remember that a Linux patch may be needed to use token ring.

Q: Can I manage an ESCON director from an EFS system?  
A: Not from a ThinkPad/EFS system! At the time of writing, the FSI ESCON adapters were not yet available and we were unable to address this question.

Q: I am interested in using FLEX-ES remote resources for disks and tapes. Do I need a FLEX-ES license just to run the resource manager on remote systems?  
A: Yes, you will need a FLEX-ES license for every system running the resource manager.

Q: If I obtain a faster server, do I need a new license? Do I need to notify someone?  
A: PID members should ask their business partner for a new license key file. (You must do this because your old key may not work with the faster processor.) You do not need to purchase a new license, assuming you have retained the same dongle. FSI does not notify IBM or any other software vendor. You need to manage any potential S/390 software fee changes related to the increased MIPS of your new machine. This is obviously a complex area, and these processes may change in the future.
Q: What is the difference between swapping and paging?
A: None, in these discussions.

Q: You mention MIPS in the 10-12 range for ThinkPad/EFS; business partners talk about 16-20. Which is correct?
A: Both and neither. An emulated S/390 has a much wider range of MIPS than a “real” hardware S/390. Any MIPS measurement is very strongly dependent on the instruction mix and the addressing mix involved, as well as the emulation techniques used. We could probably contrive code to produce anything from below one MIPS to 30+ MIPS. We try to use conservative numbers in this series of redbooks, while marketing representatives tend to use optimistic numbers. An exact MIPS number simply does not exist.

Q: How is floating point performance?
A: It is acceptable for casual use, but you would probably not select this platform for major floating point applications. Binary (IEEE) floating point has considerably better performance than hex floating point.

Q: There are always new Linux functions available, as well as kernel fixes and upgrades. Should we install these? Do we need to reinstall any part of FLEX-ES afterward?
A: If you have a stable system, we suggest you do not apply Linux changes. You should regard your EFS system as a S/390 and not as a Linux machine. If you do rebuild the Linux kernel, you will need to reinstall the FLEX-ES msgmgr package and you will need to make a special effort to make the USB dongle interface work.

Q: How can I tell whether my Ethernet connection is running at 10 or 100 Mbps?
A: We did not find a convenient way to do this at the PC end. Many Ethernet hubs have an LED that indicates 100 Mbps operation and you might look for this.

Q: Can I use DHCP to obtain a LAN IP address?
A: Yes, you could do this for Linux TCP/IP, but not for z/OS TCP/IP. However, this is probably not a practical solution. Your EFS system is essentially a S/390 server, not a client, and you will probably need a stable IP address for it.

Q: How difficult is it to change the IP address of my system?
A: You are likely to have two IP addresses: one for Linux and one for z/OS. To change the z/OS address you must edit the PROFILE data set and then restart TCPIP. If you are sharing an Ethernet adapter with Linux, you will also need to change the IP address in the FLEX-ES resource stanza that defines the emulated 3172. To change the Linux IP address:

(gnome): Programs --> System --> Network Configuration. Select the Interfaces tab. Deactivate the appropriate interface. Edit the interface to change the IP address. Activate the interface. Save. Quit.

Q: As a developer, I sometimes need a real hardcopy listing. You provide very little information about setting up printers. Why?
A: We did not have time to experiment with printers while writing this redbook. This can become a complex topic quickly when network printers are considered. Larger EFS systems often have channel-attached S/390 printers and no special documentation has been required to cover these.

Q: How long does the ThinkPad battery last?
A: With our fully loaded T20 system (512 MB, two disk drives, heavy computational load) the battery usually lasted about 45 minutes.

Q: Is SNA available over Ethernet?
A: Yes.
Q: Can I use a locally administered MAC address for Ethernet?
A: We have never done this, although it appears that some Ethernet drivers appear to support this option.

Q: How many Ethernet adapters do I need if I want to use SNA and TCP/IP under z/OS?
A: One.

Q: You did not say much about the ICA adapter. Can I emulate EP lines (or PEP lines)?
A: We did not say much about this FSI adapter because z/OS does not support ICA devices. No, you cannot use it for EP (or EP under PEP) lines. This mode is not emulated.

Q: My Netfinity has two processors, with one licensed for S/390 emulation. Can I also license the second processor for S/390 emulation? Will this make a faster system? How much faster?
A: Yes, you can license the second processor for S/390 emulation. You cannot dedicate it for this purpose, because this would leave no processor for other required Linux functions and auxiliary FLEX-ES processes. You need to balance the cost of the additional license against the performance increase you might see from partial use of the second processor.

You might run two S/390 instances, one with a dedicated processor, but this is probably not what you are trying to accomplish. You could have one dedicated processor and one non-dedicated processor for a single S/390 instance--however, this is not a good performance option and we recommend against it. You could use two processors enabled for S/390, with no dedicated option. This is the most likely method and, given processors in the 1 GHz (or faster) range, this will definitely create a faster system. We do not have specific performance data for this situation.

Remember that enabling a second processor for S/390 work changes the MIPS rating of your machine and may affect the license costs for much of your S/390 software.

Q: Why is a dedicated processor faster? What makes it different?
A: A complete context switch for a processor creates additional overhead for the “hardware” of the system. The processor cache contents (L1 and L2, typically) become useless and many cache misses occur for a while. Likewise the virtual address translation mechanism must be refreshed with new address ranges. A processor dedicated to S/390 emulation will have fewer context switches and avoid this overhead. We have no specific measurements on current hardware as to how much performance is gained by dedicated processors.

Q: Does all the FLEX-ES work occur only on the licensed number of processors? Does it ever use the additional processor(s)?
A: All the S/390 instruction emulation is done only on the licensed number of Pentium processors. Some of the I/O and control processes run as normal UNIX processes and will be dispatched on the “other” Pentium processors.

Q: Can I buy a larger Netfinity server, with four processors, and provide faster S/390 emulation?
A: Yes. Under current circumstances a maximum of three processors may be enabled for S/390 emulation.

Q: I have an older S/390 and am interested in changing to a Netfinity/EFS machine. I currently use 7 channels on my machine. Do I need this many channels on the Netfinity/EFS system?
A: Probably not, but this takes careful analysis. In this redbook, we assume that your DASD will be emulated on internal server disks. (Indeed, nothing else was supported at the time of writing. However, FSI plans to support external CKD DASD attachment through both parallel
and ESCON channels.) Even if external DASD can be connected, the internal emulated DASD is probably faster and larger. This topic (channel analysis) is highly specific to each situation and we suggest you work with a knowledgeable business partner who is trained on Netfinity/EFS configurations.

Q: Can an ESCON adapter (on an EFS system) go through an ESCON director? How is it configured for director port addresses?
A: The FSI ESCON adapter was not available at the time of writing, but we expect that it will be usable through ESCON directors.

Q: How many Parallel Channel Adapters can I have?
A: The limitation is usually the number of PCI slots available in your machine. A secondary limitation may be with cable connectors. There is no reasonable limit from the software viewpoint.

Q: I am interested in a substantial number of parallel channels, say fifteen, on my Netfinity/EFS system. How are cables handled? Is this a problem?
A: This would require five FSI parallel channel adapter cards, with three channels per card. FSI provides cables as part of the adapter package. These cable have small connectors on one end (for the adapter card) and standard bus and tag connectors on the other end. There are two basic issues: "finger space" at the back of the Netfinity becomes very tight when several of these adapters are used, and strain relief for the cables is extremely important. You must plan your cable routing and physical support. FSI can provide a strain relief structure (made from plexiglass) or you could construct your own. In any event, this number of cables cannot simply hang from the back of the Netfinity. You should talk with your business partner about your plans.

Q: How much electrical power does your Netfinity/EFS system use? How much does it cost to run all the time?
A: This will vary with system configurations, of course. We informally measured the current used by our system. Once started, it averages about 2.0 amps (at 120 volts) when the disks are active and the flat panel console display is on. When the system is idle and the console is off, it uses about 1.5 amps. Assuming a power factor near 1, this is a range of 180 - 240 watts. At typical U.S. electricity prices, it will cost about $.60 (60 cents) per day to run the system. You should provide power connections that meet the published requirements for your server, and you should realize that startup currents are higher than the average running currents mentioned here. Also, remember that a CRT display (instead of the flat panel we used) may substantially increase the power required.

Q: I do not understand the purpose of the /holding file system you recommend. Comments?
A: It is simply an additional work area. We attempt to have minimal file creation, expansion, or deletion in the /s390 file system to avoid potential disk fragmentation. We use the root file system and /holding for work areas and ignore fragmentation in these areas. In particular, we tend to use /holding as a temporary area for tar work; for example, to tar a S/390 volume (with compression) before ftp'ing it somewhere for backup.

Q: Why do you make the root file system so large? Why not make separate file systems for /usr and /tmp?
A: We like to have ample work space. Again, the goal is to minimize activity in /s390 (and /s391, if it exists) that might lead to disk fragmentation. You can create separate file systems for /usr and /tmp (and anything else you want) while installing Linux. If the sole purpose of the system is to run S/390 emulation, there is no particular need for separate file systems (partitions) for various traditional parts of Linux.
However, we note that you might improve long-term stability by making a separate file system for /tmp and/or /var. If some logging function (assuming it writes to /tmp) fills up /tmp, the system may crash. It is less likely to crash if /tmp is a separate file system. We have not had any such problems, but it could be a consideration for a production operation.

Q: Are you a little paranoid about disk fragmentation? I have been told that Linux file systems have very little fragmentation.
A: Perhaps. Fragmentation can certainly slow down a Windows, OS/2, or UNIX system and we simply want to avoid this possibility.

Q: Can I upload from diskettes using the INDSFILE function?
A: Yes. We used PCOM to log onto TSO (from another PC). We then used PCOM’s SEND and RECEIVE commands at the PC. If you upload/download using x3270, you must ensure that the Linux file system involved is mounted. If it is diskette, you could mount -t msdos /dev/fd0 /mnt/floppy.

Q: Your examples always assign a Linux file name that is the same as the volser of the S/390 volume in that file. I used a different convention, but now I forget which volser is stored in which Linux file. How can I easily check the logical content of a file?
A: Assume you have an emulated 3390 volume in /my/linux/files/AAA and you want to determine the volser. Use the ckddump command:

```
$ ckddump /my/linux/files/AAA 0 0 | more
```

and scroll to the third record. This will be a standard VOL1 label and the volser starts in byte five. The ckddump command produces a storage dump type of listing (in hex and characters) and includes all the CKD fields of the volume. A Ctrl-C keyboard function will terminate ckddump output. If you want to see only the volser, use the following:

```
$ ckddump /my/linux/files/AAA 0 0 | grep VOL1
```

Q: You said that sysplex is not supported. Can I run basic sysplex, using shared DASD for coupling data sets?
A: No. This is not supported and does not work at this time. Monoplex operation works, but sharing coupling data sets between multiple S/390 instances does not work. (Somewhat unusual CCW programs are used (by z/OS) for controlling shared access to coupling data sets and these are not supported at this time.)

Q: Can I run z/OS under z/VM? Can I emulate sysplex functions under z/VM?
A: Yes and yes, to the extent that z/VM emulates various sysplex and CF functions.

Q: When will the Linux kernel fix the multiprocessor and token ring problems?
A: We do not know. Red Hat 7.2 uses a 2.4.7 kernel and requires patching. Red Hat 7.3 (not available for FLEX-ES at the time of writing) uses a 2.4.18 kernel. We understand that the patches are not included in that level, but will probably be included in the next level after that.

Q: Using the installation processes described in this redbook (and not using raw devices for emulated volumes), it seems that the AD CD-ROMs install faster if the target files do not already exist. That is, if I do not overwrite existing emulated volumes. Is this normal?
A: We have noticed the same effect. It might be faster to delete the existing volumes and create new ones (using the unzip and ckdconvaws piped commands described in “Unzip and convert a volume” on page 43). New AD systems are not installed often enough to worry much about the installation speed, so we usually overwrite old volumes (for example, rename Z2RES1 to Z3RES1 and overwrite it) and ignore any performance issues.
Operational listings

This appendix lists the FLEX-ES definitions and shell scripts we used when installing our EFS systems and writing this redbook. These files were copied from our working systems and represent the actual controls and scripts we used.

These examples are from our ThinkPad/EFS systems. Our initial Netfinity/EFS definitions are exactly the same, except that all emulated S/390 volumes are in the /s390 file system. (For the ThinkPad, they are split between /s390 and /s391.)

Basic definitions for a single HDD

Our system and resources definitions (both in the same file) used for the four volumes required to IPL the z/OS Release 1.2 AD CD-ROM system were as follows:

```plaintext
system S12A:
  memsize(262144)
  cachessize(2048)
  essize(64)
  instset(esa)
  cpu(0)
  channel(0) local
  channel(1) local
  channel(2) local
  cu devad(0xA80,8) path(2) resource(CU3990)
  cu devad(0xA90,4) path(2) resource(CU3991)
  cu devad(0x700,5) path(0) resource(CU3174)
  cu devad(0xE20,2) path(1) resource(CU3172)
  cu devad(0x560,1) path(2) resource(CU3480)
end S12A

resources R12A:
  CU3990: cu 3990
  interface local(1)
  device(00) 3390-3 /s390/Z2RES1
  device(01) 3390-3 /s390/Z2RES2
  device(02) 3390-3 /s390/OS39M1
  device(03) 3390-3 OFFLINE
```
Some FLEX-ES users elect to define emulated 3270 devices as OFFLINE and then use FLEX-ES mount commands (in a shell script) to define and name terminals. For our small system, we found it easier to simply define a number of 3270 terminals (with names such as M701) in our resources definition.

The S/390 volumes used for our initial IPL are normally mounted at addresses A80, A81, A82, and A87 (for the four system volumes) and A91 (WORK02) and A93 (SARES1). The “gap” between A82 and A87 required the definition of several OFFLINE 3390 volumes. (In this case, the device model (3390-1, -2, -3) is not meaningful.) Likewise, the “missing” devices in the A90 address range require OFFLINE devices.

It would be possible, of course, to mount these volumes at any 3390 addresses contained in the z/OS IODF. We could have placed the A87 volume on A83 and avoided defining the OFFLINE devices, for example. However, other AD CD-ROM documentation (including several Redbooks) uses the addresses we selected and we want to preserve compatibility with this existing documentation. This is the only reason for inserting OFFLINE devices to manipulate the addresses.

The tape definition is OFFLINE. This means that FLEX-ES mount commands are needed before the device can be used. This is typical, and allows the UNIX files to be associated with such devices to be selected and changed while the z/OS system is operational.
We strongly suggest that new EFS users study this example until every detail is understood. It is a simple example, and contains nothing clever or unusual.

Shell script for a single HDD

After using `cfcomp` to compile the above definitions, we used this shell script to start operation. (A `resadm -s R12A.rescf` command must be issued first.)

```
flexes S12A.syscf
xmodmap -e 'keysym Alt_L = Alt_L Meta_L'
xset fp+ /usr/flexes/fonts
xset fp rehash
echo 'Verify your master console is active'
x3270 -model 3 -keymap pc -port tn3270 localhost:mstcon &
x3270 -model 3 -keymap pc -port tn3270 localhost:M701 &
flexescli localhost S12A
```

Definitions for two HDDs

The following definitions assume that the second ThinkPad hard disk is mounted and contains the optional volumes discussed in the text. Since these definitions reference files on the second hard disk (with mount point `/s391`), they can be used only when the second hard disk is installed (and the file system mounted). Note that the system name is now S12B and the resources name is R12B.

```
system S12B:
memsize(262144)
cachesize(2048)
essize(64)
istset(esa)
cpu(0)
channel(0) local
channel(1) local
channel(2) local
cu devad(0xA80,12) path(2) resource(CU3990)
cu devad(0xA90,4) path(2) resource(CU3991)
cu devad(0x700,20) path(0) resource(CU3174)
cu devad(0xE20,2) path(1) resource(CU3172)
cu devad(0x560,1) path(2) resource(CU3480)
end S12B

resources R12B:
CU3990: cu 3990
interface local(1)
device(00) 3390-3 /s390/Z2RES1
device(01) 3390-3 /s390/Z2RES2
device(02) 3390-3 /s390/OS39M1
device(03) 3390-3 /s391/Z2DB21
device(04) 3390-2 OFFLINE
device(05) 3390-3 /s391/Z2DIS1
device(06) 3390-3 /s391/Z2DIS2
device(07) 3390-3 /s390/US39S1
device(08) 3390-3 /s391/Z2DIS3
device(09) 3390-2 /s391/Z2IMS1
device(10) 3390-3 /s391/Z2WAS1
device(11) 3390-3 /s391/Z2WAS2
end CU3990
```
Address A84 would normally contain volume Z2CIC1 (if the complete z/OS AD 1.2 system is loaded). There was a problem using this volume with FLEX-ES releases before 6.1.13, and we placed this address OFFLINE in the definitions.

Shell script for two HDDs

This shell script is just like the first one, except that it uses system name S12B instead of S12A.

```
flexes S12B.syscf
xmodmap -e 'keysym Alt_L = Alt_L Meta_L'
xset fp+ /usr/flexes/fonts
xset fp rehash
echo 'Verify your master console is active'
```
Customized Offerings Driver definitions

We used the following FLEX-ES definitions and shell script with the Customized Offerings Driver.

```plaintext
system SCOD:
  memsize(262144)
  cachesize(2048)
  instset(esa)
  cpu(0)
  channel(0) local
  channel(1) local
  channel(2) local
  cu devad(0x320,2) path(2) resource(CU3990)
  cu devad(0xF00,3) path(0) resource(CU3174)
  cu devad(0xA1,1) path(0) resource(CU3175)
  cu devad(0x390,1) path(2) resource(CU3490)
end SCOD

resources RCOD:
CU3990: cu 3990
  interface local(1)
  device(00) 3390-3 /s390/TEMP01
  device(01) 3390-3 /s390/TEMP02
end CU3990

CU3174: cu 3174
  interface local(1)
  device(00) 3278 LF00
  device(01) 3278 LF01
  device(02) 3278 LF02
end CU3174

CU3175: cu 3174
  interface local(1)
  device(00) 3178 mstcon
end CU3175

CU3490: cu 3490
  interface local(1)
  device(00) 3490-E /dev/sg1
end CU3490

end RCOD
```

The following is the shell script we used to start the system.

```plaintext
flexes SCOD.syscf
xmodmap -e 'keysym Alt_L = Alt_L Meta_L'
xset fp+ /usr/flexes/fonts
xset fp rehash
echo 'Verify your master console is active'
x3270 -model 3 -keymap pc -port tn3270 localhost:LF00 &
x3270 -model 3 -keymap pc -port tn3270 localhost:mstcon &
```
flexescli localhost SCOD
FLEX-ES definitions and commands

This appendix reviews a selected subset of the FLEX-ES parameters needed to define emulated S/390 systems and their resources. The formal FLEX-ES documentation should be reviewed for additional information, more emulated devices, and many other parameters. The last section also discusses basic CLI commands that can be used at the *flexes* prompt.

The definitions in this section are only for discussion. See Appendix A, “Operational listings” on page 97 for listings of our working definitions.

FLEX-ES definitions must be exactly correct or they will not work. Take special care with colons, parentheses, and hex numbers.

System definitions

System definitions are usually something like this:

```plaintext
system S10A:
  memsize(262144) # K Central Storage = 256 MB
  cachesize(2048)  # M Expanded Storage = 64 MB
  essize(64)
  instset(esa)
  tracesize(256)
  feature lpar
  cpu(0) dedicated # Single Pentium, dedicated
  channel(0) local
  channel(1) local
  channel(2) local
  channel(3) local
  cu devad(0x00C,3) path(0) resource(R10A2821) # 2450R, 2540P, 1403
  cu devad(0x560,1) path(0) resource(R10A3480) # tape drive (4mm)
  cu devad(0x700,16) path(1) resource(R10A3174) # 3270 terminals
  cu devad(0xA80,12) path(2) resource(R10A3990) # 3390 disks
  cu devad(0xE20,2) path(3) resource(R10A3088) # CTCs for TCP/IP
end S10A
```

© Copyright IBM Corp. 2002
The first line (system S10A) provides a name for the system being defined. The FLEX-ES compiler will create a file named, for this example, S10A.syscf.

memsize always expresses kilobytes of storage. The largest accepted value is 2097152, which would define a 2 GB system. The size specified will be the size of central storage in the emulated S/390. There is no direct relation to the memory size on the server, although we strongly recommend that memsize be such that no paging is required by the underlying Linux system.

The parameter essize specifies expanded storage size and expresses megabytes of storage. Furthermore, the number specified must be an even multiple of 16. If essize is not specified, then no expanded storage is emulated. A specification of 64 means 64 MB of expanded storage.

The instset parameter should always be specified as instset(esa) for the systems we are discussing.

The cachesize parameter specifies the amount of storage to be used by FLEX-ES to cache dynamically compiled S/390 instructions. This parameter expresses kilobytes of S/390 storage. (The server storage required averages 11 times the specified S/390 storage value.) Practical values are in the range 1024 to 8192. For a Thinkpad/EFS system, 2048 appears to be a practical value; a larger Netfinity/EFS system might use 4096. We suggest you do not use values outside the recommended range unless you are clearly in an experimental mood. A specification of 2048 means 22 MB of server storage will be used by FLEX-ES for a S/390 instruction cache.

The tracesize parameter specifies the number of entries in a FLEX-ES trace table. We suggest 512 as a reasonable number. The FLEX-ES instruction trace function runs only when started. There is no processor overhead caused by defining a tracesize parameter. This is a somewhat imprecise parameter because the size of each entry in the trace table is not well defined. In any event, the total memory used (with the recommended value) is fairly trivial.

There is also a processor event trace that uses a permanently defined buffer. This event trace is always active and is not associated with the tracesize parameter.

The feature lpar statement causes the LPAR bit to be set in the scpinfo response. If VM detects this bit, it will not use active wait. (Active wait means that VM loops waiting for work.) This feature is most meaningful if you are using VM, and not using a dedicated processor.

The cpu parameters indicate the number of emulated S/390 CPUs and assign identification numbers to each one. Multiple cpu parameters are used to specify multiple emulated S/390 CPUs within a single emulated S/390. For example,

```
cpu(0)
cpu(1)
cpu(2)
```

as part of a system definition would specify an emulated S/390 with three CPUs. The CPU numbers (which are defined S/390 architecture elements) are 0, 1, and 2. The number of CPUs specified may not exceed the number of server processors enabled for S/390 emulation. For a ThinkPad/EFS system, this will be one. (The single CPU may be given any number in the range 0 - 7, but there is no reason to use anything other than 0.) For a dual-processor Netfinity, you might have one or two CPUs defined, depending on how many processors are specified in your FLEX-ES license.
The **dedicated** keyword on a CPU definition statement means that the underlying ThinkPad processor will be used only for S/390 emulation; it will not be used for Linux work. This option cannot be used with a ThinkPad/EFS system because only one processor is available for both S/390 and other Linux work. Netfinity/EFS systems will normally have at least two server processors and all but one will be dedicated to S/390 emulation (unless you are running multiple S/390 instances; in that case, the situation is more complex).

The **channel** parameters specify emulated channels and assign channel numbers. For example:

```plaintext
channel(3) local
channel(1) local
```

defines two channels, with the channel numbers 3 and 1. The channel numbers must be unique, but need not be in sequence. The **local** keyword means the channels are emulated block multiplexor channels and are emulated using resources on your server. Other types of channels may be specified, such as byte multiplexor and channels using remote (via TCP/IP) resources. These are described in detail in the FSI documentation.

The **cu** statement defines an emulated control unit. The **devad** parameter defines the beginning address and the number of (emulated) devices connected to the control unit. This address (or device number) is the address seen by the operating system. For example, if you want to “IPL from A82,” then address A82 would appear in a devad parameter. The **path** parameter specifies an emulated channel connected to the control unit. The **resource** parameter specifies the name of a section in a resource definition (described below) that defines the emulated devices connected to the control unit. Resource names, used in **cu** statements, must be unique.

The **cu** definition specifies a number of units for each **cu**. You **must** define this number of devices in the matching resource definition. For example:

```plaintext
cu devad(0xA80,12) path(2) resource(R10A3990)
```

specifies that 12 devices are attached to this control unit. The R10A3990 clause in the resources definition used with this system definition must contain 12 devices.

An **end** statement is required for the system definition, and the parameter should specify the same name as the **system** statement.

### Resource definitions

Resource definitions refer to actual devices or files and are defined separately from system definitions. In practice, both a system definition and the corresponding resource definition might be in the same text file. Both must be “compiled” by the FLEX-ES **cfcomp** program, to produce syscf and rescf files, before they can be used.

---

66. Other types of channels may be specified, such as byte multiplexor and channels using remote (via TCP/IP) resources. These are described in detail in the FSI documentation.

67. For this example, we might have **devad(0xA80,8)** as the parameter. This defines 8 addresses, A80, A81, A82, and so forth. The A82 name is implied in this sequence.
Emulated control unit types

FLEX-ES emulates many control unit types (cutype parameters) and device types; the full documentation should be consulted for a complete list. The commonly used cutypes include:

- 3990: DASD control unit (for all 3380 and 3390 devices)
- 3480 and 3490: Tape control unit (for FakeTape and SCSI tape drives)
- 3174: Terminal control unit (for emulated locally-attached 3270s)
- 3172: LAN control unit (for Ethernet LANs)
- 3172TR: LAN control unit (for token ring LANs)
- 3215: Typewriter system console (mostly for VM and Linux for S/390)
- 2821: Unit record control unit (for emulated card reader/punch, printer)
- ctc: Channel-to-channel control unit

Emulated device types

FLEX-ES emulates many device types, and these are documented in the full FLEX-ES materials. The following list includes commonly used devices for an EFS system running current S/390 operating systems.

- 3390-1: 3390 model 1, with 1117 cylinders (1113 usable cylinders)
- 3390-2: 3390 model 2, with 2230 cylinders (2226 usable cylinders)
- 3390-3: 3390 model 3, with 3343 cylinders (3339 usable cylinders)
- 3390-9: 3390 model 9, with 10038 cylinders (10017 usable cylinders)
- 3172: LCS device, for Ethernet connections
- 3172TR: LCS device, for token ring connections
- 3215: Typewriter console (not for OS/390 or z/OS)
- 3178: Most typical 3270 definition
- 3420: “Round” tape, emulated with SCSI device or FakeTape
- 3480: Emulated with SCSI device or FakeTape
- 3490: Emulated with SCSI device or FakeTape
- 3490-E: Emulated with SCSI device or FakeTape
- 1403: Printer, emulated with UNIX lp functions
- 2540R: Card reader, emulated with UNIX files
- CTC: Channel-to-channel adapter, emulated with UNIX sockets

Typical resource definitions

A simple set of resource definitions, corresponding to the system definitions above, might be:

```
resources R10A:
  mema: memory
    264
  end mema

R10A2821: cu 2821
  interface local(1)
  device(00) 2540R OFFLINE
  device(01) 2540P OFFLINE
  device(02) 1403 OFFLINE
  end R10A2821

R10A3480: cu 3480
  interface local(1)
  device(00) 3480 OFFLINE
  end R10A3480

R10A3174: cu 3174
  interface local(1)
  device(00) 3278 mstcon
```
device(01) 3278 payroll1
device(02) 3278 develop2
device(03) 3278 OFFLINE
....
device(15) 3278 sysprogs
end R10A3174

R10A3990: cu 3990
  interface local(1)
  device(00) 3390-1 /s390/.....
  device(01) 3390-3 /s390/.....
  device(02) 3390-3 OFFLINE
  ....
  device(10) 3390-3 /s390/..... devopt *writethroughcache,trackcachesize=50*
  device(11) 3390-1 /s390/.....
end R10A3990

R10A3088: cu 3172
  interface local(1)
  options *ipaddress=9.12.17.211*
  device(00) 3172 eth0
  device(01) 3172 OFFLINE
end R10A3088

end R10A

A resources block must have a name; R10A is used in this example. Each set of resources in the block must also have a name. Each set name should match a name in a cu statement in the system definition. The general syntax for defining a resource set is:

\[
\text{resourcename: cu cutype parameters end resourcename}
\]

You assign a resource name. It must follow the FLEX-ES rules for names. The name is followed by a colon and the keyword cu. The cutype must be one of the cutypes known to FLEX-ES. This is followed by whatever parameters are needed to specify the particular device(s) being defined. The keyword end is then followed by the resource name.

The memory definition is needed only if your system contains an FSI parallel channel adapter (either model) and you are running under UnixWare or OpenUNIX 8. The parameter value is normally 1.03*memsize, where the memsize is specified in the system definition.68 You will need to consult the full FLEX-ES documentation to understand server memory management when Parallel Channel Adapters are used. If you do not have a Parallel Channel Adapter you should omit the memory definition in the resources block. Do not use this statement with a Linux-based system.

In the following paragraphs, we show examples of commonly used device resource definitions and describe parameters used for each device type.

**CKD disk resources**

R10A3990: cu 3990
  interface local(1)
  device(00) 3390-1 /s390/.....
  device(01) 3390-3 /s390/.....
  device(02) 3390-3 OFFLINE
  ....

---

68 If multiple S/390 instances are used, the situation is more complex. The maximum memory parameter is the smaller of 512 MB or 3/8 of the server real memory size.
device(10) 3390-3 /s390/..... devopt 'writethroughcache,trackcachesize=50'
device(11) 3390-1 /s390/.....
end R10A3390

The interface local(1) parameter means that the defined devices are local (on the same server running FLEX-ES) and that there is one path (channel + control unit) pointing to these devices. Multiple paths are typically found when running multiple S/390 instances (that is, emulating multiple S/390 systems at the same time) and sharing control units among the multiple instances. Shared DASD is the most common example of this.

The device(00) parameter describes the first device of this resource set. The index number is decimal; it must begin with 00 and be incremented by one for each additional device. If your device addressing scheme (at the z/OS level) requires addressing gaps, you must define dummy (OFFLINE) devices in the resource set. The addresses seen at the z/OS level are set by the cu statement (in the systems definition section) that points to this resource set.

The device type (3390-1 and 3390-3 in this example) must be from the list of known FLEX-ES emulated device types. (These are listed in “Emulated device types” on page 106.) The next parameter, /s390/xxxxx, is the Linux file that contains the emulated 3390 volume. This can be the actual name or a symbolic link.

The keyword devopt indicates that optional parameters follow. The writethroughcache parameter causes the FLEX-ES 3390 emulation program to consider a S/390 write operation complete when the data is actually written to disk. This is described in “Disk caches” on page 70.

The trackcachesize= parameter tells the emulation program to allocate the indicated number of track buffers (3390 tracks, in this example) for a cache. The default cache size is one emulated cylinder (15 tracks for a 3390) for each emulated S/390 disk volume. A larger cache size may improve performance, but at the expense of using more server memory. In general, the default cache size is acceptable.

3270 terminal resources

Emulated 3270 terminals, which are actually TN3270 sessions (or x3270 sessions) to a FLEX-ES server running under Linux and appear to the S/390 as local, channel-attached, non-SNA terminals, are defined as follows:

R10A3174: cu 3174
  interface local(1)
  device(00) 3278 mastcon
  device(01) 3278 term701
  device(02) 3278 term702
  device(03) 3278 OFFLINE
  device(04) 3278 @9.12.17.210
  device(15) 3278 term70F
end R10A3174

The interface, local, device, and device index parameters have already been described. The cutype is a provided FLEX-ES value and device type 3278 is a device type emulated by FLEX-ES. Each emulated 3270 is connected through the FLEX-ES TN3270 server. If the 3278 keyword is followed by @IP-address (such as @9.12.17.210 in the example), the FLEX-ES TN3270 server waits for a TN3270 connection from this IP address. If the 3278

---

69 FLEX-ES can use remote resources. For example, a separate UNIX server (connected via TCP/IP) could contain emulated 3390 volumes, but not contain an emulated S/390. We believe that new ThinkPad/EFS users are unlikely to use remote resources while getting started with EFS operation and we do not further describe remote resources in this redbook.
keyword is followed by the OFFLINE keyword, then a FLEX-ES `mount` command must be issued to assign either an IP address for the connection or a name for the Terminal Solicitor. If the 3278 keyword is followed by a name (such as `mstcon` or `term70F`), this name is added to the list of terminals available through the FLEX-ES Terminal Solicitor (which is described in “Building a shell script” on page 50.)

Typical mount commands, from a `flexes` prompt, might be:

```
flexes> mount 703 bills
flexes> mount 704 09.12.17.211
```

These commands would add a new terminal name (`bills`) to the Terminal Solicitor screen, and also enable a TN3270 connection from 9.12.17.211. Where do the addresses (703 and 704) come from? The `cu` statement in our example (“System definitions” on page 103) specified an addressing range with 16 addresses. The resource name in the `cu` statement matches the resource set name of the 3270 definitions, so these resources have addresses beginning with 700.

The same mount commands could be placed in a shell script used to start an emulated S/390. The shell statements would be:

```
$ echo 'mount 703 bills' | flexescli localhost S10A
$ echo 'mount 704 09.12.17.211' | flexescli localhost S10A
```

**Tape resources**

The tape resources discussed here involve emulated S/390 tapes. If you channel-attach “real” S/390 tapes on an EFS machine, they would be defined differently, as channel devices. An emulated tape might be a SCSI-attached device, such as a 4mm drive, a DLT drive, or an SCSI 3480/3490 unit. These are not normally used on ThinkPad/EFS systems, although such a system could use a FLEX-ES network channel to connect to another FLEX-ES system that has SCSI tape drives.

A SCSI tape drive might be defined as:

```
R10A3480: cu 3480
  interface local(1)
  device(00) 3480 /dev/sg0
end R10A3480
```

You can define FakeTape devices, and these would be the normal case for a ThinkPad/EFS system. The definition might be:

```
R10A3480: cu 3480
  interface local(1)
  device(00) 3480 /tmp/tapes/222222
end R10A3480
```

The most likely definition for FakeTape usage would be:

```
R10A3480: cu 3480
  interface local(1)
  device(00) 3480 OFFLINE
end R10A3480
```

The keyword 3480 specifies a standard FLEX-ES emulated device. The following parameter, `/tmp/tapes/222222`, is the Linux file name to be used for the emulated tape volume. If the definition is OFFLINE, then a `flexes mount` command can be used to temporarily assign a Linux file to the drive. FLEX-ES supports 3420, 3480, 3490, and 3490-E emulation. You should use the appropriate type to match whatever is defined in z/OS.
Several specialized parameters can be used with tape resource definitions. These would be written like this:

```
device(00) 3480 /tmp/tapes/222222 devopt 'maxwritesize=200'
```

The options are placed (in single quotes) after a `devopt` keyword. The `maxwritesize` option is probably the most commonly used. It specifies, in megabytes, the maximum size of the emulated tape media. When the amount of data written approaches this size, an end-of-tape reflective strip is emulated and the operating system would normally write EOV labels. This is especially important with FakeTape, to prevent a runaway program from filling the Linux file system with tape output.

### LAN resources

LAN resources are used to define z/OS TCP/IP connections to the S/390 or to define an SNA LAN connection to the S/390. A definition for z/OS TCP/IP use might be:

```
R10A3088: cu 3172
  interface local(1)
  device(00) 3172 eth0
  device(01) 3172 OFFLINE
end R10A3088
```

The `cu` type 3172 is a defined FLEX-ES keyword and defines an LCS-type LAN connection to S/390. (On the S/390 side, there would normally be two CTC addresses (even/odd addresses) defined. A LAN device for use by z/OS TCP/IP must be specified exactly as shown (except that the device name of the interface might be different). That is, it will be defined as two devices (00 and 01), with the second device OFFLINE. The IP address for this interface would be specified in the z/OS TCP/IP profile.

The device designation in this example, eth0, may appear strange to experienced UNIX users, who will want to write something like `/dev/eth0`. However, eth0 is the proper format.

An additional parameter may be required for a LAN definition. A “real” 3172 unit may have up to four adapters. These are numbered 0-3, and this number appears in the z/OS TCP/IP or VTAM specifications. By default, FLEX-ES uses number zero. If you want a different number, you must include an `options` statement:

```
R10A3088: cu 3172
  options 'adapternumber=1'
  interface local(1)
  device(00) 3172 eth0
  device(01) 3172 OFFLINE
end R10A3088
```

This emulates the “real” 3172 adapter number 1. This number is used in the z/OS TCP/IP PROFILE definition (and perhaps by VTAM). This number is not related to the actual LAN adapter number in your Linux. The actual PC LAN adapter that is used is specified by the device parameter, such as eth0, eth1, and so forth.

You can share a LAN adapter between Linux and z/OS TCP/IPs. In order to do this, you need to specify the IP address to be used by z/OS, as a parameter in the resource definition. (You should specify the same address in your z/OS TCP/IP parameters.) For example,

```
R10A3088: cu 3172
  options 'ipaddress=9.12.17.211'
  interface local(1)
  device(00) 3172 eth0
```

---

70 The same situation exists with a P/390 or MP3000 emulated I/O LAN adapter. It is known as the MPTS number in these cases.
Using this example, the single Ethernet adapter in our ThinkPad/EFS machine appears as IP address 9.12.17.210 (defined when we installed Linux) and as address 9.12.17.211 (when we start FLEX-ES). The single adapter responds to two different IP addresses.

**Cloned devices**

The FLEX-ES documentation includes an optional `cloned` parameter for many resource definitions. We did not use this parameter. This parameter applies to a limited set of devices—emulated printers and card readers are good examples—for which the resource definitions can be automatically replicated to serve multiple S/390 instances. This is meaningful only when multiple S/390 instances are used. It potentially saves a little time when creating resource definitions. We suggest it may be better to take the extra time to directly create all the resource definitions you need and ignore the cloned parameter.

**Compiled files**

FLEX-ES system definitions and resource definitions are compiled (with the `cfcomp` command) to produce `syscf` and `rescf` files. You can have separate source files for system and resource definitions, or combine them into one file (like we do in this redbook). You can have multiple system and/or resource definitions in a single file. For example:

```plaintext
system patti:
    system definitions
end patti

system billy:
    definitions for another system
end billy

system fran:
    definitions for yet another system
end fran

resources mixed:
    resource definitions
end mixed
```

If you compile this file with `cfcomp`, you will have four output files: `patti.syscf`, `billy.syscf`, `fran.syscf`, and `mixed.rescf`. The `syscf` files are potential operands for a `flexes` command, and the `rescf` file is a potential operand for the `resadm` command. We could produce exactly the same results by breaking this input file into four files and compiling each one separately.

Remember that you can have only one resource definition active (via the `resadm` command) on a server, but you can have multiple emulated S/390 systems active. Assuming we defined compatible systems and resources in this example, we could start the resources (`resadm -s mixed.rescf`) and then start three systems (`flexes patti.syscf`, `flexes billy.syscf`, `flexes fran.syscf`). For practical purposes, we would probably also want to start three interactive `flexescli` windows to control the three systems.
Common rules

FLEX-ES names, in general, may be up to 255 characters and use uppercase/lowercase letters, numeric digits, and a few special characters. These special characters are underscore, hyphen, dollar, at, and period. A name cannot consist of a valid decimal or hexadecimal number. The following are not valid names: 2, 123, 1403, 0X1, 7F, 7f, 1C2a30.

White space (usually blanks) may be included almost anywhere between words. Two special cases exist:

- No white space may exist in the parameters of an options statement or in a devopt parameter. For example,
  
  device(00) 3390-1 /S390/VOLAs1 devopt ‘trackcachesize = 10’
  
  is incorrect; there should not be spaces before or after the equal sign.

- If a device number range is used, there must be spaces before and after the hyphen. For example:

  device(00 - 15) 3278 OFFLINE

  is correct because there is a space before and after the hyphen.

FLEX-ES keywords cannot also be used as names in system or resources sections.

Numbers (in system and resource definitions) are always decimal. If you want a hexadecimal number, you must indicate it. For example:

  cu devad(0xA80,10) path(1) resource(xyz)

contains a hexadecimal number (0xA80) and a decimal number (10). CLI commands differ; they assume device addresses are hexadecimal.

You should not use duplicate names in any definitions.

The resadm command

The resadm command has a number of options that are frequently used. Using R10A.rescf as an example of a compiled resource definition, the options are:

```
# resadm -s R10A.rescf (start the resource manager)
# resadm -k (kill the resource manager)
# resadm -t cu:cu3990A (terminate an individual resource)
# resadm -T (terminate all resources)
$ resadm -r (list all active resources)
$ resadm [-h hostname] -n (list node names)
# resadm -x R10.rescf (refresh the resource definitions)
```

A few notes about the resource manager may be useful:

- The -r and -n options may be used by anyone. The other options are available only to root.
- The -r option is the most frequently used option.
- For all except the -s option, the resource manager is assumed to be running when these commands are issued.
- The -n option is for use when multiple servers are cooperatively running FLEX-ES resources. This environment is not described in this redbook.
- You should terminate resources gracefully, with -t or -T options, instead of killing them (with -k or Linux commands).
- A common sequence is to -T (terminate all resources), then -x (refresh with a newly compiled resource file), and then re-IPL a S/390 operating system.
You can -t (selectively terminate a resource) and then -x (refresh the resources file). This will refresh only resources that are not running at that time.

The resource manager writes log information to the /var/log/messages file (via the Linux syslog logging facility).

The resource manager automatically uses TCP/IP port 555 to communicate with other elements of FLEX-ES, whether there is a single server or there are multiple servers involved.

CLI commands

Every S/390 emulated system instance has an associated main console for FLEX-ES control commands. This is a virtual console that is not directly connected to a real terminal. The commands it processes are the CLI commands, some of which are described here. A program named flexescli is used to communicate with the virtual main console. flexescli works from a Linux command line, and you can start as many flexescli instances as you like (and work via FLEX-ES networking if you have multiple FLEX-ES servers). The flexescli program sends commands to a main console and returns the results.

In discussions we typically ignore the details of the flexescli and main console interaction and describe operations in terms of commands to flexescli. Also, for our basic EFS discussion, we usually assume that only a single S/390 is being emulated at any given time.

CLI commands can be entered two ways:

- You can start the flexescli program in interactive mode and then issue CLI commands directly at the flexes prompt provided by the CLI program.
- You can use ECHO to pipe a command to flexescli. In this case, it executes the command and terminates.

The syntax for the flexescli command is:

```
# flexescli IPname systemname
```

If the flexescli program is not in the current PATH, you would need to issue the full path name for it. The IPname can be an IP address, a host name that is resolved by DNS, or a local name (in /etc/hosts). The standard IPname localhost is used to reference the local IP loopback address. The systemname is the name of a syscf file used to start an emulated S/390. In our examples, this is S10A.syscf. (The syscf suffix is omitted for the flexescli command.) The flexescli program must always be directed to a specific S/390 instance (even if there is only one running).

An example of executing a single command through the CLI interface might be:

```
# echo 'ipl a80 0a8200' | flexescli localhost S10A
```

The flexescli program senses whether input is waiting for it in a pipe; if so, it executes the waiting command and quits. If started this way:

```
# flexescli localhost S10A
flexes>
```

it issues a flexes prompt and waits for commands. It will run until it receives a quit or shutdown command.
There are a considerable number of CLI commands. Many of them are for low-level S/390 hands-on operation (such as instruction stepping, inspecting registers and memory, and so forth). These are described in detail in FSIMM210: CLI Language Reference, from FSI. We describe only a few of the higher-level commands here. We assume you are in flexescli interactive mode for these examples:

flexes> halt (stops current emulated S/390 CPUs)
flexes> go (resumes operation, if not single step)
flexes> hwc ..... (enter text for system console)
flexes> ipl devaddr [parm] (IPLs the first emulated CPU)
flexes> iplc devaddr [parm] (clears state and memory first)
flexes> mount devaddr [filename | Offline] (discussed later)
flexes> notready devaddr (make emulated device not ready)
flexes> quit (terminate flexescli; also 'exit')
flexes> ready devaddr (makes emulated device ready)
flexes> restart (like the 'restart' button on S/390)
flexes> rewind devaddr (only for FakeTape or SCSI drives)
flexes> shutdown (gracefully terminates S/390)
flexes> unload devaddr (unloads FakeTape or SCSI drive)

(like a 'mount devaddr Offline' for other devices)

Here are a few brief notes concerning CLI commands:

- Unlike with the resource manager, CLI numeric parameters are assumed to be hexadecimal.
- There are many aliases for commands, not shown here (for example: go = g = start).
- The mount command is perhaps the most common CLI command. Do not confuse it with the Linux mount command. They are completely different commands that perform different functions.
  - A resources definition can define a device as Offline. This means that the emulated device exists but is, in effect, turned off.
  - A CLI mount command can, while an emulated S/390 is running, dynamically “turn on” the device, using a specified file or name. Some examples are:
    flexes> mount A90 /s390/WORK01 (assume A90 is a 3390)
    flexes> mount 710 altcons (assume 710 is a local 3270)
    flexes> mount 560 /home/tape3 (a FakeTape file)
  - The first example is similar to mounting a disk pack (if disk packs were available for 3390s). The named file (or symbolic link) should point to a properly formatted emulated 3390 volume.
  - The second example causes the name altcons to be added to the list of connections available through the Terminal Solicitor.
  - The third example, in effect, is a tape mount.
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 115.

- *S/390 Partners in Development: ThinkPad Enabled for S/390*, SG24-6507
- *S/390 Partners in Development: OS/390 (and z/OS) New Users Cookbook*, SG24-6204

Other resources

These publications from Fundamental Software, Inc., are also relevant as further information sources:

- *FLEX-ES Concepts*, FSIMM020
- *FLEX-ES Technical FAQ*, FSIMM040
- *FLEX-ES Planning Guide*, FSIMM100
- *FLEX-ES CLI Language Reference*, FSIMM210
- *FLEX-ES System Programmer’s Guide*, FSIMM300
- *FLEX-ES Resource Language Reference*, FSIMM310

Referenced Web sites

These Web sites are also relevant as further information sources:


How to get IBM Redbooks

You can order hardcopy Redbooks, as well as view, download, or search for Redbooks at the following Web site:

[ibm.com/redbooks](http://ibm.com/redbooks)

You can also download additional materials (code samples or diskette/CD-ROM images) from that site.
IBM Redbooks collections

Redbooks are also available on CD-ROMs. Click the CD-ROMs button on the Redbooks Web site for information about all the CD-ROMs offered, as well as updates and formats.
Index

Symbols
/boot 24
/etc/init.d/flexes file 55
/holding 12, 24, 63
/holding file system 95
/s391 63
/usr/flexes/bin 48, 68
/usr/flexes/rundir 49
/var/adm/flexes 59

Numerics
1403 carriage tape 92
3174 or 3274 control units 90
3270 terminal resources 108
3339 cylinders 90
3390 drives, odd size 90
4mm tape drive 10, 79
4mm tapes, compatibility 90

A
active wait 104
AD system 39
ADDRSSU backups 75
ASCII telnet sessions 33, 91
AWSCKD format 40
AWSOMA format 66
AWSTAPE format 66

B
backup and restore 72, 75
battery life 93
BIOS upgrades 21
boot manager 89
business partners, IBM 2

C
cachesize parameter 72, 104
cachestats command 72
CD-ROM drive 63
CD-ROM formats 40
CD-ROM, mounting 43
CD-ROM, OS/390 40
CD-RW drive 75
cfcomp command 48–49, 105, 111
channel 105
channel emulation 90
chkckd command 78
CKD disk resources 107
ckdcachestats command 71
ckdconvaws command 41–43, 46, 59
ckdconvaws program 44
ckddump command 96
cutdown command 59
CLI commands 113
CLI program 50
cloned parameter 111
COD system 81
command line interface (CLI) 34
consoles, sessions, screens 87
control unit types, emulated 106
control units and channels, which 89
cpu parameters 104
cu statement 105
Custom installation option 18
Customized Offerings Driver definitions 101
Customized Offers Driver 81

D
DASD cache 61
debugging information 59
debugging, basic 58
dedicate Server processor(s) 32
dedicated keyword 105
dedicated processor 94
devad 105
Device Support Facilities (DSF) 82
device types, emulated 106
device(00) parameter 108
devopt 108
devstate command 71
DFSMSdss 82
DHCP 93
DHCP server 12
disk cache 63, 70
disk caches 84
Disk drives 9
Disk Druid 14, 23–24
disk fragmentation 12, 96
disk layout 64
diskette drive 63
Domain Name Server (DNS) 12
dongle 90
dtprint command 60

E
ECC memory 9, 77
emulated S/390 devices, maximum 90
end statement 105
EP lines 94
ESCON and parallel channels 85
ESCON Channel Adapter 84
ESCON director 92, 95
essize, definition 104
eth0, resource 110
Ethernet adapters 10
Ethernet adapters, how many 90
Ethernet adapters, number 94
Ethernet connection, speed 93
Ethernet interface 2
Ethernet port 27

F
FakeTape 91
FCBs emulated 92
fdisk 14, 23, 64
feature lpar 104
Firewall security option 16, 25
firmware options 80
flexes command 50, 67
FLEX-ES console 31
FLEX-ES formats 41
FLEX-ES installation 34
FLEX-ES instance 4
FLEX-ES license 5
FLEX-ES license key, installation 35
FLEX-ES manuals 91
FLEX-ES names, rules 112
FLEX-ES releases 91
FLEX-ES remote resources 92
FLEX-ES traces 92
FLEX-ES upgrades 78
FLEX-ES, product 5
flexescli program 113
floating point performance 93
FSI channel adapters 90
FSI dongle 89
FSI hardware adapters 84
ftp 74
Fundamental Software, Incorporated 5

G
graphics adapter 7, 19

H
hardware installation and upgrade 21
Hot-Swap Power Supply 10
hwc command 60

I
IBM 3490-F01 drive 80
ICA Adapter 84
ICA adapter 90, 94
IND$FILE function 96
initawstape utility 66
instance, definition 4
instances, multiple 33
instruction cache 63
instset parameter 104
Integrated Server (IS) 84
Intel Pentium III 10
interface local(1) parameter 108
IOCDS 53
IODF requirements 53
IP address, changing 93
IP addresses 3
ipl command 52
ITSO/EFS project 1

K
kernel fixes and upgrades 93
kernel patches 27, 96
key mapping, x3270 68

L
LAN adapter, shared 54, 110
LAN adapters 33
LAN network 12
LAN resources 110
license key 92
license key file 89
lilo 29
lilo.conf 29
Linux commands 86
Linux kernel patch 27
Linux swapping 62
Linux windows 57
local 105
logo, USSTAB 91
LPARs 2

M
MAC address 90, 94
maxwritesize option 66
memory allocations 63
memory definition 107
memory, server 33
memsize, definition 104
MIPS 93
mount command 98
mount command, Linux 42
Mouse 10
MPTS adapter number 55
MPTS number 110
Multiprise 3000 84
multiprocessor patches 96
Multiprocessor servers 27
Multi-system setup 67

N
N3270 sessions, limit 91
Netfinity 5100 Server 9
Netfinity Installation 20
Netfinity/EFS 4
Network connections 12
networking setup 20

O
Operating Systems Messages console 60
OS/390 device configuration 41
OS/390 R10 AD system 41
OSA Express 3
Overland T490E tape drive  

P
P/390 system 80
P/390, R/390 84
Parallel Channel Adapter 84
Parallel Channel Adapters 95
Partners in Development 39
Partners in Development organization 1
PATH environment 37, 48
path parameter 105
PCI slots 9
PCOM 3
PEP lines 94
pipe 44
port 24, 33, 51
power consumption 95
power control 79
power supplies, redundant 77
printers 93
processor speeds 91
processor, definition 4
processors, four 94
processors, two 94
processors, use 94
PSW, display 77
PTFs 92

R
RAID configuration, steps 22
RAID driver diskettes 23
RAS characteristics 85
RAS discussion 77
raw disks/devices 12
Red Hat 7.1 Linux 5
Redbooks Web site 115
Contact us x
Redundant power supplies 10
Remote Operation 78
remote resources 69, 92
resadm command 49, 53, 55, 59
resadm command, options 112
rescf command 56
rescf files 111
resource definitions 105
resource manager 31
Resource Measurement Facility (RMF) 54
resource parameter 105
resources definitions 48, 97
root file system 95
rpm commands 78
rpm -U option 78

S
S/390 central storage 91
S/390 CPU, definition 4
S/390 expanded storage 91
S/390 identification 76
S/390 parallel channels 94
S3 SuperSavage/IxC16 (16 MB) adapter 19
SCO UNIX 1
screen saver 20, 29
SCSI adapter 9–10, 81
second hard disk 45
second Linux hard disk 63
Second processor 10
security 60
SEND and RECEIVE commands 96
serial number, CPU 76
Server memory 61
ServeRAID BIOS 22
ServeRAID Manager 22
ServeRAID upgrade 21
ServeRAID-4Mx Ultra160 SCSI Controller 9
ServerGuide CD 20–22
shell script 50
shutdown process 58
shutting down 55
slim-high hot-swappable disks 9
SMP system 27
SNA over Ethernet 93
SPECIFY SYSTEM PARAMETERS message 83
staging 45
Staging, disk loads 13
Stand-alone z/OS restore 81
Starter System 81
StarterPak systems 84
startup process 57
STIDP instruction 76
suspend function 79
swap (paging) rates 62
swap partition 24
syscf files 111
sysplex functions 96
sysplex, basic 96
system console 60
system definitions 97, 103
System Partition 24
system update function 22

tape devices, emulated 65
tape drive, 3480/90 80
tape resources 109
tar and ftp 74
tar to back up S/390 volumes 73
TCP/IP for OS/39 54
TCP/IP port 555 60, 69
telnet and ftp servers 19, 27
telnet application 3
terminal logo 91
Terminal Solicitor 31, 33–34, 51, 60, 91, 109
Terminology 4
text editors 87
text mode installation 23
ThinkPad displays 18
ThinkPad hardware 7
ThinkPad Installation 13
ThinkPad/EFS, name 4
ThinkPads, T20 and T23 7
TN3270 client 33
TN3270 emulator, Linux 53
TN3270 server 33
TN3270 session 53
token ring adapters 6
token ring LAN adapters 90
Token ring support 19
trace, processor event 92
trace, processor instruction 92
traces 92
traces, I/O device 92
tracesize parameter 104
trackcachesize 67, 70, 108
TSO and CICS users 91
TSO user 53
two processors 94

U
Ultra160 SCSI hot-swap slim-high drives 9
Ultrabay 63
Uninterruptable Power Supply (UPS) 92
UNIX System Services 3, 33
UnixWare PTFs 20
unzip program 42, 44
unzip, wrong order 46
USB device 89
userid, Linux 89
USSTAB logo 51
USSTAB module 91

V
vi editor 89
video adapters 6
video driver 7
VM/ESA 4
vmstat command 61
volser 96
VSE/ESA 4

W
Windows 89
writeback 70
writeback cache 71
writethrough cache 71
writethroughcache parameter 70, 108

X
X Windows configuration 19
x/Server EFS 84
x3270 client 67
x3270 command 50, 68
X3270 installation 37
x3270 sessions 34, 91

Z
z/OS TCP/IP 33
z/OS TCP/IP setup 91
z/VM 96
zip files, separate 46
ZIPed files 40
S/390 Partners in Development:

EFS Systems on a Linux Base

ThinkPad/EFS systems
Netfinity/EFS systems
Installation, customization, operation
z/OS AD CD-ROM software

A ThinkPad Enabled for S/390 (ThinkPad/EFS) or a Netfinity Enabled for S/390 (Netfinity/EFS) are the smallest S/390-compatible systems currently available that have been tested and approved by IBM. They are based on an IBM ThinkPad or Netfinity running Linux and the S/390 emulation product FLEX-ES. FLEX-ES is a product of Fundamental Software, Incorporated (FSI) of Fremont, California. The resulting system can run current S/390 operating systems, such as z/OS.

Small systems, such as these, are attractive for education, development, and less demanding production operations. This IBM Redbook describes the installation and use of these systems, based on usage of a particular z/OS package. This package is available only to members of the IBM S/390 Partners in Development (PID) organization who obtained systems through this organization. Consequently, this document is intended primarily for this group.

The information about Netfinity/EFS systems under Linux should be regarded as preliminary planning information. General support for Netfinity/EFS under Linux was not yet available at the time of writing.