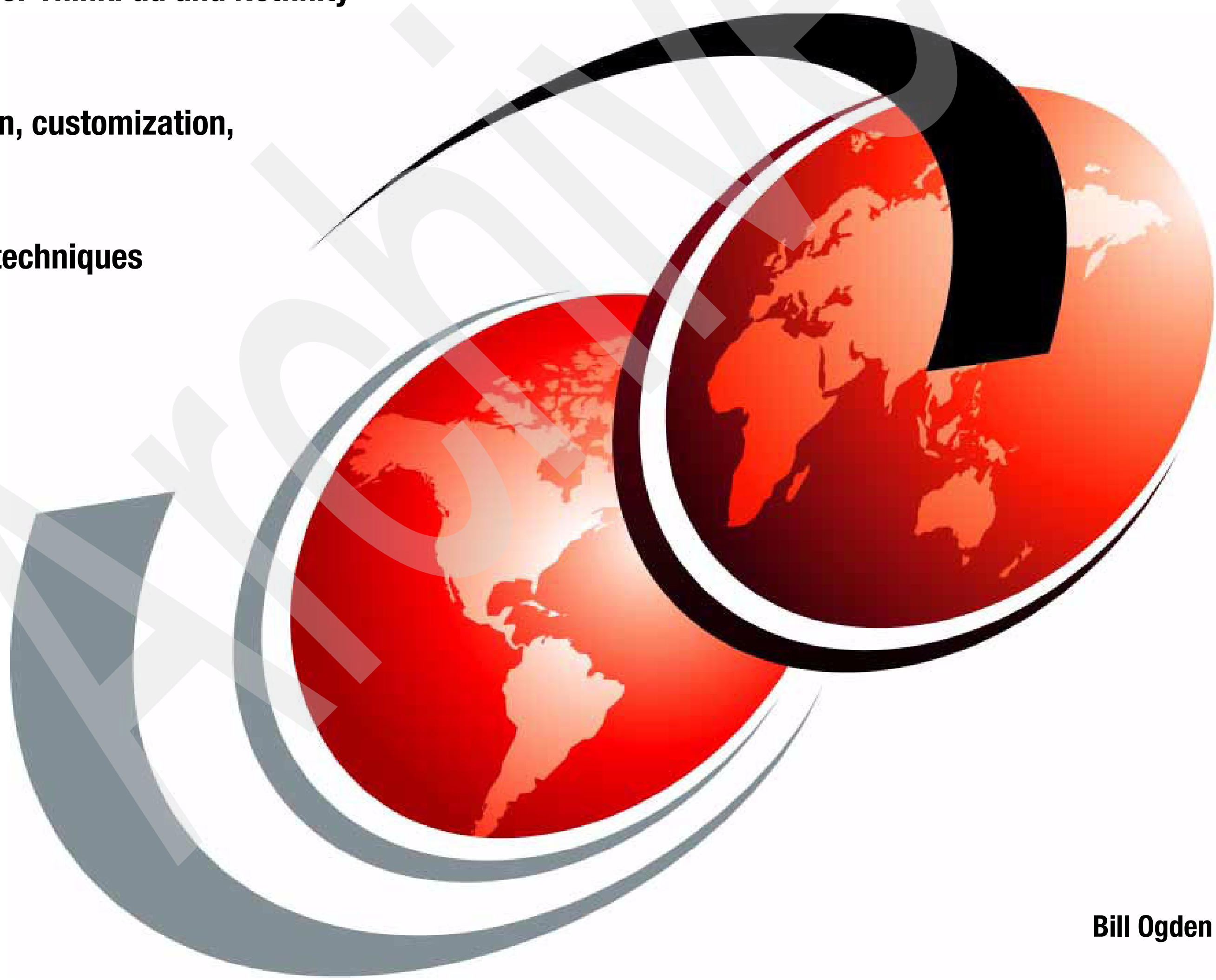
IBM

EFS Systems on a Linux Base: Getting Started

- Planning for ThinkPad and Netfinity bases
- Installation, customization, operation
- Common techniques



Bill Ogden

Redbooks

ibm.com/redbooks

International Technical Support Organization

EFS Systems on a Linux Base: Getting Started

November 2003

SG24-7007-01

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

Second Edition (November 2003)

This edition is based on FLEX-ES Release 6.2.14, Red Hat Linux Release 9.0, and the z/OS Application Development package for z/OS 1.4s.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrate 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:

@server®
ibm.com®
xSeries®
z/OS®
z/VM®
zSeries®
CICS®
DB2®
DFSMSdss™
ESCON®
FICON™
HiperSockets™
IBM®
IMS™
Multiprise®
MVS™
Netfinity®
OS/2®
OS/390®
Parallel Sysplex®
PartnerWorld®
Redbooks(logo) ™
Redbooks™
RMF™
S/370™
S/390®
ServerGuide™
ServeRAID™
ThinkPad®
Ultrabay™
VM/ESA®
VSE/ESA™
VTAM®
WebSphere®

The following terms are trademarks of other companies:

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

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

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

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

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

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

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

Preface

This IBM® Redbook describes the basic installation of FLEX-ES (a product of Fundamental Software, Inc., Fremont, California) with several z/OS® operating system packages. Much of the material is in cookbook format. More advanced techniques and options are described in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008. These books are intended primarily for members of the IBM PartnerWorld® for Development organization and for internal IBM users of the ITSO/EFS package.

This second edition includes the latest updates to the first edition of this book. It also contains more information about using Red Hat 9.0 with FLEX-ES Release 6.2.14.

The team that wrote this redbook

Bill Ogden is a retired IBM Senior Technical Staff Member, still working on favorite projects with 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. He is the originator of the ITSO/EFS package for internal IBM use. Bill has been with the ITSO since 1978. He completed this update for this book while working at St. Michael's Indian School, on the Navajo Reservation in Arizona.

We thank the following people from Fundamental Software, Inc. for their contributions to this project:

Gary Eheman, who helped us understand many additional aspects of FLEX-ES and Linux, and who was especially helpful with the updates in this book.

Michael Ryan, who helped with T40 ThinkPad®, performance, and upgrade questions.

Scott Carter, who is an eagle-eyed proofreader, corrector of errors, and a source of knowledge for obscure FLEX-ES details.

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners and/or customers.

Your efforts will help increase product acceptance and customer satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

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:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introduction

EFS systems provide small S/390 platforms. The EFS name is an IBM term and is derived from “Enabled For S/390.” EFS systems may be used to run z/OS (and OS/390®), z/VM® (and VM/ESA®), VSE/ESA™, and Linux for S/390. EFS systems may also be used for older versions of these operating systems. This document discusses z/OS use, but the reader should not conclude that the EFS is limited to z/OS.

EFS systems use FLEX-ES to provide S/390 emulation. FLEX-ES is a licensed product of Fundamental Software, Incorporated (FSI) of Fremont, California. A high-level view of an EFS system is presented in Figure 1-1.

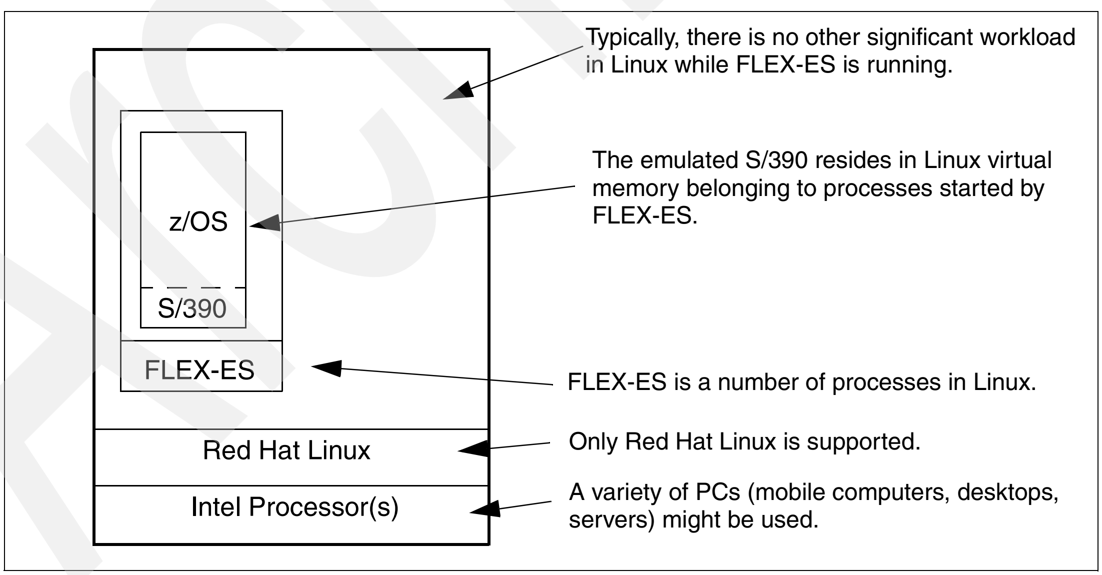


Figure 1-1 High-level view of an EFS system

FLEX-ES runs on an Intel-based PC under Linux or UnixWare operating systems. This redbook discusses only the Linux-based version.

FLEX-ES is a licensed product. The licenses are normally arranged through FSI Business Partners.[1] 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 for Linux is keyed to a specific serial number in a hardware dongle that is connected to the USB port of the PC and runs only in a system that has this dongle connected.

An EFS system is an S/390. *S/390 software licenses must be obtained as appropriate, both for IBM software and other vendor software.* These software licenses are arranged through IBM, Business Partners, or other vendors. This document does not further address these license requirements and assumes that any EFS user has all the appropriate S/390 software licenses.

In this document we use the term *S/390 emulation* when the term *S/390 and/or IBM @server zSeries® and/or S/370™ and/or other emulations based on S/360 designs* might be more accurate. FLEX-ES can emulate zSeries architecture at the Architecture Level Set 3 (ALS 3) level. It can also emulate S/390 and older architectures. For the purposes of readability we refer to all of these, collectively, as S/390 emulation and the resulting EFS system as a S/390.

EFS systems can serve many purposes. These include application development, demonstrations, education, and production work. While IBM discourages the use of MIPS measurements and comparisons, it may be helpful to note that typical EFS systems are roughly in the 12 to 25 MIPS range. The actual performance is directly related to the speed of the underlying PC, and the number of processors on the PC. A FLEX-ES license can limit the S/390 MIPS produced (assuming the PC is capable of faster operation) in order to meet S/390 software license limitations.

1.1 Purpose of this redbook

This IBM Redbook describes only Linux-based EFS systems and describes the installation and initial startup process for these systems. The specific combination described in much of this document, that is, EFS systems running z/OS AD CD-ROM software, is available to members of the IBM PartnerWorld for Developers (PWD) organization[2] who have obtained their systems through the PWD program.

EFS systems for PWD 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 an owner may not need to perform.

However, we believe this material will be useful 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.

This redbook updates and replaces an earlier version, *EFS Systems on a Linux Base*, SG24-6834.

[1] You can go to http://www.funsoft.com to contact FSI or locate an FSI Business Partner.
[2] This organization was previously known as S/390 Partners in Development (PID).

1.2 Terminology

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

- *Processor* means a PC processor. ThinkPads have a single processor, while some PC servers have multiple processors.
- *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 PC hardware.
- *Server operating system* means Linux in the PC.
- *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 PC resources.
- *z/OS* means the S/390 operating system. We could also use OS/390, z/VM, VM/ESA, Linux for S/390, or VSE/ESA, but we elected to work only with z/OS.
- *ThinkPad EFS* and *xSeries® EFS* are generic names. Business Partners providing these products often have their own names for their specific implementation of the products.
- *IBM @server xSeries* is the name of an IBM product line. Older products used the name *Netfinity®*.
- *S/390*, in this document *only*, is used to include both S/390 and zSeries functionality. This abbreviated reference is solely for readability.

For any *FLEX-ES instance*, there is usually 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 logical partitions (LPARs), but it is not quite the same. We can run multiple *instances of single-CPU S/390s* in a ThinkPad EFS system, for example.

> **Note:** Throughout this book, we frequently refer to the IBM Redbook *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008.

1.3 Connectivity overview

An EFS system can be self-contained in a PC. That is, the system can provide the required S/390 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 PC display. For practical purposes, this is a single-user system.

We can easily make a multi-user system by connecting the system to a LAN. This typically is done with a single Ethernet interface. The single LAN connection, as illustrated in Figure 1-2 on page 4, can be used in a number of ways at the same time:

- Client workstations, using 3270 emulators, can connect to a TN3270e *server* running under Linux in the EFS system. This server is the FLEX-ES Terminal Solicitor. It transforms client TN3270e connections to appear as local, channel-attached, non-SNA DFT 3270 terminals that can be used as z/OS operator consoles or VTAM (TSO, CICS) terminals. z/OS TCP/IP is not involved in these functions and need not be running. These *local 3270* terminals are the most basic z/OS terminal and z/OS is unaware they are actually remote TN3270e terminals.

- Client workstations, using 3270 emulators, can connect to z/OS TCP/IP (assuming it is configured and running). This interface can be used for normal VTAM purposes (such as TSO) but not for traditional operator consoles.
- Other z/OS TCP/IP functions, such as FTP, Telnet, a Web server, and so forth, function normally through this interface. If UNIX System Services Telnet is active, a remote user could connect to this interface.

An important point is that all these connections can operate simultaneously through a single physical LAN interface to the EFS system. If used as described, this interface has two IP addresses: one for Linux and one for z/OS.

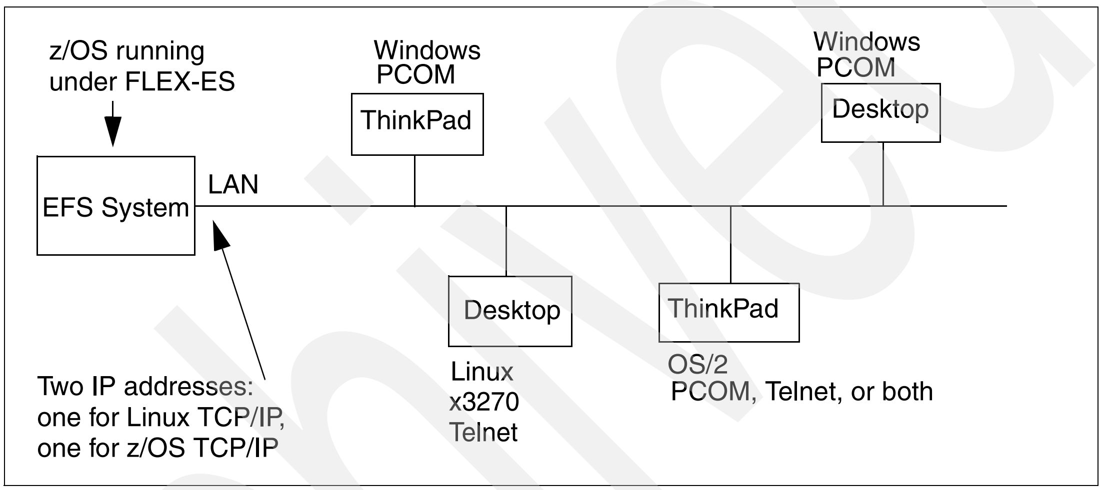


Figure 1-2 Multi-user System

With the configuration shown in Figure 1-2, a typical client might use a Windows-based mobile computer connected to the LAN. PCOM[3] can provide 3270 sessions that appear as locally-connected, non-SNA sessions to z/OS (if connected through the Terminal Solicitor) or as z/OS TCP/IP TN3270 connections. The distinction is made by which IP address and port number the client uses for the EFS connection. 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.

The EFS *host* system in this illustration could be a ThinkPad, a desktop PC, or a large xSeries server.

1.4 Basic EFS requirements

Key points in understanding hardware requirements are:

- No unique hardware is used for z/Series emulation. There is no “S/390 adapter.” Everything is software, with these exceptions:

[3] PCOM is the common name for the IBM eNetwork Personal Communications product. It provides TN3270e client functions.

  - A *dongle* is required. This is a device (match-box size) with a cable that connects to the USB port of your PC. 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, with the limitations mentioned below.
- The base operating system of the hardware platform is Linux. In particular, it is Red Hat Linux at selected release levels. Descriptions in this redbook assume the use of Red Hat Linux 9.0. *Only Red Hat Linux is supported.*
- A dual-boot system, probably with a version of Microsoft Windows as the alternate operating system is possible. Microsoft Windows is not otherwise relevant to EFS operation.
- 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 PC, you could emulate a zSeries machine with 256 MB of real memory. A PC with 1 GB memory could safely emulate a S/390 with 768 MB memory.
- Sufficient disk space is required. We recommend about 6 GB for Linux, although the actual requirements can considerably less. The remaining disk space required is whatever is required for your S/390 volumes. A single 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 PC disk space required.
- 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 larger server-based EFS systems. We recommend a separate SCSI adapter for these instead of a SCSI adapter that is part of a RAID adapter.
- Larger servers (but not ThinkPads) can use unique hardware adapters from Fundamental Software, Inc. At the time of writing, these are not supported under Linux; they are supported only for the UnixWare-based FLEX-ES systems. We expect these adapters to be supported for Linux-based FLEX-ES sometime in the future. These adapters 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. Two adapter versions will be available: a single ESCON channel and four ESCON channels.[4]
  - 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.
- If a PC server is involved, we recommend a ServeRAID™-4Mx (or later) Ultra160 SCSI Controller as the RAID adapter.

These considerations allow a range of systems to be considered. This list does not address the speed of a PC to be used for EFS. We must note that the relative clock speeds of two PCs (often expressed in Gigahertz) does not necessarily indicate the performance ratio between the two systems. Memory bandwidth, bus speed, DMA operation, disk performance, and cache operation have major effects on performance.

Assuming you want to run z/OS, we suggest a system with 1 GB memory, at least 1 GHz clock speed, and 40 GB disk space as a reasonable starting point. Smaller and slower

[4] We use the future tense here because the ESCON adapters were not yet available at the time of writing.

systems also work. Larger and faster systems may be needed for specific applications or workloads.

1.4.1 System selection

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 memory 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 xSeries EFS system is much more robust and is suitable as a production server. Important factors include:

- ECC memory (or equivalent) is used. The importance of this for long-term stable operation cannot be overstated.
- RAID disks are used and assumed in all xSeries EFS discussions. RAID 5 is typically used and provides automatic protection against any single disk failure. A hot spare is always recommended and is considered normal in larger configurations.
- Multiple, redundant power supplies are available for the approved xSeries servers and provide N+1 protection.
- Tape connections are more likely to be available in this environment, making routine backups more practical.
- An xSeries 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).

[5] This requires additional FLEX-ES license capabilities, with a corresponding increase in license fees.

1.5 EFS limitations

There are a number of functions not available for EFS systems. 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 adapters or coprocessors are not available.
- OSA-Express functions are not available.[6]
- HiperSockets™ are not available.
- LPARs are not available. However, if sufficient memory is available in the PC, we can simultaneously emulate multiple z/Series machines. This is not quite the same as using LPARs in a 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.
- The new and altered instructions in the zSeries z990 systems are not available at the time of writing.
- Multiple paths to a device are not supported.

FLEX-ES has formal support only on specific xSeries models and specific ThinkPad models. FSI adds to the list of supported models over time. This is discussed in more detail in the next chapter.

1.6 Detailed FLEX-ES overview

This section provides a slightly more detailed description of FLEX-ES than was presented earlier. FLEX-ES can be viewed as the following components:

- A S/390 instruction emulator, which is the heart of the system. It examines each S/390 operation code and emulates that operation, using the instructions of the 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 TN3270e 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 1-3 provides a simplified view of FLEX-ES operation. FLEX-ES is simply a process[7] 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.

[6] Basic OSA functions over Ethernet and token ring are available.
[7] It is actually a number of interrelated processes and threads, but this detail can be ignored at the conceptual level.

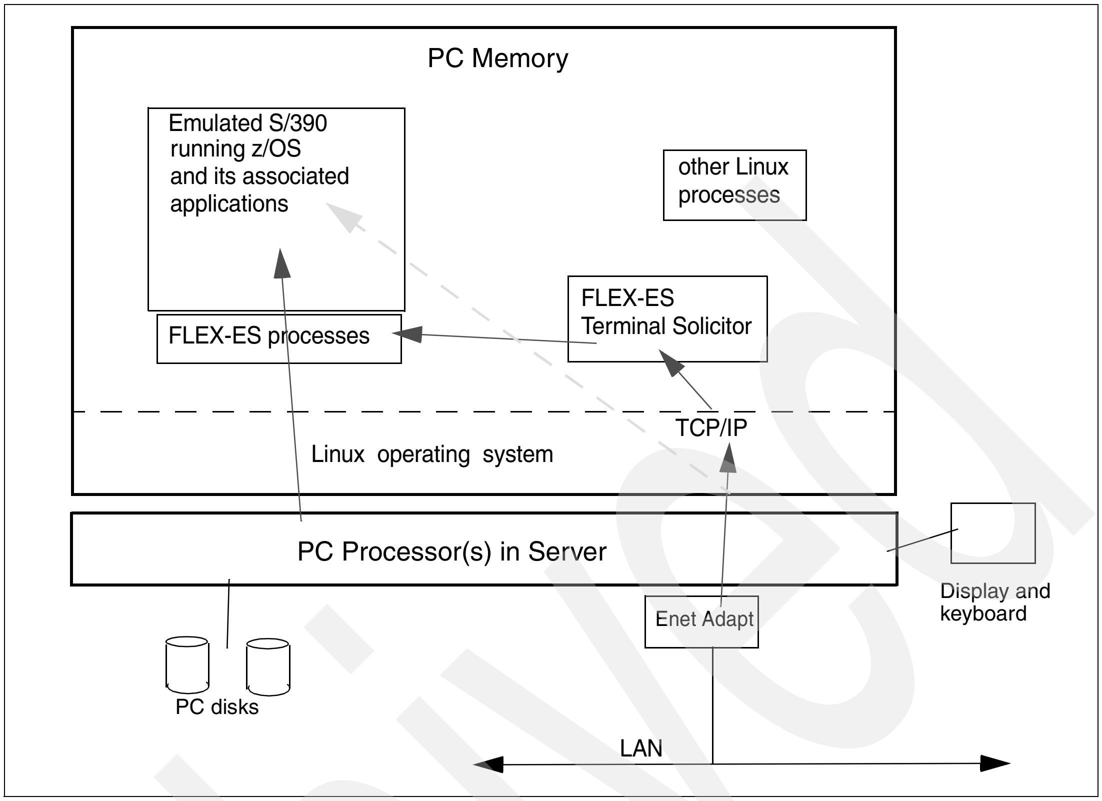


Figure 1-3 Conceptual view of FLEX-ES system

Key points include:

- FLEX-ES is a software product.[8] All the hardware shown (in Figure 1-3) is standard PC hardware. (The hardware dongle required by FLEX-ES is not shown in the figure.)
- The FLEX-ES program 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 PC’s disks.
- The FLEX-ES license specifies how many PC processors may be used (at any one instant) for S/390 emulation. For a ThinkPad EFS system, this is one processor. For an xSeries EFS system, this can be multiple 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 dedicated, then the normal Linux dispatching function dictates which processors are being used by FLEX-ES at any instant. This number never exceeds 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

[8] The optional S/390 channel adapters available for use with FLEX-ES are ignored in this discussion.

describing, this means that only a single-processor S/390 can be emulated. With an SMP server a S/390 with multiple CPUs can be emulated.

- The emulated S/390 can use LAN adapter(s) on the PC. z/OS TCP/IP can run this way. *A single physical 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.[9]
- 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 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.[10] Each one must have appropriate VTAM definitions for a local 3270, of course.
- If a LAN adapter is defined 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.[11] The server processor(s) licensed for S/390 emulation are dispatched among the several emulated S/390s. Only one Terminal Solicitor is used; it is 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 PC 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. z/OS paging (in the emulated S/390) is a completely different issue and a reasonable amount of paging has no drastic 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.
- 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.

[9] It also differs from P/390-based machines and the MP3000, none of which permit sharing of LAN adapters by multiple TCP/IP stacks.

[10] There is an overall FLEX-ES limitation of 2048 emulated devices. The actual limit is also controlled by the FLEX-ES license being used and is typically 512.

[11] 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 FLEX-ES console is actually a Linux process and might be described as a virtual console. It might not have any terminal (window) connected to it, or it might have several connections. The `flexescli` command can send commands from a Linux window to the FLEX-ES console. Used another way, the **flexescli** command can create a full-time connection between a Linux command window and the FLEX-ES console.

2

Practical planning

You must do some practical planning before starting a FLEX-ES installation. We can divide the planning into several areas:

- General topics that are relevant to any FLEX-ES installation:
  - Your goal: A production or sandbox system?
  - The Linux distribution to be used
  - Newer hardware and Linux support
  - Disk space needed
  - Linux file system layout
  - Mode of disk usage for emulated S/390 disk volumes
  - Network connections
- Topics that are especially relevant to ThinkPad installations:
  - Device planning (Ultrabay™ usage)
  - Choices for disk layouts
  - Dual boot, alternate boot
- Topics that are especially relevant for larger server installations:
  - SMP kernels, large memory, and new LAN devices
  - BIOS and RAID adapter updates
  - RAID configuration
  - External hardware (SCSI tapes, channels, and so forth)

These topics interact with each other. We strongly suggest you read all the discussion in this chapter rather than reading only isolated topics. The discussions here (and throughout this document) assume that you use FLEX-ES Release 6.2.14 or later.

2.1 General planning

These topics are relevant to any platform used for FLEX-ES and you should understand the following discussions.

2.1.1 Production or sandbox

By *production*, we mean a high-performance, highly stable system. By *sandbox*, we mean a convenient, easier-to-install system. These two categories do not cover all situations, of course, but they are convenient terms. In the sense used here, *production* implies some combination of serious business, important work, maximum performance, or a highly-stressed system. For example, a basic sysplex using multiple S/390 instances in a ThinkPad falls in the production category, even if used only for self-education, because it represents a highly-stressed system.

Production systems, in the sense used here, require a little more installation effort than a sandbox system. For example, a production system should include:

- Superior hardware, including RAID disks and ECC or similar memory
- The latest BIOS updates and RAID firmware updates
- The latest Linux drivers and utilities for managing RAID adapters
- The latest Logical Volume Manager (LVM) updates (if LVM is used)
- The use of raw disk interfaces instead of simple Linux files for emulated S/390 volumes
- Installation of appropriate Linux fixes and updates
- Careful tuning of FLEX-ES disk buffers
- Selection of Linux partitions and file systems to avoid “file system full” exposures

For various redbook projects, we installed FLEX-ES systems using none of the above options and other systems using most of these options. Both approaches produced useful, working FLEX-ES systems. New users may start with a simple sandbox installation to gain experience and confidence with FLEX-ES, while investing less time and effort than needed for a production system. There is no firm dividing line between production and sandbox systems. Your system may use some of the listed options and ignore others.

This redbook concentrates mostly on a sandbox installation, although “production-level” facilities are briefly mentioned in various sections.

2.1.2 Selecting a Linux distribution

FLEX-ES is tested and supported only with selected releases of Red Hat Linux. It *may* work with other Linux distributions, but there is no support if something goes wrong.

It is important to understand what is meant by a release of a Linux distribution:

- A Red Hat release contains a *stock* Linux kernel; that is, a pre-built kernel that is installed as part of the standard installation process. (There may be more than one stock kernel; perhaps one for uniprocessors and another for SMPs.)
- There is a *kernel source tree* that can be used to rebuild the kernel with different options.
- There are many *packages* that can be selected during installation. For example, you may want the gcc compiler, a Web server, X Windows interfaces, and so forth. Each of these packages has its own release level.

For example, Red Hat 9.0 provides a stock kernel at level 2.4.20 (with a corresponding kernel source tree) and the gcc compiler at level 3.2.2. For reasons discussed later, you may replace the kernel source tree with a different level and build a new kernel at the different level. Your system is still considered to be Red Hat 9.0, but with a different kernel. Alternately, you may rebuild the kernel that was provided with your distribution but with different kernel options selected.

In most cases, the stock kernel with the Red Hat release may be all you need. In other cases, you *must* rebuild the kernel because the stock kernel does not provide the options you need. We know of the following cases in which you *must* rebuild the stock Red Hat 9.0 kernel:

- If an SMP machine is used for FLEX-ES
- If some of the newer Ethernet adapters are used, especially 1000BaseT models
- If large server memory is to be used, where *large* means 1 GB or more
- If new RAID or LVM modules are to be integrated in the kernel

At the time of writing, we used or tried the following releases (with their stock kernels) with the indicated results. In all cases, the kernel may need to be upgraded, rebuilt, or both for the reasons just mentioned.

- Red Hat Linux 7.2: This release works well for older systems, including ThinkPads through T21 models. The standard graphics driver does not work with later ThinkPads, or with ThinkPad docking stations. This was a solid release that was used with many Server-based FLEX-ES installations. If appropriate graphics drivers are installed, it is still a good release.
- Red Hat Linux 7.3: The stock kernel does not work with FLEX-ES and we do not recommend this release.
- Red Hat Linux 8.0: The stock kernel works well. We recommend that you use this release unless your PC has USB 2.0 ports. This release does not work with USB 2.0 ports.
- Red Hat Linux 9.0: The stock kernel works well with FLEX-ES Release 6.2.11 and later. (Earlier FLEX-ES releases require a kernel rebuild to use this Red Hat release.) This release does not work correctly with some PCMCIA cards. (You can find more information about this on Red Hat Web sites.) A kernel rebuild may be necessary for some ThinkPad Ethernet adapters.

It is important to understand the criteria we use to recommend a Linux distribution and release. The sole concern here is to build a small S/390, where Linux is one of the building elements. We are not concerned with other uses of Linux, and these recommendations would probably be different if we were interested in other Linux uses. We strongly suggest that, if you want to work with Linux for purposes other than FLEX-ES, you build *another* Linux system. For that system, you can install whatever distributions and releases you wish.

2.1.3 Newer hardware and Linux

PC hardware changes quickly. Commonly available Linux distributions and their stock kernels may not support the latest hardware. Sometimes this can be fixed by rebuilding the kernel (or a newer kernel) with different options.[1] Sometimes it can be fixed by installing new device drivers (if you can obtain them).

Particular areas of concern are: the processor (which must be absolutely compatible with Pentium 3 or Pentium 4), graphics chips (the most common problem), USB 2.0 interfaces, RAID adapters, LAN adapters (there are always new ones), and memory size (a relatively new concern).

These are generally Linux issues. For the most part, if Linux is working properly with the hardware, then these issues are transparent to FLEX-ES. However, FLEX-ES requires some

[1] A frequent question is, "Why not include all the options in the kernel?" The answer is that some may be mutually exclusive and, ignoring this, it would make the kernel quite large. Do not confuse the selection of kernel options with the selection of Red Hat installation options.

specific Linux kernel options and FLEX-ES installation must access a kernel source tree that accurately reflects the kernel being used.

FLEX-ES does not claim support for all PC hardware. FSI publishes a list of supported machines, and this is a relatively small list. It is our experience that many other machines work well, but that the very latest machines may have problems. (As mentioned, these are usually Linux problems, but nevertheless may prevent you from using FLEX-ES successfully.)

Therefore, if you want to use FLEX-ES with the very latest, largest, fastest, super PC server or mobile computer you must recognize that you are probably beyond the range of *supported* systems. It may work, but you *may* need to learn something about Linux drivers, kernel builds, and kernel levels and you may need a little more patience than users of more prosaic machines. Restrictions (and fixes) discussed here are subject to frequent changes. If any of these concerns apply to you, you should obtain the latest advice from whomever provides your FLEX-ES support.

USB 2.0 ports

If your system has USB 2.0 ports, we suggest you use FLEX-ES Release 6.2.11 (or later) and Red Hat 9.0.

1000BaseT LAN adapters

The newest mobile computers and server systems may include 1000BaseT Ethernet adapters. These are not supported by the stock Red Hat 8.0 kernel. The kernel source distributed by FSI and Red Hat 9.0 both support the more common 1000BaseT LAN adapters.

Large memory

Servers with 1 GB or more memory require kernel changes to be most effective. This requires rebuilding the kernel.

2.1.4 Disk space needed

PC disk drives are much larger than they were a few years ago. A FLEX-ES system running z/OS requires a considerable amount of disk space. What is considered ample space for normal PC usage may not be adequate. Consider the following:

- Red Hat Linux 9.0 needs 2 to 7 GB of disk space, depending on the installation options selected. This includes a reasonable amount of work space and the space needed by FLEX-ES modules.
- If you have a dual-boot system (probably with Microsoft Windows as the "other" system), you need whatever disk space that system requires.
- Most S/390 (and zSeries) software is installed on 3390 volumes. FLEX-ES emulation of 3390 volumes uses almost exactly the same amount of space as the original device being emulated. For example, a 3390-3 can contain about 2.8 GB of data and FLEX-ES requires 2.8 GB of disk space on your PC to emulate a 3390-3.
- The amount of disk space needed for a z/OS system varies greatly. It might be as little as a single 3390-3 (2.8 GB), although such a system would probably not be very useful. A small TSO-oriented system might be four 3390-3 volumes and perhaps a 3390-1 for local data sets. This would be slightly over 12 GB. A larger system, such as the prebuilt systems available to certain users on CDs, if fully installed, might need 12 or 13 3390-3 volumes, plus at least one local volume, with a total of about 40 GB. If a large amount of local data is involved, space is needed for this.

- You might or might not install the z/OS distribution library volumes (DLIBs). This can easily be four 3390-3 volumes (and these are included in the 40 GB just mentioned). If you do not need these, you save about 11 GB.

The question, "How much disk space do I need?" is almost impossible to answer unless you have a fairly detailed idea of what volumes you need. A general answer is that, if you want to use a recent z/OS release with many of the common subsystems and products, you should have *at least* 40 GB disk space. More is better.

2.1.5 Linux file system layout

Our installation descriptions produce a very basic file system layout, with only four partitions:

- A small boot partition. (A separate partition might not be necessary, but almost all Linux documentation seems to do this.)
- A small swap partition (because a Linux system used for FLEX-ES should not do much paging).
- A root file system that contains all the normal Linux file systems, such as /usr, /tmp, /etc, and so forth.
- A separate, large file system for emulated S/390 volumes. This can be a large LVM area instead, if raw disk devices are used.

Experienced Linux (or UNIX) users may object to this, saying that it is better to have separate partitions (file systems) for /usr and /tmp and /var and so forth. This is probably correct, but we believe this is not needed for a basic FLEX-ES installation. This redbook concentrates on *getting started* with FLEX-ES and attempts to simplify the process as much as possible.

If substantial other usage of the Linux system is anticipated or if exposures due to inadvertent filling of file systems are a concern, then it is better to have separate partitions (or logical volumes) for such things as /usr, /tmp, and so forth. If this is done, then you must decide on partition sizes for these file systems, of course.

We could make the root file system very large and place all the emulated S/390 volumes there. We elected to create a separate partition for the emulated volumes for two reasons:

- It is easier to "scratch and reinstall" Linux without disturbing the S/390 volumes.
- There are fewer potential fragmentation problems for the S/390 volumes.

Our descriptions use Linux *ext3* file systems where appropriate. The ext3 file system offers better recovery than other common file system types.

2.1.6 Disk modes for FLEX-ES

Emulated S/390 volumes (usually 3390 devices) can be handled in two ways:

- An emulated 3390[2] can be a single Linux file. It might be a large file, 2.8 GB to emulate a 3390-3, for example, but it is a simple Linux file.
- An emulated 3390 can be a Linux *raw device*. The raw device is then equated to a disk partition (as created by the **fdisk** utility or something similar) or a logical volume (as created by a LVM). A separate partition or logical volume is needed for each emulated 3390 volume.

[2] We use 3390s in this discussion, but the same comments apply to any type of emulated S/390 disk drive.

Each method has advantages and disadvantages:

- The simple Linux files are easiest to set up, especially for a relatively unskilled Linux user. Less planning is needed and disk management is more automatic.
- The simple Linux files are a little slower than the raw device method, and emulated disk response is less predictable. (Normal Linux disk I/O processing, including caching, is used for Linux files. This Linux processing uses instruction cycles. The Linux disk caching and the FLEX-ES disk caching interact in ways that make response less predictable.)
- The raw devices are more complex to set up. In practice LVM is used instead of "real" disk partitions[3] and the user must learn something about LVM usage.
- The raw devices offer smoother disk performance and use less processor overhead.
- Fragmentation is not an issue when raw devices are used.

When a z/OS application writes a block to disk, FLEX-ES writes to a FLEX-ES cache for that volume. By default, each emulated 3390 has 15 tracks[4] of cache.[5] When FLEX-ES decides to write the contents of a cached track to disk, it uses a normal `write` interface for Linux. If the file is a normal Linux file, in a normal Linux file system, Linux then places the data in its own cache and flushes this cache to disk sometime later. If the file is a raw device, the FLEX-ES write operation causes a direct disk write. No Linux file system processing or caching takes place.

FSI recommends the use of raw devices for all FLEX-ES installations. We, in this redbook, offer slightly different recommendations:

- We recommend that you use a raw device for production systems or if every possible bit of disk performance is needed.
- We recommend that you use simple Linux files for less demanding sandbox environments, especially for ThinkPads used by relatively few users and used for self-education, demonstrations, or minor development. In general, we believe the use of simple Linux files is good enough for this environment and makes installation less complex.

The remainder of this redbook works with FLEX-ES installations using simple Linux files for S/390 disk emulation. A chapter in a companion redbook, *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008, describes the use of raw devices.

It is possible to use both disk methods. Some emulated volumes can be in standard Linux files and other volumes on raw devices using logical volumes.

FSI Business Partners installing commercial FLEX-ES systems usually work with raw devices. This is appropriate because they have the skills to plan and install such systems and these production systems usually have a relatively static emulated disk layout.

2.1.7 Network connections

Your system can be connected to the Internet or 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

[3] Only a limited number of "real partitions" can be placed on a disk. Also, subsequent changes with `fdisk` or a similar utility can easily cause the loss of the contents in all the partitions on the disk.
[4] That is, 15 3390 tracks, where a 3390 track is about 57 KB.
[5] This is variable and a sophisticated FLEX-ES installation may tune the FLEX-ES cache options.

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 *need a static IP address* in order *to use z/OS TCP/IP*.[6]

- 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) using 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.

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 *non-routable* LAN addresses in the 192.168.x.x address range. If you are connected to a larger LAN, we suggest you install our way first (with no LAN, or with a small temporary LAN[7]). 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.

2.2 ThinkPad topics

ThinkPads have a few unique issues, mostly related to I/O device combinations and attachments. The considerations are not complicated, but we suggest that potential ThinkPad users review this material.

2.2.1 T40 ThinkPads

You should be aware that T40 ThinkPads (and newer ThinkPads) have USB 2.0 ports. It is best to use Red Hat 9.0 and FLEX-ES Release 6.2.11 or later with these ThinkPads.

2.2.2 BIOS settings

You should verify that the *Boot device List F12 Option* is 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*.

We adjusted some of the power management options we found in the BIOS. This may not be appropriate for you, but we did not want the screen to suddenly go blank when it contains the z/OS console window, for example. These controls are found by pressing F1 during ThinkPad startup and following **Config->Power**. Be certain to scroll down (using the down arrow key) in this window to find all the controls.

z/OS cannot tolerate a ThinkPad Suspend or Hibernate operation. We disabled the *Hibernate by timer* option. We did not disable the *Suspend when LCD is closed* function, but we are careful not to close the ThinkPad cover when z/OS is running. We did leave a 3-minute *HDD off timer* active.

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.

2.2.3 Device planning (Ultrabay usage)

Most ThinkPads have an internal disk drive bay and an Ultrabay. The Ultrabay can contain a second hard disk, a diskette drive, or a CD-ROM (or CD-RW or DVD) drive, but only one

[6] This discussion applies to any S/390 operating system run under FLEX-ES, of course, and is not unique to z/OS.
[7] This can consist of a single Ethernet hub or token ring MAU.

device at a time. Furthermore, you can boot from any of these devices, by making appropriate BIOS selections when you start the ThinkPad.

FLEX-ES systems tend to use a large amount of disk space, primarily for emulated S/390 volumes. Also, many owners want to use their ThinkPads for other purposes, such as day-to-day Windows operation. These requirements can be met in a variety of ways.

A key factor here, considering z/OS use under FLEX-ES, is that you may need to access the disk containing your emulated S/390 disk volumes and a CD-ROM drive at the same time. This occurs when loading a new release of z/OS from CD to a hard disk, for example. You may also need to access a diskette drive while running Linux.

These points may appear obvious, but please consider common situations:

- You want to keep your current internal disk, containing your day-to-day Windows functions, and add Linux as a dual-boot function on this disk. You then obtain a second hard disk for the Ultrabay and place all the large S/390 emulated volumes there.
- Another variation of this is that you place your complete Linux + FLEX-ES + S/390 volumes on a second hard disk and boot it from the Ultrabay.

In both cases, your internal disk drive and your Ultrabay are occupied. Now, if you need to use a CD-ROM drive to install new S/390 volumes, where do you install the CD-ROM drive? The same question exists for a diskette drive.

There are several solutions and you should think about these before deciding how to configure your ThinkPad system for FLEX-ES:

- An attractive option is to obtain a ThinkPad docking station.[8] This has another Ultrabay slot where you can install a CD-ROM drive, a diskette drive, or another hard disk. An additional advantage is that it appears to provide better air flow (cooling) for a long-running system. A docking station is not as portable as a ThinkPad, but you probably do not need to use the additional Ultrabay very often. An ability to borrow a docking station might be an option.
- Some port replicators, USB attachments, or both support diskette and CD-ROM drives. *Provided that Linux can recognize and use these drives*, these options may solve the problem. We do not have a specific list of these attachments that work with Red Hat Linux 9.0.
- You can obtain an internal disk drive that is large enough to hold your Windows partition(s), Linux partitions, and all the emulated S/390 volumes you need. This generally means a 60 GB drive or larger.
- You can remove the internal Windows drive and install a Linux drive in the internal disk slot. This is obviously not a recommended ThinkPad procedure although we have done it many times without problems. It must be done carefully and gently, with particular attention to the external plastic cover that is needed to “grab” the disk for removal.[9]
- You can *stage* S/390 volumes from a CD-ROM drive (in the Ultrabay) to space in a Linux partition on the internal disk. You then remove the CD-ROM drive, install your second hard disk in the Ultrabay, and copy the volume from the internal drive to the Ultrabay drive. The amount of free Linux space on the internal drive determines how many volumes can be staged at one time. A single 3390-3 volume requires 2.8 GB. You must stop and

[8] The IBM Web pages indicate that docking stations are not supported for Linux. Based on this, FSI does not support their use. If you are interested in docking stations, read the Redbook *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008, for a brief discussion of our experience.
[9] We noticed that some recent ThinkPad hard drives are shipped without a lightweight metal carrier on the bottom of the drive. This carrier is needed to reasonably install the drive in the internal drive slot.

reboot Linux for every Ultrabay change because Linux does not support hot swapping of Ultrabay devices.[10]

2.2.4 Choices for disk layouts

The preceding discussion refers to a number of disk layout options. This section describes these in more detail. Figure 2-1 roughly illustrates several options.

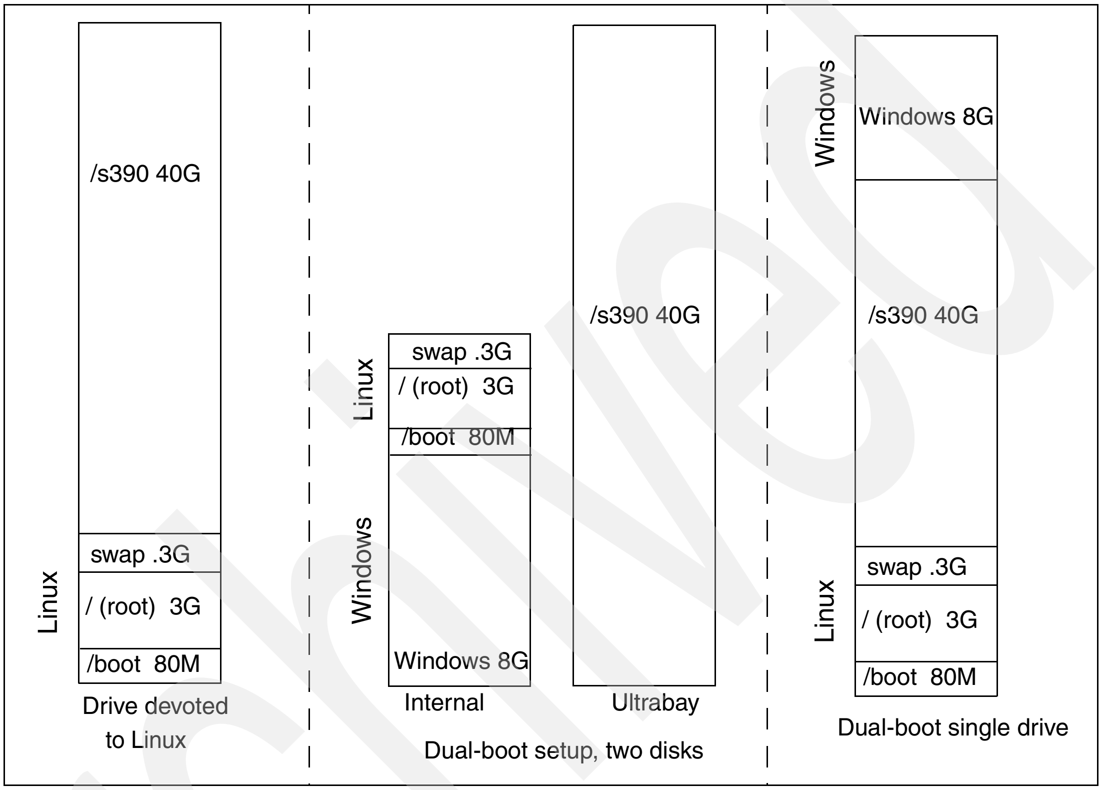


Figure 2-1 Three disk layout options

The sizes shown, such as 3 GB for a Linux root partition, are somewhat arbitrary and depend on your particular requirements. The first drive in the illustration, devoted to Linux (and FLEX-ES and S/390 emulation), can be used three ways:

- It can be the sole drive for the system and your ThinkPad is devoted to S/390 operation.
- It can be installed and booted in the Ultrabay, leaving your internal disk (which may contain Windows) unused for the duration of your Linux session. Some adjustments are needed to boot Linux from the Ultrabay. These are discussed in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008.
- It can be swapped with your normal internal hard disk. The internal drive bay in a ThinkPad is not designed for frequent changes, and this limitation needs to be considered.

The second set of drives in the illustration has both Microsoft Windows and Linux on the internal drive. The total space needed for Linux is less than 4 GB; all the remaining space can be left for Windows. A dual boot function is used to select either Windows or Linux for booting. A second hard disk, in the Ultrabay, contains all the emulated S/390 volumes. This disk needs to be in the Ultrabay only when you are using S/390 emulation, of course.

The third option illustrated is the most straightforward, if your disk is large enough.

[10] This was true at the time of writing, but may change in the future, of course.

2.2.5 Dual boot, alternate boot

Dual boot means that two (or more) operating systems are present on a bootable disk. Typically, a small program is loaded when the system is booted, and this program allows the user to select which of the main operating systems to load.[11]

Alternate boot, in the context used here, means to boot from an alternate hard disk in an Ultrabay slot. With a ThinkPad you can, for example, arrange BIOS options such that the system boots from a hard disk in the Ultrabay if one is present. If a hard disk is not in the Ultrabay, it boots from the internal disk.

Dual boot

There are many ways to provide dual boot functions. We describe one method in this redbook. It involves:

- Starting with a hard disk that contains only a Microsoft Windows partition, we use PartitionMagic (a product of PowerQuest) to “squeeze” the Windows partition into a smaller area to make room for Linux partitions.
- The Linux installation process detects the presence of the Windows partition and offers to install a dual boot function; we used `grub` and the Master Boot Record for this purpose.[12]
- When we boot the resulting system, we have a choice between Windows and Linux.

A concern with dual boot solutions is maintenance. The various methods of implementing dual boot functions are not always compatible with all operating systems. For example, you might apply maintenance to the non-Linux operating system that causes the dual-boot loader to be overlaid. If this happens, you need to recover the dual boot loader somehow. (One option is to create a Linux boot diskette and use this to begin recovery from such problems.)

The process we used to “squeeze” an existing partition (to make space for Linux partitions) is briefly described in 2.4, “Dual boot preparation” on page 21. The relevant point during Red Hat Linux 9.0 installation is mentioned in 3.1, “Basic Red Hat Linux 9.0 installation” on page 24.

2.3 Larger server topics

By *larger server* we typically mean an SMP (multiprocessor) machine with RAID disks and large memory. It might also have FSI channel adapter cards, SCSI tape drives, and other peripherals attached to it.

A simple desktop PC falls between a ThinkPad and a larger server. In most cases, it can be regarded as similar to a ThinkPad without the Ultrabay issues.

2.3.1 SMP, large memory, new LAN adapters

At the time of writing, you must rebuild the Linux kernel if you plan to run FLEX-ES on an SMP system. You may need to rebuild the kernel if you have unusually large memory or newer LAN adapters. This is discussed more in 3.3, “Kernel rebuild” on page 30.

[11] There are other ways to do this, of course. We do not attempt to explore all the ways to implement dual-boot systems.

[12] `grub` is one of the loader programs available for Linux.

2.3.2 BIOS and ServeRAID adapter updates

IBM and FSI strongly recommend that you install BIOS and firmware upgrades before installing FLEX-ES. (If you obtained your system from an FSI Business Partner, this work may have been done for you.) You can obtain these updates by downloading the latest ServerGuide™ CD, ServeRAID CD, or both from the IBM Web sites. The exact installation procedures vary with different server models. We do not attempt to provide specific installation instructions here.

2.3.3 RAID configuration

The ServerGuide CD allows you to customize the RAID configuration, using the ServeRAID Manager program on the CD.[13] This is a graphical user interface (GUI) program, and it is difficult to describe the operation in a few words. The goal for our sandbox system 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.

The ServeRAID Manager displayed our initial, default configuration as:

```
Logical drv     Size MB     RAID Level     Array     Hot Spare
 1 New           52071         5            A          Yes
```

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. This was acceptable for our minimal test system. However, we strongly recommend a hot spare for a production system.)

2.4 Dual boot preparation

There are many ways to install dual boot capabilities and we do not attempt to address the general topic. The following description provides the base for a simple dual boot that has worked well for us. This section describes only part of the setup. The actual dual boot code (using Linux **grub**) is installed during Linux installation, as described in the next chapter.

This description is oriented to a ThinkPad, but you should have no difficulties adapting it to other systems.

We started with a hard disk completely occupied by a single partition with a version of Microsoft Windows installed. We used PartitionMagic (a product of PowerQuest) to reduce the size of the Windows partition, making space for several Linux partitions. You can allocate the Linux partitions using PartitionMagic, or you can allocate them during Linux installation. We suggest allocating the Linux partitions while installing Linux (and while using Disk Druid), but you need to take care not to disturb the Windows partition at that time.

In our particular example, we had a 15 GB internal disk drive on a ThinkPad. The drive contained Microsoft Windows and a number of applications. Our goal was to install Linux (with FLEX-ES) on this disk without destroying the existing contents. The disk is too small for z/OS volumes. We intended to place them on another hard disk in the Ultrabay slot. With this setup, we can boot either Windows or Linux from the internal ThinkPad disk. If we intended to work with FLEX-ES, we would install the second hard disk in the Ultrabay before booting Linux.

[13] The ServeRAID CD should also have the same programs.

We obtained PartitionMagic on two diskettes. After booting it (and after it called for the second diskette), we received a display something like that illustrated in Figure 2-2.

This reflects a 15 GB drive (14396 MB), with a single partition. 2385 MB are used in the partition, and 12010 MB are free. We arbitrarily decided to reduce the Windows partition to 8 GB.

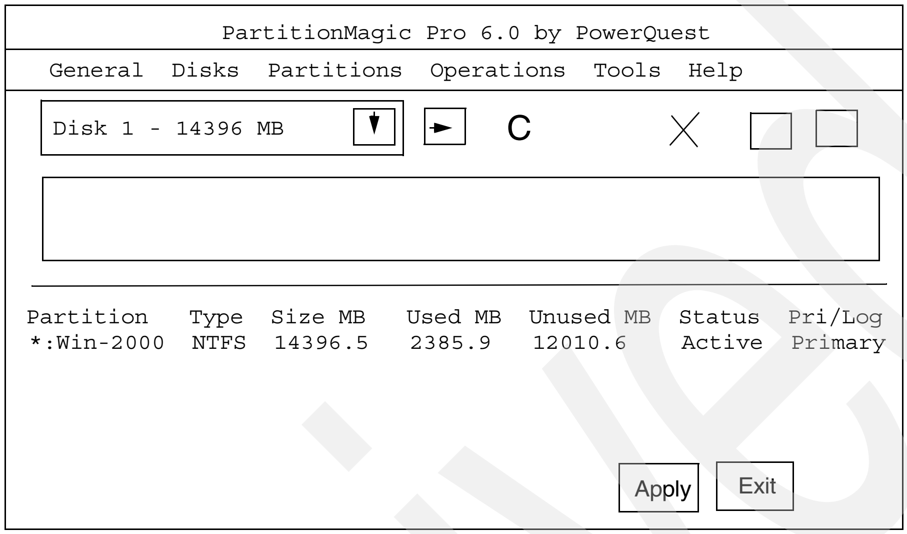


Figure 2-2 PartitionMagic display

This is done by first selecting the line describing the Windows partition (and leaving this selection highlighted), selecting **Operations-> Resize**, and entering `8000.0` as the New Size. PartitionMagic then updates the display to show a smaller partition for Windows and a new line for unallocated space.

PartitionMagic does not actually make disk changes until you click **Apply**. It queues the changes until that time, but reflects them in the displayed information. At this point, you can click **Apply** and make the change. You can then further partition the unallocated space while you are installing Linux. Changing the disk layout takes time. After we clicked Apply, the changes took 17 minutes in this example, using a T20 ThinkPad. Larger disks take longer; a 60 GB drive required 55 minutes in a T23 ThinkPad.

Alternately, you can define your Linux partitions now (before clicking Apply). This is done by highlighting the unallocated space line and then selecting **Operations-> Create**. In the following display, you can specify:

```
Primary Partition
Linux ext2
/boot
80 MB
OK
```

This creates a Linux /boot partition. You can then create a Linux root partition, a swap partition, and so forth. While installing Linux, you can redefine the ext2 file type to be ext3.

The actual dual boot code is installed during the Linux installation, when you specify the boot configuration. At that time, we elected to place the dual boot program in the Master Boot Record (MBR).

3

Linux installation

This chapter describes in cookbook style how we installed Red Hat Linux 9.0. A new Linux user may find the step-by-step instructions useful. An experienced Linux person should review the comments about the choices we made. Some are arbitrary choices, but a few are required.

This description is for installation on a ThinkPad. Installation on a server is similar, although hardware related details (display selection and disk device names, for example) may be different. This description and this whole redbook assume that you use simple Linux files for S/390 volumes. The use of Logical Volume Manager (LVM) and raw devices for emulated volumes is described in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008.

Red Hat Linux installation is easy and quick. If you do not like your first results, simply install it again. If you do not change the disk partitioning during subsequent installation, the contents of other partitions (such as, where emulated S/390 volumes are stored) are not disturbed.

3.1 Basic Red Hat Linux 9.0 installation

We booted from the first Linux CD (using the ThinkPad F12 option to select a temporary boot device). This produced a Red Hat logo display and offered the choice of *graphics mode* or *text mode* installation.[1] We selected text mode, primarily because it was easier to document for this redbook. Then we went through a number of installation prompts. Many of the prompts (in text mode) allow you to use the Tab key to change choices and to move to the OK button.

```
linux text                                   (To select installation mode)

Welcome to Red Hat Linux: OK
Language: English
Keyboard: US
Mouse: Generic 3-button mouse
Installation Type: Custom                    (Do not select mobile computer or server)
Select partition tool: Disk Druid
```

You can select **fdisk** instead of Disk Druid, and you should use whichever one you find most comfortable. If you are not familiar with either, select Disk Druid.

```
Warning: Partition table unreadable......Initialize? Yes
```

You may get this message if you have a completely blank disk.

Be aware that Disk Druid (and **fdisk**) cannot resize partitions. They can only create or delete partitions. If you need to resize partitions, you must do this before you start the Linux installation process.

Our ThinkPad disk had no installed partitions and was 100% available for allocation. You may have disk contents from a previous use or from a dual boot setup. You need to adapt your disk layout to accomplish approximately what we outline here. Using Disk Druid, we allocated four partitions, using ext3 file systems for root and /s390. Our final Red Hat 9.0 display was like this:

```
Device               Start    End     Size      Type       Mount Point
/dev/hda
   /dev/hda1             1     10      73M      ext3       /boot
   /dev/hda2            11    417    3004M      ext3       /
   /dev/hda4           418    452     258M      Extended
      /dev/hda5        418    452     258M      swap       /
   /dev/hda3           453   7752   53894M      ext3       /s390

OK
(If it complains about the /boot size, say "Yes" to continue):
```

We obtained this layout (using a 60 GB disk) by requesting (in order):

- /boot, ext3, 76 MB
- / (for root), ext3, 3000 MB
- swap, 256 MB
- /s390, ext3, all the remaining space

Your *hda* numbers may differ, depending on the order in which you created the partitions. The position of the Extended partition seems to vary, depending on the whims of Disk Druid, but it does not matter. Remember that, if you have an operating system already installed (Microsoft Windows, for example), you see that partition already in place when you start Disk Druid.

[1] You can use the graphical user interface (GUI) installation. The steps are the same as in text mode and the input required is the same as in text mode.

We selected a small, arbitrary size (256 MB) for a Linux swap partition. We may not need this much, although Red Hat documentation recommends swap space equal to memory size (which was 1 GB in our case). 256 MB seemed to be a reasonable compromise and has worked without problems. Red Hat documentation recommends 75 MB for /boot, but the disk allocation rounding method reduced our allocation below this point; we ignored the warning.

The creation of the /s390 partition assumes that standard Linux files are used for emulated S/390 volumes. If you plan to use raw devices, you make this an LVM partition.

It may take some experimentation with Disk Druid to obtain the desired configuration. We suggest that you simply work with the Disk Druid options (Add, Delete, Edit, scroll bars) 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 (especially the Filesystem type window). Also, Disk Druid automatically creates the *extended* partition after you create a few other partitions, and move some of your partitions into the *extended* space.

The /hda terminology is used for an IDE disk drive. If you have a SCSI drive, it is /sda. If your SCSI drive is really a RAID array, and you have already created your RAID partition, you see only a large SCSI drive. That is, a simple RAID array is not visible to Disk Druid (or to Linux).

If we selected Linux components (in a later step) to obtain a small Linux suitable for FLEX-ES, we actually used about 1.9 GB in the root file system. If we elected to *install everything*, we used 4.9 GB in the root file system. The file system should contain a reasonable amount of free space. For this reason, we suggest that you make your root file system at least 3 GB to 6 GB.

If you have more Linux or UNIX experience, you may 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 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. A second hard disk will eventually replace the CD-ROM drive in the Ultrabay of the ThinkPad; 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:

```
Boot Loader Configuration:  grub
Boot Loader Configuration:               (Possibly remove the hdc=ide-scsi parameter)
Boot Loader Configuration:               (We did not use a GRUB password)
Boot Loader Configuration: Red Hat Linux /dev/hda2: OK
Boot Loader Configuration: /dev/hda  Master Boot Record (MBR)
```

The *hdc=ide-scsi* parameter appears if you have a CD R/W drive in the Ultrabay (or an IDE CD R/W unit in an xServer). This parameter causes the Ultrabay to be treated as a pseudo-SCSI device. This is required to support CD writing. However, it does not allow a hard disk to be used in the Ultrabay. You can easily change this later.

If you have a second operating system installed, additional Boot Loader Configuration panels ask which system you want loaded by default and give you a chance to change the displayed name for the other system.

Next you must specify a network address for your system:

```
Network Device: eth0
     [ ] use bootp/dhcp    (Probably deselect this)
     [*] Activate on boot  (You probably want this)
     IP address:           192.169.0.110
     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. These network device parameters are easy to change later.

```
Hostname Configuration:
     Hostname: t23
Security level: No Firewall              (Use the Spacebar to change it)
Language Support: English (USA)          (As appropriate)
Time Zone:                               (As appropriate)
Root 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[2]. You can select the options appropriate for your needs. If you need firewall protection, select **high** and **customize**. In the customization window, specify `24:tcp,555:tcp`.

```
Authentication Configuration:
     [*] Use shadow passwords
     [ ] Enable MD5 Passwords
     [ ] Enable NIS
     [ ] Enable LDAP
     [ ] Enable Kerberos
OK
```

These authentication options are up to you. If you are not certain, we suggest that you use the options shown here. Next you select the general packages you want installed. The easiest option is to select Everything (which is the last option in the list). Be certain your root file system is *at least* 5 GB. If you want to save a little disk space, you can use the choices we used for Red Hat 9.0:

```
Package Group Selection: Customized     (You need to scroll the following list)
   X Windows System
   Gnome Desktop Environment
   Editors (We clicked Details and removed Emacs)
   Graphical Internet
   Text-based Internet
   Office / Productivity
   Server Configuration Tools
   FTP Server
   Network Servers
   Development Tools
   Kernel Development
   Administration Tools
```

[2] FLEX-ES needs to use TCP/IP port 24 on Linux for its Terminal Solicitor function. Almost all FLEX-ES installations require this. FLEX-ES may also use port 555 if you have remote FLEX-ES resources installed. The typical ThinkPad EFS user may not use this, especially for initial FLEX-ES use. If you install firewall functions, be certain that port 24 is not blocked.

```
    System Tools
    Printing Support
OK
```

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. You must include Development Tools and Kernel Development. You may select other options. As far as we know, additional selections have no impact on FLEX-ES operation. The package groups we selected required 1.9 GB disk space.

The installation took 10 to 15 minutes. It called for the second CD and third CD (if needed) during the installation.

```
Boot Disk: No                          (Could not switch to diskette drive)
```

In our case, the Ultrabay was being used for a CD-ROM drive and we could not install a diskette drive to create a boot diskette. If your environment allows you to write a diskette at this time, we suggest that you answer **Yes** to this question.

The installation program then probes the video configuration. On our T23 ThinkPad (IBM 2647-3U5), Linux sensed the following:

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

Monitor Configuration: Unprobed Monitor  <Change>      (Select Change)
Hsync Rate: 31.5 - 48.5                  <Change>
VSync Rate: 50-70
```

We selected the Change option for the monitor. This produced a list with a large number of monitors. We scrolled this list (using the down arrow key) and selected:

```
Generic Laptop Display Panel 1400x1050                (For our T23 model)
OK
```

Your requirements may be different, of course. We accepted the Hsync and VSync rates produced by the system.

```
X Customization: High Color (16 Bit)    <Change>
X Customization: Resolution: 1024x768  <Change>       (Changed to 1400x1050)
X Customization: Default Login: Graphical
```

We changed the video resolution to 1400x1050 for our T23. You should select the correct screen resolution to match your system, of course. We left the 16-bit color selection unchanged, although you may want to optimize it to match your system.

```
Complete:
OK
```

Selecting OK at this point causes the system to reboot. You should remove the second Red Hat CD while you are rebooting.

After the reboot, you are asked to add a new user. You should probably do this, but do not add user ID *flexes*.

Select the **Forward** option in the graphics window and change your system date and time if necessary. After selecting Forward again, our system offered to test our sound card. (This produced no useful results.) Selecting Forward again, the system offered to register our machine with the Red Hat Network.

```
   Yes, I would like to register with Red Hat Network.
X  No, I do not want to register my system.
```

If you elect to register your system, *do not* download updates and fixes unless you have independent information that the updates will not affect FLEX-ES installation.

Selecting Forward again produced a panel that offered to install additional software. This is not needed, and you can select Forward again. This produced the Finished Setup notice and selecting Forward produced the standard login screen.

We logged into Linux as root and examined our filesystem sizes. After **gnome** starts, click the right mouse button and select New Terminal to obtain a command-line window.)

```
# df -h                                  (display file system usage)
   Filesystem          Size   Used   Avail   Use%   Mounted on
   /dev/hda2           2.9G   1.4G   1.3G    51%    /
   /dev/hda1            71M   9.2M    58M    14%    /boot
   none                503M     0    503M     1%    /dev/shm        (swap)
   /dev/hda3            52G    33M    49G     1%    /s390
```

As seen, the Linux selections we chose used 1.4 GB in our root partition, leaving a considerable amount of free space. There is nothing in the /s390 file system except the inodes; it is not clear how the **df** command computes its displayed sizes.

3.2 Installation notes

The following notes may be of interest.

ThinkPad clock accuracy

An emulated S/390 under FLEX-ES on a Linux base on a ThinkPad can cause the ThinkPad clock to lose time if the emulated S/390 CPU is heavily loaded. This was first observed with Red Hat Linux 8.0 and is also present in Red Hat 9.0. This effect appears to be due to the Linux software for ThinkPad power management. You can correct this by removing the power management function in Linux. You need to use an *apm=off* parameter during Linux boot up. One way to do this is to edit /boot/grub/grub.conf and use the line:

```
kernel /vmlinuz-2.4.20-8 ro root=LABEL=/ apm=off
```

Here, we added the *apm=off* parameter to the existing line. A side effect of this change is that the *system shutdown* function of Linux no longer turns off the power for the ThinkPad.

Second hard disk on ThinkPad

If you install Red Hat Linux using a CD-RW (or DVD/CD-RW) drive in the Ultrabay, the installation process detects that you are using a device with *write* capability. Linux uses a pseudo-SCSI interface for CD write support. In this case, it automatically includes a boot parameter that causes the Ultrabay to be treated as a pseudo-SCSI device. A normal ThinkPad hard disk mounted in the Ultrabay does not work when the system is booted with this parameter. The parameter is *hdc=ide-scsi*, and is located in /boot/grub/grub.conf. You can remove the parameter, or create two boot selections so that you can choose, at boot time, whether to keep the parameter (allowing CD write operations) or remove the parameter (allowing use of a hard disk in the Ultrabay). Here is a grub.conf example to do this:

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-8) (CD rw)
      root (hd0,0)
```

```
      kernel /vmlinuz-2.4.20-8 ro root=LABEL=/ apm=off hdc=ide-scsi
      initrd /initrd-2.4.20-84.img
   title Red Hat Linux (2nd HDD)
      root (hd0,0)
      kernel /vmlinuz-2.4.20-8 ro root=LABEL=/ apm=off
      initrd /initrd-2.4.20-8.img
```

Your **grub** parameters may be slightly different. Notice that we included the *apm=off* parameter in both cases. When you boot the system the next time, you can choose between *Red Hat Linux (2.4.18-14) (CD rw)* and *Red Hat Linux (2nd HDD)*, the first one being the default (due to *default=0*). If you create an SMP kernel, as described in the following section, you have more boot choices.

SMP kernel

If the installation process detects that you are installing on a server with multiple processors, it installs an SMP kernel. If you intend to use FLEX-ES on an SMP machine, you must rebuild the kernel as described in 3.3, "Kernel rebuild" on page 30. The stock SMP kernel does not work with FLEX-ES.

Network services

After the installation outlined above, you can log into the system and **ping** another host on your subnet.[3] If the gateway IP system you specified (if any) is functional, you may be able to access remote hosts.

The installation options we selected installed **telnet** and **ftp**. You can Telnet to another machine but other systems cannot Telnet to your machine. Likewise, you can FTP to another machine but they cannot FTP to you. To enable the Telnet and FTP daemons, select **Main menu -> System Settings -> Server Settings -> Services** and enable **telnet** and **vsftp** in the Services menu. (Red Hat installed **vsftpd** as the FTP server when we selected Everthing in the Package Group Selection list.)

On an earlier Red Hat release, we discovered that an unattached LAN port may cause generally sluggish operation of Linux. In one case, we used a token-ring PCMCIA card for LAN connectivity and 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.[4] 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 some of our changes effective.

Token-ring support

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

[3] 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.
[4] While testing LAN operation on a server with multiple Ethernet interfaces, 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.

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).[5]

System freeze

We had a minor problem with earlier Red Hat releases used on T20 and T23 ThinkPads. We do not know if the problem exists for other ThinkPads or other releases. By default, **gnome** invokes a *screen saver* function after a period of no keyboard activity. This function uses a random series of different screensaver programs for the screen. At least one of these patterns caused the system to freeze.

We have not seen this freeze with Red Hat 9.0, but we prefer not to have a screen saver function. We removed it by selecting the **Start Here** icon (on the gnome desktop). We then selected **Preferences**, **Screensaver**, and **Mode: Disable Screen Saver**.

Boot diskette

If you did not create a boot diskette during Linux installation, we suggest you create one now. It is useful if something overlays the master boot record on your hard disk. Log into Linux as root, insert a blank diskette,[6] and enter this command:

```
# /sbin/mkbootdisk  2.4.20-8
```

This command places the stock Red Hat 9.0 kernel on the diskette. It appears that this command only works with vmlinuz kernels, not bzImage kernels. You can look in /boot to determine exactly which vmlinuz kernel name to use. We had /boot/vmlinuz-2.4.20-8 and this accounts for the operand for the **mkbootdisk** command.

Ownership of /s390

We created a filesystem with mount point /s390. This file system is intended to hold our S/390 emulated volumes when they are installed. For this plan to work with FLEX-ES, the /s390 mount point needs to be owned by user *flexes*. However, user flexes is not created until we install FLEX-ES. We must remember to change ownership of /s390 after installing FLEX-ES.

3.3 Kernel rebuild

You may need to build a new Linux kernel for any of several reasons. These include:

- You have an SMP system. The stock SMP kernel with Red Hat releases cannot be used with FLEX-ES.[7]
- You have large PC memory that requires special Linux handling, including selection of the HIGHMEM I/O option.
- You have a LAN adapter not supported by your existing kernel.
- You have a stock kernel that does not run FLEX-ES.
- If you have updated the ips driver (for ServeRAID).
- If you want the latest LVM support.

[5] This patch will be incorporated into the Linux kernel starting with release 2.4.21.
[6] You may need to stop Linux, shut down your ThinkPad, and swap the CD-ROM drive for a diskette drive.
[7] This may change in the future, of course, but this statement is true for all Red Hat releases up to the time of writing.

FSI provides Linux kernel source trees containing the options and FSI patches required for FLEX-ES. FSI has provided these at the 2.4.17, 2.4.19, 2.4.20, and 2.4.21 kernel levels. The following description assumes that you are using the 2.4.21 level, but should apply to other levels with minor file name changes.

The FSI source tree already contains the FSI patches necessary for FLEX-ES SMP support. It may not have the options you need for other cases, such as certain LAN adapters or large memory. In these cases, you need to make additional kernel parameter changes as you work through the installation steps. Kernel builds made with source trees not supplied by FSI are not supported, especially for SMP systems.

> **Important:** Your kernel source code must match your kernel or you will be unable to install and use FLEX-ES.

The following material was developed by Gary Eheman and Michael Ryan, of Fundamental Software, Inc.

1. If you plan to install the latest LVM support, obtain the necessary updates before starting your kernel build. To obtain the updates, see 3.4.2, “LVM updates” on page 35.
2. If you plan to install the latest ServeRAID support, obtain the necessary modules before you start your kernel build. This is briefly discussed in 2.3.2, “BIOS and ServeRAID adapter updates” on page 21.
3. Obtain a copy of the 2.4.21 Linux kernel source tree, with the proper FSI patches and options installed.[8] This is typically in the form of a compressed tar file. We assume that it is named linux-2.4.21.pset.tar.gz. For our examples, we assume it is on CD. Alternately, you may download it.
4. Log into your Linux system as `root`. All the following work is done as root.
5. Untar the source tree:

```
# mount /dev/cdrom /mnt/cdrom            Or let Linux automount do the mount
# cd /usr/src                            Put the source tree here
# tar -zxvf /mnt/cdrom/linux-2.4.21.pset.tar.gz      Untar into current directory
```

6. Verify that a directory named linux-2.4.21.pset now exists, and change the “linux” symbolic link to point to this directory. If the “linux” symbolic link does not exist, add it as shown (it did not exist in our RH9.0 system):

```
# cd /usr/src                            Work in this directory
# ls -al                                 See if linux-2.4.21.pset is there
# ls -ltr                                See if “linux” symbolic link exists
# rm linux                               Delete “linux” symbolic link if present
# rm linux-2.4                           Delete this symbolic link
# ln -s linux-2.4.21.pset linux          Make a new “linux” link
# ln -s linux-2-4-21.pset linux-2.4      Make a new link
# cd linux                               Move to this directory
# ls -al                                 Verify that you see files and directories
```

7. Force a new kernel name to avoid duplicate names. The constant we use, flexes1, is arbitrary. If you make multiple new kernels (and want to preserve all of them), you can increment this name to flexes2, flexes3, and so forth:

```
# cd /usr/src/linux                      If not already in this directory
# vi Makefile                            Use a different editor if you prefer
...
VERSION = 2
PATCHLEVEL = 4
```

[8] Your FSI Business Partner, or whomever you obtained your FLEX-ES software from, should help you with this.

```
SUBLEVEL = 21
EXTRAVERSION = flexes1                  <== Change this line
```

Notice the space before and after the equal sign. Save the altered Makefile and exit from the editor.

8. Save a copy of the .config file:

```
# cd /usr/src/linux                     If not already in this directory
# cp .config flexes.config              Keep the FSI configuration parameters
# cat README | less                     (Might be interesting, but you are
                                         not required to read it now.)
```

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

```
# make clean                            Expect about one screen of output
# make mrproper                         Expect several screens of output
```

10. At this point, make the kernel modifications for new ServeRAID modules. These steps are described in 3.4.1, “ServeRAID updates” on page 35. If you use this step, you must coordinate it with the updating of ServeRAID firmware as described in 2.3.2, “BIOS and ServeRAID adapter updates” on page 21.

11. At this point, perform kernel modifications for new LVM modules. These steps are described in 3.4.2, “LVM updates” on page 35.

12. Make any kernel configuration changes that are needed:

```
# make xconfig                          Several compiles, much output
```

13. This step should result in a GUI screen with many buttons:

    a. Click **Load Configuration from File** (lower right part of the display).

    b. For the file name, enter `/usr/src/linux/flexes.config` and click **OK**.

    You need to load the FLEX-ES starting configuration file before you start looking at the options. The following table has abbreviated titles in some of the entries to avoid using a very small type font. Many of the buttons on this screen open windows that have additional buttons that open other windows, and so forth. It is safe to “look around” in the options and windows. Do *not change* anything unless you are fairly certain you know what you are doing.

Code maturity level options	SCSI support	File systems
Loadable module support	Fusion MPT device support	Console drivers
Processor type and features	IEEE 1394 (Firewire) support (EX	Sound
General setup	I2O device support	USB support
Memory Technology Devices (MT	Network device support	Bluetooth support
Parallel port support	Amateur Radio support	Kernel hacking
Plug and Play configuration	IrDA (infrared) support	Library routines
Block devices	ISDN subsystem	
Multidevice support (RAID, LVM)	Old CD-ROM drives (not SCSI)	Save and Exit
Networking options	Input core support	Quit Without Saving
Telephony Support	Character devices	Load Configuration from File
ATA/IDE/MFM/RLL support	Multimedia devices	Store Configuration to File

14. This interface lets you make kernel option changes. Again we stress that you should *not change anything* unless you know what you are doing. The FSI patches and options are already applied, and the kernel is configured for FLEX-ES SMP use. A few options you may *need* to change are:

- Large memory. Select **Processor type and features**. Click the button labeled **High Memory Support**. This should open a menu with three options: off, 4 GB, or 64 GB. If your machine has less than 1 GB, select the **off** option. For 1 to 4 GB memory, select the **4GB** option. If your machine has more than 4 GB memory, select the **64 GB** option. The 64 GB option produces a kernel that works only on Intel IA32 processors.[9]
- HIGHMEM I/O support. If you select the 4 GB or 64 GB large memory option, you should select this also, using the **y** option. It is on the Processor type and features panel.
- Newer IBM processors may have 1000BaseT LAN adapters. These may not be enabled with the default kernel options. Select **Network device support** on the GUI kernel configuration panel, and then on the next panel, select **Ethernet (1000 Mbit)**. The final panel lists the 1000 megabit devices supported. Recent IBM systems may have a *Broadcom Tigon3* adapter (servers) or an *Intel PRO/1000 Gigabit Ethernet* (T40 ThinkPads) and you see it listed on the panel. Select **m** and then **OK** for support.[10] You can select more than one adapter option.
- If you have a built-in Ethernet adapter using the Intel PRO/100 VE Network Connector (used on T23 ThinkPads, for example) or the Dual-port IBM 10/100 adapter, select **Network device support** and then select **Ethernet (10 or 100Mbit)**. In this option, set Ether Express pro/100 (e100, Alternate Intel Driver) to **m**.

15. After you make any kernel option changes, complete these steps:

    a. Click **Store Configuration to file** and select a new name. We used **/usr/src/linux/flexes.config2**. You may not need this file, but a backup may be useful sometime.

    b. Click **Save and Exit**.

16. Run the actual kernel-building steps. These steps take some time and produce many screens of output, including various warning messages.

```
# make dep                              To make kernel dependencies
# make bzImage                          Use an upper case “i”
# make modules                          Longest step
# make modules_install
```

17. Copy the bzImage and System.map files to /boot:

```
# cp /usr/src/linux/arch/i386/boot/bzImage  /boot/bzImage-2.4.21flexes1
# cp /usr/src/linux/System.map  /boot/System.map-2.4.21flexes1
```

18. There is 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-2.4.21flexes1 System.map       New symbolic link
```

19. Make a ram disk image.

```
# cd /boot                              If not already in this directory
# /sbin/mkinitrd -v  initrd-2.4.21flexes1  2.4.21flexes1
```

This command looks in various standard directories for modules related to 2.4.21flexes1 (the second operand). Then it builds a ram disk using the name you specify in the first operand. If you built a previous image with the same name, you must delete it before you run `mkinitrd` again.

[9] We used ThinkPads with 1 GB memory using kernels with the **off** option. This may produce slightly degraded performance, but it was not observable.
[10] This simply adds support to the kernel. It does not require you to have the adapter installed.

20. Update /boot/grub/grub.conf to include the new kernel. Ignoring the comment lines, our original /boot/grub/grub.conf looked like this:

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-8)
      root (hd0,0)
      kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
      initrd /initrd-2.4.20-8.img
```

We edited the file to look like this:

```
default=0
timeout=10
splashimage=(hd0,0)/grub/splash.xpm.gz
title Red Hat Linux (2.4.20-8)
      root (hd0,0)
      kernel /vmlinuz-2.4.20-8 ro root=LABEL=/
      initrd /initrd-2.4.20-8.img
title New kernel (2.4.21)
      root (hd0,0)                      (Copy numbers from original grub entry)
      kernel /bzImage-2.4.21flexes1
      initrd /initrd-2.4.21flexes1
```

Save the file and exit from your editor. Your *hd0,0* parameters may be different, especially if you have a dual boot system. Copy the parameters from the first title line in /boot/grub/grub.conf. (Note that there may be symbolic links from other names to /boot/grub/grub.conf; you can work through any of these links.)

21. Shut down your Linux cleanly and reboot.

22. When the Red Hat splash screen appears, you should see two options: one for the original kernel and one for your new kernel. Use the down arrow key to select the new kernel. You need to do this fairly quickly because your grub parameters specify a 10 second timeout with the default selection of the original kernel. You can change these in /boot/grub/grub.conf:

```
default=1                  (Make second kernel the default)
timeout=12                 (Give user 12 seconds to manually select)
```

Note that the "title" sections in the grub configuration are counted starting with zero.

23. Verify that you are running with your new kernel:

```
# cd /proc
# cat version
```

The text in this line contains "2.4.21flexes1" if you are running with your new kernel.[11]

You can build and test an SMP kernel on a single processor system. We performed the complete kernel rebuild just described on ThinkPads and on an xSeries SMP.

Rebuild FLEX-ES module

If you previously installed FLEX-ES, you must rebuild one of the modules for your new kernel. (Be certain you are running with the new kernel when you do this!) Do the following under the *flexes* user ID:

```
$ cd /usr/flexes/modules                 Do this as flexes, not root
$ make clean
$ make all
(reboot to pick up the new msgmgr module)
```

[11] The "flexes1" portion of the name reflects whatever constant you specified in the EXTRAVERSION name.

3.4 Additional kernel steps

You can insert these optional steps into the appropriate place when you rebuild the kernel.

3.4.1 ServeRAID updates

For our sandbox projects used when writing Redbooks, we do not use the Linux ServeRAID management programs. We build a RAID array and create a single logical partition by using the stand-alone programs on the ServerGuide CD. We then install Linux with no additional steps. This has worked for our sandbox projects, but does not provide the full RAID management needed for a production FLEX-ES system. A production installation should include the most current ServeRAID updates and Linux support.

You should have a ServeRAID CD ROM and install BIOS and ServeRAID firmware updates from the same CD. That is, the BIOS, firmware, and Linux updates should all be at the same level. There are extensive instructions on the ServeRAID CD for installing the upgrades and Linux commands. If you follow the instructions, you can add these upgrades and commands after Linux is installed.

You can shorten the process by installing the upgrades at the same time you are rebuilding the kernel. FSI uses the following steps, at the appropriate time, to do this:

```
# mount /dev/cdrom /mnt/cdrom                     (Or let automount do it for you)
# cd /mnt/cdrom/linux
# cp cmdline/ipssend /usr/local/sbin              (Get the command line utility)
# chmod 500 /usr/local/sbin/ipssend               (Only root can use it)
# rpm --install manager/RaidMan-*                 (Install ServeRAID manager program)
# mkdir /root/ips/                                (Make a working directory)
# cp /mnt/cdrom/linux/driver/ips-610.tgz /root/ips
# cd /root/ips
# tar -zxf ips-610.tgz
  (We need only the ips.c and ips.h files.  Ignore the patch files)
# mv /usr/src/linux/drivers/scsi/ips.c  /usr/src/linux/drivers/scsi/ips.c.orig
# mv /usr/src/linux/drivers/scsi/ips.h  /usr/src/linux/drivers/scsi/ips.h.orig
# cp /root/ips/ips.? /usr/src/linux/drivers/scsi/
# ls -l /usr/src/linux/drivers/scsi/ips*
  (Verify that ips.c, ips.h, ips.c.org, and ips.h.org are present)
# cd /usr/src/linux                               (Correct place for next kernel step)
```

After you do this, return to the normal kernel building process, where the next step is to **make xconfig**.

3.4.2 LVM updates

The LVM modules in Red Hat 9.0 work well. However, for the best possible operation of a production FLEX-ES system, should install the latest LVM version. At the time of writing, this was 1.0.7. You must integrate the new modules into a new Linux kernel as part of the kernel building process. To install the latest version, obtain the material before you start building a new Linux kernel:

1. With a browser (which need not be running under Linux), go to:
   http://www.sistina.com
2. Click **Products**.
3. Click **Logical Volume Manager for Linux (LVM)**.
4. Click **Download**.
5. Click **Download LVM 1.0.7** or later, if available.
6. Select **lvm_1.0.7.tar.gz (372 KB)**. We copied this to a DOS diskette.

Unpack the tar file in the /root directory of your Linux system:

```
# mcopy a:lvm_1.0.7.tar.gz /tmp/lvm_1.0.7.tar.gz     (Assume it is on a DOS diskette)
# cd /root
# tar -zxvf /tmp/lvm_1.0.7.tar.gz
```

At the appropriate place in your kernel build (after **make clean** and **make mrprogram**), run the following commands:

```
# cd /root/LVM/1.0.7
# ./configure
# make
# cd /root/LVM/1.0.7/PATCHES
# make
# cd /usr/src/linux
# patch -p1 < /root/LVM/1.0.7/PATCHES/lvm-1.0.7-2.4.21flexes1.patch
patching file include/linux/lvm.h
...
patching file drivers/md/lvm-fs.c
# cd /root/LVM/1.0.7
# make install                                 (Install the new LVM)
# cd /usr/src/linux                            (Correct place for next kernel step)
```

The file name in the PATCHES directory reflects the name of the kernel that you are building and may differ from the name shown here. After doing these steps, return to the normal kernel building process, where the next step is to **make xconfig**.

3.5 Additional Linux steps

Red Hat 9.0 *may not* automatically use DMA for disk accesses on the ThinkPad. This varies for different ThinkPad models. This has a drastic impact on performance. You should perform the following actions after logging into the system as *root*:

```
# hdparm -d /dev/hda                    (Check DMA usage)
    using_dma = 0 (off)                 (This is bad)
    using_dma = 1 (on)                  (This is good)
(If using_dma = 0, then change it with the following command)
# hdparm -d1 /dev/hda                   (Change to dma = on)
# hdparm -d /dev/hda                    (Verify that it worked)
```

This should make an immediate change in disk performance. You can use **hdparm -d0 /dev/hda** to turn off DMA again. You can run a little benchmark (and try it with DMA on and off) by entering:

```
# hdparm -t -T /dev/hda
```

You can make the change permanent by editing /etc/rc.d/rc.local and including this line in the file:

```
hdparm -d1 /dev/hda
```

We checked our CD drive in the Ultrabay (**hpdarm -d /dev/hdc**) and found that it already had DMA turned on.

The only diskette drive we found for a T40 ThinkPad was connected through the USB ports. The simple DOS diskette commands (**mdir**, **mcopy**) could not find the drive.[12] We used it via a mount command:

```
# mount /dev/sda /mnt/floppy            (Mount diskette as a file system)
```

[12] You may be able to use these commands by editing /etc/mtools.conf and making appropriate changes.

Remember to **umount** a diskette when you are finished with it.

3.5.1 Red Hat Linux updates

Red Hat offers many fixes and updates through the Red Hat Network. The Linux **up2date** tool provides an interface to obtain these updates. We make the following recommendations:

- Do not obtain kernel updates or replacements through this process unless you are willing to verify that the new kernel works properly with FLEX-ES. Remember that you must keep the kernel source tree in your system synchronized with the working kernel. You must also reinstall the FLEX-ES msgmgr module after any kernel change. Also remember that the FSI kernel source includes FSI patches that are needed for use of FLEX-ES on an SMP system.
- Obtaining and applying fixes and updates for various Linux packages is acceptable provided they have no kernel dependencies that may require a kernel change.

The owner of a production system may want to work with **up2date** and obtain updates in an orderly manner followed by appropriate system testing. For sandbox systems, we suggest that you do not obtain updates unless needed for a specific problem.

4

FLEX-ES installation

This chapter describes the installation process for FLEX-ES. System definitions, resource definitions, and operation are described later. The same installation process is used for ThinkPad and xSeries 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 the X Window System 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? If not, you need to resolve these problems now.
- If you are using an SMP system, have you built a new kernel using a source tree provided by FSI?
- Have you installed the required Red Hat components, such as *Development Tools* and *Kernel Development*? (If you elect to install Everything during Red Hat installation, this is not an issue.)

This chapter describes the installation of FLEX-ES Release 6.2.14. The general process should match any recent FLEX-ES release.

> **Attention:** You *must* install and use FLEX-ES with a Linux kernel that matches the Linux kernel source tree in your system. If you have not installed new kernel source, or have not downloaded anything that changes the stock kernel, this should not be a concern. (The msgmgr module of FLEX-ES is sensitive to this pitfall.)

4.1 FLEX-ES installation steps

FLEX-ES can be delivered a number of ways:

- FTP over the Internet
- CD-ROM
- Preinstalled by a Business Partner

The FLEX-ES package is not large (about 4 MB), and an FTP download is easy. 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 is normally shipped with your system. For our examples, we assume that you have FLEX-ES on a CD.

The FLEX-ES distribution file is normally in tar format. You want to untar the file and place the results in the /tmp directory:

```
# mount /dev/cdrom /mnt/cdrom              (Linux may do the CD mount automatically)
# cd /mnt/cdrom
# ls
   flexes-6.2.14.i386.rpm.tar                      (Or a name similar to this)
# cd /tmp                                          (Untar into /tmp)
# tar xvf /mnt/cdrom/flexes-6.2.10.i386.rpm.tar      (If working from CD)
# umount /dev/cdrom                                (CD not needed any more)
```

If you are downloading FLEX-ES, 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. The `tar` command should produce several files in /tmp:

```
# ls
flexes-6.2-14.i386.rpm                            (Primary FLEX-ES modules)
msgmgr-6.2-14.i386.rpm                            (FLEX-ES kernel communication)
ftlib-6.2-14.i386.rpm                    (Source for creating programs using FakeTape)
```

Again, your exact file names may differ but the general names are as shown here. You must be *root* to use these `rpm` commands.

```
# cd /tmp                                         (Location of FLEX-ES rpm files)
# rpm -i flexes-6.2-14.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 later set the password manually)
# rpm -i msgmgr-6.2-14.i386.rpm
gcc .........                                     (Inspect any messages carefully)
ld ..........                                     (Inspect any messages carefully)
# rpm -i ftlib-6.2-14.i386.rpm
```

These three `rpm` commands perform the basic FLEX-ES installation. Installation of msgmgr involves compilation and link steps. These must run without errors. You see a long `gcc` command and a long `ld` command displayed. There should be no error messages associated with these commands. If there is an error here, FLEX-ES is unable to load or start msgmgr later.

If you chose to install "Everything" when you installed Linux, the system has the /usr/bin/expect file and password "abcdef1" is automatically set by the rpm processing. If you selectively installed a smaller Linux system, the /usr/bin/expect file is probably not present and you need to manually set a password for user ID *flexes*. This is easy to do and no side effects are involved.

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*. In some cases the FLEX-ES install process cannot set the password. If this happens, you can use:

```
# passwd flexes
abcdef1                          (After prompt to enter password)
BAD PASSWORD: it is too simplistic/systematic      (You can ignore this warning)
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*.

4.1.1 Default PATH

We recommend a minor step that we think makes operation easier:

```
# cd /usr/flexes
# mkdir rundir                                 (Conventional location for FLEX-ES files)
# chown flexes:flexes rundir                   (Make flexes the owner)
# vi /root/.bash_profile
   ...
   ...
   USERNAME=”root”
   PATH=$PATH:/usr/flexes/bin                  <=== Add this line
   export USERNAME BASH_ENV PATH
```

Adding /usr/flexes/bin (where all the FLEX-ES executables are located) to PATH makes operation easier.[1] IBM 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.

4.1.2 Ownership of /s390 mount point

For our disk allocation plan, we created a file system with mount point /s390. This file system is used for our emulated S/390 disk volumes. This mount point must be owned by user *flexes*, and this userid was created automatically by the FLEX-ES installation process. Following our instructions, the /s390 mount point was created during Linux installation and is owned by *root* at that time. This needs to be changed. Working as *root*, enter:

```
# chown flexes:flexes /s390
```

4.1.3 Location of x3270 program

Most Red Hat installations contain an x3270 program. FLEX-ES installs a different x3270 program that is found in /usr/flexes/bin. Using the wrong x3270 results in this message when you attempt to start x3270 when the **flexes** program[2] is running:

```
Error: app-defaults version mismatch: want 3.2.18, got 3.1.1.0
```

We need to ensure that the correct x3270 program is used. A way to do this is to always use a full path name for it, but this is awkward. Another way is to rename the Red Hat x3270 to a different name. If /usr/flexes/bin is added to the standard search path, this causes the FLEX-ES version of x3270 to be found. You can enter:

```
# cd /usr/X11R6/bin                       Where old x3270 is found
# mv x3270 x3270RH                        Rename x3270
```

[1] The /usr/flexes/.bash_profile already contains PATH=$PATH:$HOME/bin where $HOME is /usr/flexes.
[2] This is the main FLEX-ES program that provides S/390 emulation.

The FSI version of x3270 was modified to allow session binding via the device names that you place in the FLEX-ES definitions (or in a FLEX-ES **mount** command).

4.2 Installing the FLEX-ES license key and dongle

Connect the dongle to the USB port. You can check the dongle with the following command:

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

On our system, the **cat** 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 ( 0%), #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 Ivl=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=cddc 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 Ivl= 0ms
E:  Ad=01(O) Atr=02(Bulk) MxPS=  16 Ivl= 0ms
```

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 should 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 an 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 needed
#
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 need to copy it from there, for example:

```
(shut down Linux if you need to swap Ultrabay devices)
(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:yourkey987.6.2.14.key  /var/adm/flexes/.flexeslicense
(done)
          (if the diskette is not in DOS format, try the following)
# mount /dev/fd0 /mnt/floppy                (For a T40, it is /dev/sda)
# ls -al /mnt/floppy
         (if the key file is displayed, copy it with the following command)
# cp /mnt/floppy/yourkey987.6.2.14.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 do 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 that is provided with the Red Hat 9.0 system should recognize and use the USB dongle, as described. However, if you build your own kernel, it may not use the dongle correctly. If you build a kernel, configure the USB Serial Converter as a *module* and say *yes* to USB Generic Serial Driver support. The kernel source trees provided by FSI already have this configured.

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. If the license key file (before you copy it to the /var/adm/flexes directory) has a name like xxxx6.2.14 then that license is for FLEX-ES Release 6.2.14. It can be used for any earlier FLEX-ES release *but not for a later release*. If you want to use a later FLEX-ES release, you need a new license key file.

4.3 Reboot

Installing (or updating) FLEX-ES changes modules that are loaded during Linux booting. You should reboot after you install or update FLEX-ES.

4.4 Installing FLEX-ES upgrades

Installing a new FLEX-ES release under Linux involves a few **rpm** commands. You must be *root* to use these **rpm** commands.

```
# mount /dev/cdrom /mnt/cdrom                   (Or let automount do it)
# ls /mnt/cdrom
   flexes-6.2.14.i386.rpm.tar                   (Or a name similar to this)
# cd /tmp                                       (Untar into /tmp)
# tar xvf /mnt/cdrom/flexes-6.2.14.i386.rpm.tar     (If working from CD)
               (This untars the new release into your /tmp directory)
# umount /dev/cdrom                             (Not needed any more)
# rpm -e ftlib                                  (Remove existing FLEX-ES installation)
# rpm -e msgmgr
# rpm -e flexes
# cd /tmp                                       (If not already there)
# rpm -i flexes-6.2-14.i386.rpm
```

```
# rpm -i msgmgr-6.2-14.i386.rpm
                                                  (Check these messages!)
# rpm -i ftlib-6.2-14.i386.rpm

(Reboot Linux)
(Use cfcomp to recompile your FLEX-ES definitions)
```

The old FLEX-ES packages are 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.

You can use the **rpm -U** option for upgrades, instead of the separate **rpm -e** and **rpm -i** commands. We show the separate commands to more clearly illustrate that an upgrade is a complete replacement of the FLEX-ES packages. Using the **-U** option is a more normal way to install an upgrade.

Also, you may need to recompile your system and resource definitions (using **cfcomp**) after installing a new FLEX-ES release.[3] This takes a few seconds and we suggest that you do it after you install any new FLEX-ES release. You may also need to reinstall any x3270 keyboard modifications you made.

After upgrading FLEX-ES, you should reboot Linux. One of the FLEX-ES modules is automatically loaded during booting and you want to have the new version of this module active.

[3] If your system starts **resadm** automatically during the boot processes, you should reboot *after* recompiling your resource definitions. The setup described in this redbook does not start **resadm** automatically, but many production FLEX-ES systems installed by FSI Business Partners do start **resadm** automatically.

5

S/390 operating system installation

In principle, you can install a S/390 operating system on your EFS system using any of the distribution media and logical packaging available within IBM (for IBM internal users) or from IBM.[1] These include, but are not limited to:

- Tapes in dump/restore format, provided you also have an appropriate restore utility. This utility might reside on an already-installed S/390 operating system or you may be able to IPL it from a stand-alone tape. Of course, you need a tape drive on your EFS system to use this method.
- Tapes in typical IBM distribution formats, such as ServerPacs. These assume you already have a working S/390 operating system on your system. This working system might have come from a stand-alone restore operation.
- CDs containing images of S/390 volumes that contain a S/390 operating system. IBM has made several of these available to members of the S/390 PartnerWorld for Development organization who obtained their systems through this program. Other CD-resident distributions have been created for IBM internal use. There are three general formats of these CDs (all of which are compressed in some way so that a 3390 volume fits on a CD):
  - Zipped AWSCKD: This format was originally created for IBM P/390 systems and has been used for IBM Multiprise® 3000 and other systems. It is the current format for the AD CD-ROM systems described below. FLEX-ES provides a utility to convert this format into its own emulated disk format.
  - Compressed tar or gzip: These are copies of emulated S/390 volumes from a working Linux-based FLEX-ES system. The Linux file containing the emulated volume is tarred, compressed, or both and then written to CD. These are easy to create but are not directly suitable for restoring to raw disk devices.
  - Compressed `ckdbackup` files: These can be restored as simple Linux files or as raw disk device files. This is the recommended format for future distributions.

[1] This statement assumes that you have a tape drive on your EFS system that is compatible with the IBM-provided media, of course.

- Emulated tapes (on CD or other media) can be used for distributions or stand-alone utilities. FLEX-ES supports FakeTape,[2] AWSTAPE, and OMA/2 formats. OMA/2 is an optional distribution format for z/VM.
- It is possible to FTP or otherwise transfer images of S/390 volumes from other Linux-based FLEX-ES systems. This can be a practical method for a classroom setup, for example.

This chapter describes installation based on four of these methods, with emphasis on the AD CD-ROM systems because that appears to be the most common distribution used by EFS owners. An expanded description of ckdbackup for creating and installing distributions appears in Chapter 5 in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008.

Required documentation

We assume that an operating system distribution provides some documentation. For example, a usable z/OS system contains an IODF (or several IODFs) that determine which S/390 addresses (device numbers) are used for various types of devices. If we place our 3390 IPL volume at address 0A80, this is because the IODF specifies address 0A80 is for a 3390 device type. There is *no way to guess* such addresses. IODFs are typically built with a large number of addresses for various device types. You can select among appropriate addresses for your EFS operation, but you require a list of addresses “generated” in your system.

z/VM is more flexible and automatically senses the addresses of more modern device types.

You may also need to know the IPL parameter needed by your S/390 operating system.

Linux file names

An emulated S/390 volume is a simple Linux file.[3] As such, it has a Linux file name. These files can be renamed with the **mv** command, copied to other locations with **cp**, and treated as normal (but large) Linux files. All our examples place the emulated volumes in a file system with mount point /s390. We use the volser of the emulated 3390 volume (in upper case) as the Linux file name. For example, if we have a 3390 with volser SYSRES, the Linux file containing this volume would be /s390/SYSRES. We find this convenient, but it is completely arbitrary. You can place the files anywhere in the Linux file system, using any file names you like. You must have the correct file names in the FLEX-ES definition files.

5.1 AD CD-ROM distributions

In practice, most of the IBM PartnerWorld for Developers (PWD) 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 PWD members who obtained systems through the PWD organization.

An AD system provides a very easy way to install a useful z/OS system. You could build 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. It would also require a hardware system with tape drives compatible with the ServerPac media.

In general, a z/OS AD system is a rather straightforward implementation of z/OS. The experience and results of using it on an EFS system is about the same as using any other

[2] FakeTape is a trademark of Fundamental Software, Inc.
[3] This section does not apply if you are using raw devices for emulated volumes.

straightforward z/OS implementation. The AD CD-ROM systems, as the name implies, are distributed on CD-ROMs. The CD-ROMs are not seen by the S/390; they are processed by Linux programs.

Basic CD-ROM formats

The z/OS volumes (on the AD CD-ROM) are zipped 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 to date 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 is a standard label, a VTOC, probably a VTOC index, and whatever data sets appear on that volume in a 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 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[4]. The two files are usually placed in a single ZIP file.

FLEX-ES formats

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 the AWSCKD format to the FLEX-ES format for emulated CKD drives.

5.1.1 z/OS AD CD-ROM addresses and IPL data

The AD CD-ROM documentation provides the addresses (device numbers) used for various devices. A large number of addresses are provided. We use the following address ranges:

Device type	Addresses
3390 DASD	0A80-0A9F
3270 MVS console	0700
3270 VTAM terminals	0701-071F
3172 (LAN)	0E20, 0E21
3480 tape drive	0560

Based on the documentation and addresses we selected, the IPL information is:

```
IPL address = 0A80   IPL parameter = 0A82CS
```

The AD system documentation indicates that four volumes are required for a minimum IPLable system; the volsers are S4RES1, S4RES2, OS39M1, and S4USS1.[5] To make our system match existing AD system documentation, we installed these volumes at addresses 0A80, 0A81, 0A82, and 0A87. Additional volumes can be installed containing DLIBs, DB2®,

[4] 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.

[5] These volsers are for the z/OS AD release current at the time of writing. Other releases will have different volsers.

CICS, IMS™, and WebSphere®. You should consult the AD CD-ROM documentation for more information about device addresses, volumes available, and IPL options.

5.1.2 Installation tasks

If you let the Linux automount function mount CDs, you can perform the complete AD installation under the *flexes* userid. Installation consists of:

- Inserting a CD in the drive (and waiting for Linux automount to recognize it)
- Using an **unzip** and **ckdconvaws** command, piped together
- Issuing an **umount** command to remove the CD
- Repeating this process for each CD

The automount window is convenient for confirming the contents of the CD. You might close this window before you **umount** the CD.

If loading a CD on your system does not trigger an automount function you can issue the appropriate commands. You may need to do this from a root window, and in this case using two windows (one as root and one as flexes) is convenient:

```
# mount /dev/cdrom /mnt/cdrom                  (Issue from root)
   (Process the CD from a flexes window)
# umount /dev/cdrom
   (Remove the CD)
```

Installing a volume

The output of the **unzip** command can be piped directly to the **ckdconvaws** command:

```
   (Insert CD and wait for automount to recognize it)
$ unzip -p /mnt/cdrom/zos14/s4res1.zip | ckdconvaws -r - /s390/S4RES1 3390-3
FSIDC146 Max head = 14, cyl = 3343, blks = 57
FSIDC180 Cylinder nnn completed in mmm milliseconds
FSIDC190 CKD Conversion Completed (3339 cyls copied)
   (Close the automount window)
$ umount /dev/cdrom
```

The **ckdconvaws** command converts the AWSCKD format (used on the AD systems) to the FLEX-ES format, and writes to the Linux file indicated.[6] In this example, the volume being unzipped is a 3390-3 (and we must tell the **ckdconvaws** program this). The second operand, /s390/S4RES1, is the target location for the Linux file that will emulate the S/390 volume. Enter the command as shown; notice the space before *and after* the single dash in the second part of the command. Use the correct *pipe* symbol (long vertical line, sometimes shown with two sections in the vertical line).

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.)

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.

Repeat the same process for the other CDs that are to be loaded. For the z/OS 1.4s system, we used:

```
CD    CD File.................      Linux File      Disk Model
2     /mnt/cdrom/zos14/s4res2.zip    /s390/S4RES2      3390-3
```

[6] The -r option indicates that **ckdconvaws** should create the output file, if it does not already exist.

```
3     /mnt/cdrom/zos14/os39m1.zip    /s390/OS39M1      3390-3
4     /mnt/cdrom/zos14/s4uss1.zip    /s390/S4USS1      3390-3
```

This produces an IPLable system.

Loading the remaining volumes

There were 14 volumes distributed with the z/OS 1.4s AD system we used. Only four volumes are needed to IPL the system and use TSO. The remaining volumes are:

CD#	volser	3390 model	assumed address	content
5	S4DIS1	3390-3	A85	system DLIBs
6	S4DIS2	3390-3	A86	system DLIBs
7	S4DIS3	3390-3	A88	system DLIBs
8	S4DB21	3390-3	A83	DB2 (including DLIBs)
9	S4CIC1	3390-2	A84	CICS (including DLIBs)
10	S4IMS1	3390-2	A89	IMS (including DLIBs)
11	S4WAS1	3390-3	A8A	WebSphere
12	S4WAS2	3390-3	A8B	WebSphere
13	SARES1	3390-3	A8C	Stand-alone system
14	S4DIS4	3390-3	A8D	system DLIBs

You may elect to install any or all of these additional volumes. If you want to install system PTFs, for example, you almost certainly need the DLIB volumes. The SARES1 volume is also a z/OS 1.4 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 have different volsers involved), but the general principles should remain the same.

FLEX-ES definitions

Two FLEX-ES definitions (and matching shell scripts) are listed in Appendix A, "Sample FLEX-ES operation files" on page 91. The defR14A definition includes only the four volumes needed for an IPLable z/OS AD 1.4s system. The defR14B definition includes all the volumes in the AD 1.4s distribution, plus several local volumes. You *may need to modify* these definitions to match your volumes, IP address, or naming conventions.

Staged loading

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 may need a few staging steps:

1. Install the CD-ROM drive. This may require shutting down and rebooting Linux.
2. 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.
3. Install (**mount**, **unzip**/**ckdconvaws**, **umount**) volumes from CDs to fill the free space.
4. Remove the CD-ROM drive and install the second hard disk. This may involve shutting down and rebooting Linux.
5. Copy the volumes from the internal hard disk to the second hard disk; for example:

   ```
   $ cp /s390/S2DIS1  /s391/S2DIS1
   $ cp /s390/S2DIS2  /s391/S2DIS2
   ```

   This example assumes you are using our naming conventions. These copy commands take several minutes for each file.
6. Delete the volumes from the internal hard disk, for example:

```
$ rm /s390/S2DIS1
$ rm /s390/S2DIS2
```

7. Repeat the process with more volumes from the CDs, if necessary.
8. 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*.

Another way to stage volumes would be to copy zipped files from CD onto the first hard disk. You could then unzip them onto the second hard disk.

5.2 Dump/restore distribution

We installed the Customized Offering 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.

For this installation we used a PC that had a SCSI-attached 3480/90 tape drive.

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:

- Four tapes containing 3390-3 volume D9ESY1
- Five tapes containing 3390-3 volume D9ECAT
- 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.[7] 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 96.

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.

[7] 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.

The general process we used is this:

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

We started by creating two empty 3390-3 volumes:

```
$ 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. The following steps compile the FLEX-ES definitions, start the FLEX-ES resource manager, and start S/390 emulation. These functions are further explained in the next chapter.

```
$ cfcomp defCOD                    (Compile the FLEX-ES definitions)
$ su                               (Switch to root)
  (password)
# resadm -s RCOD.rescf              (Start the resource manager)
  (pause for about 5 seconds)
# resadm -r                        (Verify that the resources started)
# exit                             (Leave root)
$ sh shCOD                         (Start a new FLEX-ES instance)
flexes>                            (Waiting for an IPL command)
```

The FLEX-ES definitions in file defCOD use the resource name RCOD and the system name SCOD. At this point, we mounted the stand-alone utility tape in our SCSI tape drive:

```
flexes> ipl 390                    (IPL from the tape)

(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)
(Alt-c is the default x3270 combination for clearing the screen)
(You should receive a number of title messages from DSF)

ICK005E DEFINE INPUT DEVICE, REPLY 'DDDD,CCUU' or 'CONSOLE'
(simply press Enter)[8]
ICK006E DEFINE OUTPUT DEVICE, REPLY 'DDDD,CCUU' or 'CONSOLE'
(simply press Enter)
ENTER INPUT/COMMAND:
init unit(320) nvfy devtyp(3390) volid(D9ESY1) nomap
(more messages)
U                                  (Reply U to initialize the volume)

ENTER INPUT/COMMAND:
init unit(321) nvfy devtyp(3390) volid(D9ECAT) nomap
(more messages)
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

[8] By Enter, we mean the 3270 Enter function. You may have customized your x3270 sessions to move this function to the right Ctrl key. By default, it is the large Enter key on the PC keyboard.

alter it after IPLing the first time, the tape should already be positioned so that the second utility is loaded.

```
flexes> ipl 390
```

This produces 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 is tape 1 of 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.

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 adapt themselves to any device addresses. However, the devices in your FLEX-ES definitions should match the addresses known in the z/OS IODF in order to IPL a system after it is restored.

5.3 The tar file distributions

The example we use here is an IBM internal package known as the Dallas DEMOpkg and is intended for IBM internal training. However, it is typical of the manner in which S/390 operating systems could be distributed in tar format. This package had one S/390 3390 volume per CD, in the form of a single compressed tar file. The file name (on the CD) was the same as the S/390 volser plus “tar” as a suffix. The first CD had a few additional files containing sample FLEX-ES definition and operational files.

The base file contained in a tar file is a complete image of a S/390 disk volume. Within it are emulated CKD tracks, cylinders, R0s, and so forth. There is a standard label, a VTOC, probably a VTOC index, and whatever data sets appear on that volume in a z/OS context. The data is in S/390 format. Text contained on a S/390 volume is usually EBCDIC and executables are S/390 binary files suitable for execution by z/OS. Due to the size of the emulated 3390 files, it is usually not practical to place several of these files in a single tar file.

External documentation listed five volumes that were needed to have a workable system. (Seven more volumes were included, containing optional products that we did not elect to install.) The base volsers were: DMTRES, DMTCAT, DMTOS1, DMTOS2, and DMTOS3. We installed these as follows:

- Insert first CD in the CD-ROM drive and wait for Linux to open a window displaying the contents of the CD. You could also use an **ls** command to display the contents:

```
$ls /mnt/cdrom                         (To display the contents of the CD)
   dmtres.tar     flexes.rsc     flexes.sys     ipl300      README.TXT
   sg24-6834-00   startFlex      startRes
```

- After inspecting the contents, we wanted two FLEX-ES definition files from the CD for later use. We copied them into the rundir directory:

```
$ cp /mnt/cdrom/flexes.rsc  /usr/flexes/rundir/DDrsc
$ cp /mnt/cdrom/flexes.sys  /usr/flexes/rundir/DDsys
```

- Untar the emulated volume into our /s390 directory:

```
$ cd /s390
$ tar -zxvf /mnt/cdrom/dmtres.tar
  (Close the CD-ROM automount window, if it is open)
$ umount /dev/cdrom
  (Remove the CD)
```

- Install the next CD and untar the contents, following the same pattern.

We then customized the FLEX-ES definitions that were on the first CD and created a shell script to help start this system. Our FLEX-ES definitions and the shell script are listed in “Definitions for the Dallas DEMOpkg” on page 92. The original definitions (from the CD) were in files DDrsc and DDsys; we merged these into a single file, DDdef. We determined the addresses (S/390 device numbers) to use by studying the sample definitions and the documentation provided with the CDs.

We compiled our definitions, started the FLEX-ES resource manager, started S/390 emulation (with the shell script), and IPLed our system:

```
$ cd /usr/flexes/rundir
$ cfcomp DDdef                              (Our FLEX-ES definitions)
# su                                        (Must be root to start resource manager)
  (password)
# resadm -s DDR.rescf                       (Start resource manager)
  (pause)                                   (Wait about 5 seconds)
# resadm -r                                 (To confirm that it started OK)
# exit                                      (Leave root)
$ sh shDD                                   (Start the shell script)
flexes> ipl 0300 0301DP.1                   (IPL the Dallas DEMOpkg system)
flexes>
```

We then logged onto TSO as SYSPRG1 (with the initial password the same as the userid). The functions described here (**cfcomp**, **resadm**, shell scripts, the *flexes>* prompt) are described in the next chapter. They are presented here for completeness. (Please note that our choice of DDR as a resource file name is not related to the DDR program often used with z/VM.)

5.4 OMA/2 distributions

The most likely use of an OMA/2 distribution is for z/VM. The following material describes the initial steps for this process.

We started with two CDs in a package labeled *Product Package 3390 DDR* for z/VM Version 4 Release 3.0. Examination of the first CD disclosed two directories: *tapes* and *9999*. The tapes directory contained a file named uaa925.tdf. This is a typical OMA/2 arrangement and a *tdf* file is always present on an OMA/2 distribution.

We first created two empty 3390-3 volumes in the /s390 directory:

```
$ ckdfmt -n -r /s390/VM1000 3390-3
$ ckdfmt -n -r /s390/VM2000 3390-3
```

The file names (VM1000, VM2000) are completely arbitrary. We then created the FLEX-ES definitions listed in “z/VM OMA/2 definitions” on page 97, along with the shell script shown. We compiled the definitions with the **cfcomp** command. (This was all done in the /usr/flexes/rundir directory.) The definitions provide two 3390 volumes at addresses 400 and 401, an emulated 3422 tape drive at address 500, and three 3270 sessions at addresses 600, 601, and 602. The 3422-type tape drive is typically used for OMA/2 emulation.

We installed the first CD in the CD-ROM drive and let Linux automount find it. (If this does not work, a **mount /dev/cdrom /mnt/cdrom** command can be used.) We used **resadm** to start the resource manager:

```
# mount /dev/cdrom /mnt/cdrom              (If needed)
# ls -al /mnt/cdrom                        (Inspect the CD)
# cd /usr/flexes/rundir                    (Our FLEX-ES definitions)
# resadm -s VMR.rescf
   (pause)
# resadm -r                                (Verify the resources started)
```

We then switched to userid *flexes* (that is, we exited from the *root* window):

```
$ sh shVM                                  (Start the shell script)
   (This should open a 3270 window)
flexes> mount 500 /mnt/cdrom:/mnt/cdrom/tapes/uaa925.tdf
flexes> ipl 500                            (IPL ICKDSF)
flexes>
```

The FLEX-ES definitions made the 3422 tape drive at address 500 offline. The **mount** command[9] makes the device ready and mounts the indicated OMA/2 TDF file. The format of the FLEX-ES OMA/2 **mount** command may appear odd. Use it exactly as shown here; there are no spaces before or after the colon in the command. We then IPLed from the OMA/2 tape drive.

Go to the 3270 window and press Enter. This should produce the message `Clear Screen When Ready`. With an x3270 emulator using an unmodified keymap, you can clear the screen with the Alt-c key combination, using the left Alt key. You can then proceed with the following sequence in the 3270 window. Remember to clear the screen whenever requested:

```
ICK005E DEFINE INPUT DEVICE, REPLY'DDDD,CUU' or 'CONSOLE'
Enter                                 (Press the Enter key)
ICK006E DEFINE OUTPUT DEVICE, REPLY 'DDDD,CUU' or 'CONSOLE'
Enter
-----DEVICE SUPPORT FACILITY 16.0X-----
ENTER INPUT/COMMAND:
cpvolume format unit(400) novfy volid(430RES) mode(ESA) nofiller
ICK003D REPLY U TO ALTER VOLUME 0400 CONTENTS, ELSE T
U
....takes about 10 minutes.  Clear screen as necessary for messages....
cpvolume format unit(401) novfy volid(430W01) mode(ESA) nofiller
U
```

9 Do not confuse the FLEX-ES **mount** command with the Linux **mount** command.

At this point the two 3390 volumes are initialized in a VM format. Now IPL the OMA/2 tape again to load the DDR restore program:

```
flexes> ipl 500
```

Wait about 10 seconds and press Enter on the 3270 screen. If the 3270 keyboard is locked, the keyboard Reset sequence for an unmodified x3270 is Alt-r.

```
Enter
z/VM DASD Dump/Restore Program
......
Enter:
sysprint cons
Enter:
input 500 tape (skip 1 rew
Enter:
output 400 dasd 430RES
Enter:
restore all
    (This takes about one minute)
HOLDING
Enter
Enter:
Enter
END OF JOB
```

We can now IPL the z/VM volume. We must include the address of the 3270 console as the IPL parameter:

```
flexes> ipl 400 600
flexes>
```

The IPL address is 400 and the IPL parameter is the address of the 3270 console. This should produce the Stand-Alone Loader screen for the Initial Installation System (IIS) for z/VM. We then followed the instructions in the *z/VM Installation Guide*, GC24-5992.[10] Remember to use the keyboard combinations Alt-c and Alt-r as needed. While following the instructions, we noted that the **instdir** command took at least a minute (and the system appeared hung during this time). We used the VM command **attach 500 * 181** at the appropriate place in the instructions, and assigned virtual tape address 181 for both tape volumes later in the installation process.

The installation process eventually issues the following messages:

```
INSTALL PROCESSING CONTINUES
(long pause)
PLEASE MOUNT VOLUME 2 ON TAPE DRIVE 181 THEN PRESS ENTER TO CONTINUE
```

At this point, do the following:

```
from flexescli     flexes> mount 500 OFFLINE
from root          # umount /mnt/cdrom
   (Remove old CD, insert second CD)
   (Wait for Linux automount or issue a mount /dev/cdrom /mnt/cdrom command)
from flexescli     flexes> mount 500 /mnt/cdrom:/mnt/cdrom/tapes/uaa926.tdf
                   flexes>
on VM 3270         Enter
```

Processing should continue with the second volume. Note that the CD file name was uaa926.tdf for the second volume. This name will be different for other releases or products, of course.

[10] Be sure to use the "instvm cd" EXEC when installing VM.

The z/VM installation process asks to process any RSU (service) tapes. We had a single CD labeled *4301RSU Stacked*. It had two TDF files in the tapes directory: uaa931.tdf and uaa932.tdf. This represents two service tapes. We followed the same procedure to process these tapes:

```
from flexescli     flexes> mount 500 OFFLINE
from root          # umount /mnt/cdrom
   (Remove old CD, insert second CD)
   (Wait for Linux automount or issue a mount /dev/cdrom /mnt/cdrom command)
from flexescli     flexes> mount 500 /mnt/cdrom:/mnt/cdrom/tapes/uaa931.tdf
                   flexes>
   (The attach 500 * 181 VM command was still in effect)
```

Following the installation instructions, we entered a **service** command in the VM session. When this finished (almost 20 minutes, many screens of output), it issued the message RSU tape 2 required on 0181. Since the second tape was on the same OMA/2 CD, we issued the command:

```
from flexescli     flexes> mount 500 /mnt/cdrom:/mnt/cdrom/tapes/uaa932.tdf
```

Mounting the new tape created a *device ready* interrupt and z/VM continued the service processing automatically. The final step in the z/VM basic installation used the command **put2prod**. This required several minutes and produced multiple screens of output.

5.5 File ownership (any distribution)

All of the files containing emulated S/390 volumes, however you received and installed them, *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. These examples assume you place the emulated volumes in /s390 or /s391. If you have a different arrangement, you need to display the appropriate directories.

6

Operational details

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.
- Install FLEX-ES.
- Install a S/390 operating system.

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

1. Create a FLEX-ES definition file.
2. Compile the definition file, creating rescf and syscf files.
3. Start the resource manager, using **resadm**, pointing to a rescf file.[1]
4. Start **flexes**, the S/390 emulator, pointing to a syscf file.
5. Start the **flexescli** program, which provides a command line interface for controlling a FLEX-ES instance.
6. Start at least one TN3270e session (for the MVS master console, or a similar purpose).
7. Issue an **ipl** command through the *flexescli* interface.
8. Use the S/390 operating system in the normal way.
9. Shut down the FLEX-ES system when finished.

These steps are described in this chapter. In general, most of the setup described here is 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.

Sample FLEX-ES definition files were referenced in the previous chapter and are listed in Appendix A, “Sample FLEX-ES operation files” on page 91. We do not discuss the syntax and construction of these definitions here. This is briefly covered in the Appendix and much of the construction is apparent by inspecting the definitions. The formal FSI documentation is

[1] **resadm** is a command used to manage the FLEX-ES resource manager. The Linux process name for the resource manager is resmanager. However, it is a common shortcut to refer to the resource manager as resadm.

very detailed and covers all the syntax and construction. The companion redbook, *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008, covers the basic syntax of common FLEX-ES definition statements.

There are many ways to set up and operate a FLEX-ES system. This chapter discusses the way we routinely do this, but there are many other valid approaches and styles. We assume you are using z/OS. If not, you can modify the following details to match your operating system.

6.1 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 (using **resadm**). The other window is used for running a shell script and is then used to enter *flexescli* commands.

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

- One (**su** to root) for **resadm** commands
- One for *flexescli* 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.

6.2 FLEX-ES system and resource definitions

Before a newly installed/restored z/OS system can be used, we must define the FLEX-ES system and resource elements required. “Minimal AD system definitions” on page 93 shows the file that we used for our initial AD CD-ROM system with four system volumes. We placed this definition file in /usr/flexes/rundir/defR14A. (File name defR14A is completely arbitrary.) We specify the name of the definition file as an argument for the FLEX-ES configuration compiler:

```
$ cd /usr/flexes/rundir
$ cfcomp defR14A
  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 defR14A
```

Our FLEX-ES definition file consists of two sections: the *system* section and the *resources* section. The compilation creates files S14A.syscf and R14A.rescf, based on the S14A and R14A names contained in our definitions.[2] 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       (If you are not already there)
# resadm -s R14A.rescf        (<--- To activate our resources. Must be root)
```

```
  (wait a few seconds)
# resadm -r                  (<--- To check that resources are started)
```

Again, we could issue the r**resadm** command directly because /usr/flexes/bin is in our PATH. You must be *root* to issue start or terminate resource manager functions with the **resadm** command. 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 (if any)
- Mode of operation (ESA, zArchitecture)
- 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

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 have S14A.syscf and R14A.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 R14A.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

[2] 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.

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 *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008.

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

6.2.1 Building a shell script

We used **vi** to create the following shell script file in /usr/flexes/rundir/sh14A. (The file name is arbitrary.) The contents are:

```
flexes S14A.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 &
flexescli localhost S14A
```

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

The next three commands[3] 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.[4] These session names match names in our resource definitions. There is no requirement to start these x3270 sessions here (in the shell script), but the terminal for the z/OS master console or consoles should start before IPLing z/OS. Using the shell script this way provides a convenient, automatic way to establish the z/OS console.[5] We normally use the second 3270 session as a TSO terminal.

The last command of this shell script, **flexescli localhost S14A**, starts the FLEX-ES Command Line Interface (CLI) program in interactive mode, with a *flexes>* prompt replacing the default Linux prompt. You can enter *flexescli* commands here,[6] 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 S14A.rescf** command), we can now invoke the shell script to start S/390 operation:

```
$ sh sh14A
flexes> ipl a80 0a82cs
flexes>
```

Here we started the `sh14A` 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 FLEX-ES CLI commands. We entered an **ipl** command to start z/OS. We could have included the **ipl** command in the sh14A 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.)

3 These are not required, but are suggested by FSI to help make smoother x3270 operation.
4 If we omitted the session identification, we would obtain the Terminal Solicitor selection menu on these sessions.
5 If you do not start the x3270 sessions this way, you would need to access the Terminal Solicitor and start the TN3270 session that will be used for the z/OS console before starting an IPL.
6 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.

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

```
flexes S14A.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 &
echo ‘ipl a80 0a82cs’ | flexescli localhost S14A
flexescli localhost S14A
```

This illustrates the two ways in which **flexescli** can be used. If a command is piped to it (with **echo**), then **flexescli** executes that command and quits. 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 (S14A). 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 (S14A) 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. See 7.6, “x3270 client” on page 71, 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 CD-ROM 1.4 system. You should read the AD documentation for additional parameters.

You can place CLI commands (typically **mount** commands) in a Linux file and supply the file name as the third parameter for the **flexescli** command. The commands in the file are executed before the *flexes*> prompt appears.

Terminal Solicitor

The Terminal Solicitor is a standard FLEX-ES component that operates as a Linux process. It is a TN3270 server that converts TN3270 client sessions to appear as local, non-SNA 3270 terminals. A remote user, for example, could connect his client TN3270 session to the Terminal Solicitor (using the Linux TCP/IP stack, not the z/OS TCP/IP function) and log onto TSO.

Starting the resource manager, using **resadm**, automatically starts the FLEX-ES Terminal Solicitor. In order to connect to the FLEX-ES Terminal Solicitor, we connect a TN3270e client to our 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[7]. The Terminal Solicitor presents a panel to us with the names of *available*[8] 3270 devices (these are often made available by the FLEX-ES CLI **mount** command, although we elected to hardcode them in the resource definitions).

[7] It can also be another window on our Linux desktop.
[8] 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.

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

Please select (X) the desired service and press enter
(PA1 to exit; CLEAR to refresh)

_ L702          _ L703
_ L704          _ L705          _ L706          _ L707
_ L708          _ L709          _ L70A          _ L70B
_ L70C          _ L70D          _ L70E          _ L70F
_ L710          _ L711          _ L712          _ L713
```

Figure 6-1 Terminal Solicitor panel (on a TN3270e screen)

The Terminal Solicitor display in Figure 6-1 might result from the definitions in "Full AD system definitions" on page 94. 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 is used for the MVS master console.[9] When we select a terminal (by marking it with an `X` and pressing Enter[10]), 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 is normally result in the USSTAB logo screen.

6.2.2 IPL z/OS

The z/OS IPL starts when you execute a *flexescli* **ipl** command. Once our z/OS was started, we watched NIP messages, responded as necessary, and watched normal z/OS startup on the x3270 window. Once you have z/OS IPLed, operation is the same as for any other z/OS installation.

6.2.3 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 6-1 on page 62. 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 the user eventually drops the TN3270e session, the S/390 3270 address is 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.

6.2.4 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.

[9] 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.

[10] Many 3270 emulator users set up the right 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 Ctrl key.

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. You can 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.

In response to a z/OS **d ios,config(all)** command, we received:

```
- 11.38.06          d ios,config(all)
  11.38.06          IOS506I 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
```

6.2.5 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) is not 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  ERB265I RMF: IOCDS INFORMATION UNAVAILABLE TO RMF.
   RESPONSE CODE 01F0
  11.22.37 STC00439  ERB260I ZZ : I/O QUEUING ACTIVITY RMF REPORT TERMINATED
  11.22.38 STC00439  ERB100I ZZ : ACTIVE
```

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

6.2.6 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 192.168.0.111 to z/OS TCP/IP, using the z/OS device addresses E20 and E21 for the interface. (Our Linux IP address was 192.168.0.110.)

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=192.168.0.111'
```

```
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 192.168.0.111 ETH1
...
BEGINRoutes
;     Destination    Subnet Mask   First Hop    Link Packet Size
ROUTE 192.168.0.0   255.255.255.0    =           ETH1 MTU 1500
ROUTE DEFAULT                      192.168.0.1   ETH1 MTU DEFAULTSIZE
ENDRoutes
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 eth*n*, or tr*n* numbers used in Linux. A more complete listing of our z/OS TCP/IP profile is in 7.8, “z/OS TCP/IP profile” on page 73.

With the specifications described here and using the z/OS 1.4s AD system, we have these connection options from another host in the network:

- **TN3270** to 192.168.0.110, port 24: This connects to the Terminal Solicitor (assuming the resource manager 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 192.168.0.111, 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 192.168.0.110, port 23: This connects to Red Hat Linux (if Telnet connections are enabled).
- **telnet** to 192.168.0.111, port 1023: This connects to z/OS UNIX System Services, as an ASCII terminal (assuming you are using a recent AD system).
- **telnet** to 192.168.0.111, port 23: This produces a line-mode connection (via VTAM) to TSO and is not very useful.

6.3 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 resources)
# resadm -k                       (Kill resource manager)
#
```

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, the resource manager is also stopped. You are not required to use the **resadm -k** command.

6.4 More comments

Some users (and FSI Business Partners) prefer to start the FLEX-ES resource manager automatically. This is done by placing a **resadm** command in the /etc/init.d/flexes file.[11] 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 resource file to be started. In our case, we had several different resource files that we used at different times. (These include R14A.rescf and R14B.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:

```
# resadm -T                (Stop all the current FLEX-ES resources)
# resadm -k                (Kill resource manager)
# resadm -s new.rescf      (Start a new resource file)
# resadm -r                (Verify that it looks OK)
----- Better method --------
# 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 the resource manager (with the -k option).

> **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 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.

[11] 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.

7

Basic discussions

Most of this redbook is concerned with the installation of a basic, entry-level EFS system. This chapter discusses additional topics that may be relevant for such systems. The companion IBM Redbook *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008, discusses more advanced topics.

7.1 Resource definitions

Typical FLEX-ES system and resource definitions are shown in the Appendix. We routinely place both *system* definitions and *resource* definitions in the same file. (This is not necessary and may not be desirable for more complex operations. However, we recommend it for getting started.)

Creating or changing FLEX-ES 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 defR14B                        (Or you can use another editor)
```

- Use the **cfcomp** command to compile the new definitions:

```
$ cfcomp defR14B
```

- Bring down z/OS, if it is running.
- Stop the emulated S/390:

```
flexes> shutdown
(You probably want to end any x3270 sessions on your screen)
```

- Stop the current FLEX-ES resources that are running:

```
# resadm -T                         (Only if resource manager is already running)
```

- Start the newly compiled resources:

```
# resadm -x R14B.rescf              (If resource manager is already running)
# resadm -s R14B.rescf              (If resource manager is not already running)
```

- Start your shell script (which starts FLEX-ES emulation) and IPL your system:

```
$ sh sh14B
flexes> ipl a80 0a8200
```

The **resadm -T** command stops any running FLEX-ES resources without stopping the resource manager itself. Note that you should edit and compile your FLEX-ES definitions while running as *flexes*, but you must be *root* to start, stop, or modify the resource manager using appropriate **resadm** operands.

You must recompile your definitions (with the **cfcomp** command) after you make any changes. You must then restart the resource manager with the newly compiled rescf file to make the changes active. Editing a definition file has no effect on FLEX-ES. Compiling a definition file has no effect until you restart the resource manager with the newly compiled definitions.

7.2 Linux windows and FLEX-ES operation

We found that we often have four windows 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 is useful for issuing **resadm** commands.[1] The window with the *flexes* prompt is generally required for FLEX-ES operation. We find that having two 3270 sessions is our most useful arrangement. Our typical startup process goes like this:

[1] This is not necessary, of course. Some users prefer to open a window, **su** to *root*, start the resource manager, exit from *root*, and run **flexescli** in that window.

```
Log into Linux with userid flexes        (Working directory is /usr/flexes)
Start two terminal windows on the Linux desktop
Window 1
   $ su                                  (Enter root password)
   # cd rundir                           (Change to /usr/flexes/rundir)
   # resadm -s R14A.rescf                (Start FLEX-ES resources)
   # resadm -r                           (Verify resources started)
Window 2
   $ cd rundir                           (Change to /usr/flexes/rundir)
   $ sh sh14A                            (Start our shell script)
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
   flexes> shutdown                      (Stop S/390 emulation)
Window 1
   # resadm -T                           (Stop FLEX-ES resources cleanly)
   # resadm -k                           (Terminate resource manager; optional)
   # exit
   $ exit
```

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

Another window (not *root*) is sometimes useful for running **vmstat** commands and similar utility functions. (This can 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.)

7.3 64-bit operation

FLEX-ES emulates an Architecture Level Set 3 (ALS 3) system if the *instset* parameter is set to z. This parameter is in the system definition portion of the FLEX-ES definitions. ALS 3 includes 64-bit operation. For example:

```
instset(esa)                             (Emulates a 31-bit system)
instset(z)                               (Emulates an ALS 3 (64-bit capable) system)
```

Remember that this refers only to the emulated S/390.[2] The base Intel system, running Red Hat Linux, is still in 32-bit mode. The largest S/390 memory that can be emulated is 2 GB. (We say that Linux is a 32-bit system because it is using 32-bit registers and integers for most functions. With the proper kernel options, Linux can use more than 2 GB of real PC memory. This has no direct bearing on the size of S/390 memory that can be emulated.)

7.4 Creating additional emulated volumes

You can create a new emulated DASD volume in two ways:

- You can *restore* it from a FLEX-ES (or AWSCKD) archived copy. This has the advantage that the internal structure of the volume (label, VTOC, catalogs, data sets) is already in place.
- You can *create* an uninitialized volume and then run jobs to initialize it.

[2] We should probably say zSeries here, but we use S/390 consistently throughout this book.

In either case you must have Linux disk space for the new volume. In our examples, we assume that directory /s390 has sufficient space. The Linux command **`df -h`** can be used to survey the available space in your file systems. In practice, there is no special need to have all the emulated volumes in the same Linux file system.

The general steps involved are:

1. Create or restore a 3390 volume as a Linux file.
2. Update the FLEX-ES definitions.
3. Compile the new FLEX-ES definitions.
4. Start FLEX-ES operation and IPL z/OS.
5. Run an ICKDSF job to initialize the volume, if needed.
6. Vary the volume online to z/OS, if needed.
7. Possibly update the VATLST00 member in PARMLIB.

You can create a new emulated volume with the following command:

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

This example creates an emulated 3390-1 volume, creating the Linux file name WORK02 in directory /s390. Many users select a file name (WORK02) that matches the volser of the emulated volume (after they initialize it to contain a volser), but there is no requirement to do this. The -r flag in the command indicates that the file should be created if it does not already exist. The **`ckdfmt`** command formats the file with 3390 cylinders and tracks but does not create any other internal structure. FLEX-ES need not be running for you to use the **`ckdfmt`** command.

You need to then add the new volume to your FLEX-ES resource definitions. Your FLEX-ES definitions must assign an address (device number) that corresponds to the correct device type for your S/390 operating system.[3] Remember that FLEX-ES definitions assign S/390 addresses (device numbers) through the *cu devad* parameter in the system definition. Also remember that the number of devices specified in the *cu* statement in the system section *must* match the number of *devices* defined for the matching control unit in the resource section. If you want to access the volume, you must compile (**`cfcomp`**) the new definitions and use **`resadm`** to restart the resource manager using the new rescf file.

It is possible to update an operational resource file while z/OS, for example, is running. However, this involves a number of techniques and is not recommended for a new user. For practical purposes (until you become a FLEX-ES expert), you must shut down your S/390 emulation to add new emulated devices.

If z/OS is IPLed with a newly created emulated drive, it detects an uninitialized volume and takes it offline. We need to run an ICKDSF job to create a label and VTOC on the volume. A typical z/OS job is:

```
//ICKDSF01  JOB  1,OGDEN,MSGCLASS=X
//  EXEC  PGM=ICKDSF,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  INIT UNIT(A90) NOVAL NVFY VOLID(WORK02) PURGE VTOC(0,1,29)
/*
```

A job similar to this can be found in recent AD systems in SYS1.P390.CNTL(DSFINIT). Do not include a *STORAGEGROUP* parameter in the INIT statement unless you are certain you want the volume controlled by DFSMS. Adjust the UNIT parameter to the address you assigned to the volume via FLEX-ES definitions. When you run this job, you need to reply `U`

[3] This depends on your S/390 operating system and its documentation. z/VM, for example, automatically senses devices and their addresses while z/OS has preassigned addresses in whatever IODF file is used for IPLing. For the AD system, the IODFs include 3390 devices in the A80-A9F range and in other ranges.

to a z/OS console message. When the job has finished, you can **VARY A90,ONLINE** on the z/OS console and begin using the volume.

We usually modify the VATLST00 member in the z/OS PARMLIB to control the mount attributes of new volumes, but this is not required.

7.5 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>
```

If you are flooded with messages, you can issue the following command to remove the pending output:

```
flexes> clear messages
```

Remember to press Enter (to return to the *flexes>* prompt) after any S/390 messages are displayed.

7.6 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 may use whatever TN3270e clients are available for their platforms. For this redbook project, we used the IBM PCOM package—and worked from OS/2 and Microsoft Windows machines.

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

```
$ x3270 -model 3 -keymap pc -port tn3270 localhost &
    |       |        |         |         |      +-- Run in background
    |       |        |         |         +--------- Local system (from /etc/hosts)
    |       |        |         +------------------- Port tn3270 (from /etc/services)
    |       |        +----------------------------- In /usr/X/lib/app-defaults/X3270
    |       +-------------------------------------- Models 2, 3, 4, 5 available
    +---------------------------------------------- Command
```

You can optionally identify the FLEX-ES terminal session you want to connect to like this:

```
$ x3270 -model 3 -keymap pc -port tn3270 localhost:L701 &
```

In this case, your resource definition (or a prior FLEX-ES **mount** command) must have defined a 3270 device named *L701*. 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** program must be in the current search path, of course. For FLEX-ES, it is in the /usr/flexes/bin directory, and we suggested placing this directory in the current PATH. The *model* parameter refers to 3270 models:

```
Model     Characteristics
  2       24 lines x 80 columns
  3       32 lines x 80 columns
  4       43 lines x 80 columns
  5       27 lines x 132 columns
```

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

Default keyboard mapping for x3270 (when started as shown) includes:

```
Function     Keys Used
PA1          Left-Alt + 1          (Hold down the left Alt key and press 1)
PA2          Left-Alt + 2
Clear        Left-Alt + c   or   Left-Alt + Home
Reset        Left-Alt + r
Enter        Enter                 (Large Enter key performs 3270 Enter function)
NewLine      Left-Alt + Enter      (Performs 3270 New Line function)
SysRq        Left-Alt + s          (The System Request key (usually for SNA))
F13          Left-Alt + F1         (And so forth for F14 - F24)
EraseEOF     Left-Alt + f
Insert       Left-Alt + i
Del          Delete
BackTab      Shift + Tab           (Tab backwards)
```

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 Cntrl key is positioned on a PC keyboard) performs the 3270 Enter function. You can change x3270 key assignments by doing the following:

```
# 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\
   Alt<Key>d:          Redraw()\n\
   Meta<Key>Return:    Newline()\n\
   Alt<Key>Return:     Newline()\n\
   <Key>Return:        Newline()\n\          (Change this line, as shown)
   !Shift<Key>Tab:     BackTab()\n\
   .....
   .....                                     (Near end of ‘x3270.keymap.pc’ section)
   Alt<Key>Home:       Clear()\n\
   <Key>KP_Enter:      Enter()\n\            (Insert this line)
   <Key>Control_R:     Enter()\n\            (Insert this line)
   <Key>Control_L:     Reset()\n\            (Insert this line)
   :<Key>:             Default()
x3270.keymap.pc84: \
.....
```

You need to use **:w!** to save the file in **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 Cntrl key to perform the 3270 Reset function, and the right Cntrl key to perform the 3270 Enter function. This is the most familiar keyboard arrangement for most 3270 users.

7.7 S/390 identification

You may need to supply the machine type and serial number of your emulated S/390. This is often needed for S/390 software licenses or registration on various Web sites. With z/OS, the easiest way to obtain this is with the **d m=cpu** console command:

```
d m=CPU
PROCESSOR STATUS
ID CPU     SERIAL
 0  +      00F0971245

CPC.SI = 1245.L02.FSI.FREM.000000000000F097
```

You can obtain the same basic information with a command from the *flexes* prompt:

```
flexes> display cpuid
CPUID=0a00f097 12450000
```

Both these displays indicate that the *machine type* is 1245 and the serial number is 00F097. (If a machine *type-model* number is needed, the model can be taken from the field containing the L02 in the **d m=cpu** display.) IBM has reserved machine types 1245, 1246, and 1247 for FLEX-ES emulated processors. Type 1247 is normally used for FLEX-ES systems obtained through the IBM S/390 Partners in Development program. The serial number is assigned through the FLEX-ES license code file and should be unique for each license.

The machine type and serial number are available through the S/390 instruction STIDP.

7.8 z/OS TCP/IP profile

This document is not intended to cover normal S/390 systems programming tasks. However, the subject of the profile for z/OS TCP/IP arises so often that we include a typical listing for a small EFS system. This is based on the profile for the AD CD-ROM 1.4s release, with all the comment lines removed. Your needs may be different; this is offered only as a working example.

```
ARPAGE 20
GLOBALCONFIG NOTCPIPSTATISTICS
IPCONFIG DATAGRAMFWD VARSUBNETTING SYSPLEXROUTING
SOMAXCONN 10
TCPCONFIG TCPSENDBFRSIZE 16K TCPRCVBUFRSIZE 16K SENDGARBAGE FALSE
TCPCONFIG RESTRICTLOWPORTS
UDPCONFIG RESTRICTLOWPORTS
DEVICE  LCS1  LCS  E20  AUTORESTART
LINK    ETH1  ETHERNET  0  LCS1
HOME
   192.168.0.121  ETH1
BEGINRoutes
;     Destination Subnet Mask   First Hop    Link Packet Size
ROUTE 192.168.0.0 255.255.255.0    =         ETH1 MTU 1500
```

```
ROUTE DEFAULT                  192.168.0.1  ETH1 MTU DEFAULTSIZE
ENDRoutes
AUTOLOG 5
  FTPD JOBNAME FTPD1       ; FTP Server
  PORTMAP JOBNAME PORTMAP1 ; USS Portmap Server (SUN 4.0)
ENDAUTOLOG
PORT
     7 UDP MISCSERV            ; Miscellaneous Server - echo
     7 TCP MISCSERV            ; Miscellaneous Server - echo
     9 UDP MISCSERV            ; Miscellaneous Server - discard
     9 TCP MISCSERV            ; Miscellaneous Server - discard
    19 UDP MISCSERV            ; Miscellaneous Server - chargen
    19 TCP MISCSERV            ; Miscellaneous Server - chargen
    20 TCP *  NOAUTOLOG        ; FTP Server
;   20 TCP *  NOAUTOLOG SAF FTPDATA ; FTP Server
    21 TCP FTPD1               ; FTP Server
;   21 TCP FTPD2 BIND FEC0:0:0:1:0009:0067:0115:0066 ; FTP IPv6
    23 TCP INTCLIEN            ; Telnet 3270 Server
;   23 TCP INETD1 BIND 9.67.113.3 ; z/OS UNIX Telnet server
    25 TCP SMTP                ; SMTP Server
    53 TCP NAMED               ; Domain Name Server
    53 UDP NAMED               ; Domain Name Server
   111 TCP PORTMAP             ; Portmap Server (SUN 3.9)
   111 UDP PORTMAP             ; Portmap Server (SUN 3.9)
;  111 TCP PORTMAP1            ; Unix Portmap Server (SUN 4.0)
;  111 UDP PORTMAP1            ; Unix Portmap Server (SUN 4.0)
   123 UDP SNTPD               ; Simple Network Time Protocol Se
   135 UDP LLBD                ; NCS Location Broker
   161 UDP OSNMPD              ; SNMP Agent
   162 UDP SNMPQE              ; SNMP Query Engine
   389 TCP LDAPSRV             ; LDAP Server
   443 TCP HTTPS               ; http protocol over TLS/SSL
   443 UDP HTTPS               ; http protocol over TLS/SSL
   512 TCP RXSERVE             ; Remote Execution Server
   514 TCP RXSERVE             ; Remote Execution Server
;  512 TCP * SAF OREXECD       ; z/OS UNIX Remote Execution Serv
;  514 TCP * SAF ORSHELLD      ; z/OS UNIX Remote Shell Server
   515 TCP LPSERVE             ; LPD Server
   520 UDP OROUTED             ; OROUTED Server
   580 UDP NCPROUT             ; NCPROUTE Server
   750 TCP MVSKERB             ; Kerberos
   750 UDP MVSKERB             ; Kerberos
   751 TCP ADM@SRV             ; Kerberos Admin Server
   751 UDP ADM@SRV             ; Kerberos Admin Server
  1933 TCP ILMTSRVR            ; IBM LM MT Agent
  1934 TCP ILMTSRVR            ; IBM LM Appl Agent
  3000 TCP CICSTCP             ; CICS Socket
  3389 TCP MSYSLDAP            ; LDAP Server for Msys
SACONFIG ENABLED COMMUNITY public AGENT 161
TelnetParms
  Port 23
  TELNETDEVICE 3278-3-E NSX32703 ; 32x80
  TELNETDEVICE 3279-3-E NSX32703 ; 32x80
  TELNETDEVICE 3278-4-E NSX32704 ; 48x80
  TELNETDEVICE 3279-4-E NSX32704 ; 48x80
  TELNETDEVICE 3278-5-E NSX32705 ; 132x27
  TELNETDEVICE 3279-5-E NSX32705 ; 132x27
  LUSESSIONPEND
  MSG07
  CodePage ISO8859-1 IBM-1047   ; Linemode ASCII, EBCDIC code pa
```

```
    Inactive 0                    ; Let connections stay around
    PrtInactive 0                 ; Let connections stay around
    TimeMark 600
    ScanInterval 120
  ; SMFinit std
  ; SMFterm std
    WLMClusterName
      TN3270E
    EndWLMClusterName
  EndTelnetParms
  BeginVTAM
    Port 23 ; 992
      DEFAULTLUS
           SC0TCP01..SC0TCP30
      ENDDEFAULTLUS
    LINEMODEAPPL TSO  ; Send all line-mode terminals to TSO
    ALLOWAPPL TSO* DISCONNECTABLE ; Allow all users access to TSO
    USSTCP USSN
    ALLOWAPPL *      ; Allow all applications
  EndVTAM
  START LCS1
```

We changed only a few lines in the profile distributed with the AD system and these are shown in **bold** above. Our corresponding resolver data (in TCPIP.TCPIP.DATA) is:

```
TCPIPJOBNAME TCPIP
HOSTNAME NF
DOMAINORIGIN  ITSO.IBM.COM
DATASETPREFIX TCPIP
NSINTERADDR  xxx.xxx.xxx.xxx     (find your own name server!)
RESOLVEVIA UDP
RESOLVERTIMEOUT 10
RESOLVERUDPRETRIES 1
ALWAYSWTO NO
```

7.9 Initial 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.

Starting msgmgr

One of the more common problems appears with messages similar to these:

```
FSIRA430 Aborting resadm -- unable to start resource manager; reason:
FSIRG840 Aborting -- unable to open "/dev/msgmgr/msg00000"  (errno = 19)
```

Errno 19 indicates that the msgmgr loadable module is not loaded. You can use the Linux command to list the Linux modules that are loaded:

```
$ /sbin/lsmod
```

The msgmgr module is found in /usr/flexes/modules. It is normally loaded by the S90FLEXES startup script that was added to Linux by the FLEX-ES installation process. If the **lsmod** listing does not include msgmgr, try this:

```
# cd /usr/flexes/modules
```

```
# ls                                          (verify that msgmgr.o exists)
# /sbin/insmod msgmgr.o                       (try to force loading)
```

Log file /var/log/messages should contain messages whenever msgmgr is loaded or unloaded; you can look for error messages there:

```
# grep msgmgr /var/log/messages | more        (look for error messages)
```

The msgmgr module is installed by the second rpm involved when installing FLEX-ES. Among other actions, this installation consists of a compilation (**gcc**) and a link (**ld**). These must complete successfully. If you did not observe this carefully during installation, you may be able to remove msgmgr (**rpm -e msgmgr**) and install it again. The most common reason for errors (with the **gcc** or **ld** steps) is that you did not include *Development Tools* and *Kernel Development* selections when you installed Linux. You need these to install the compiler and linker, and to install the kernel source. (If you selected the *Everything* option for Red Hat installation, these are included.)

If no errors occur during the **gcc** and **ld** operations but you have this error when attempting to use **resadm** to start the resource manager:

- The Linux kernel *source tree* in your Linux system may not be suitable for FLEX-ES. The Red Hat releases we recommend have an appropriate kernel.
- If you rebuild the kernel you should use a kernel source tree provided by FSI.
- Perhaps you downloaded Red Hat updates that included a new kernel. If so, this kernel no longer matches the source tree in your system. (You may be able to boot Linux with the original kernel, installing and running FLEX-ES from there.)

The right kernel

You *must* run msgmgr with the same kernel you used when you installed it. You may have built a new kernel. (This will become more common in the future because newer hardware may require that you rebuild your kernel.) If you build and use a new kernel, you must reinstall msgmgr to match that kernel.

Kernel or compiler mismatch

You may see messages about a mismatch between the compiler and modules. The messages usually appear when you start msgmgr, and might be found in /var/log/messages. The problem is that your kernel does not match the kernel source used to install msgmgr. The most common reasons are:

- You downloaded a kernel (probably as part of a package of updates). Your kernel no longer matches your kernel source and msgmgr does not work in this situation. You may be able to boot with your original kernel and bypass this problem.
- You built a new kernel, but you forgot to select it when you last booted. (After appropriate testing, you should consider making your new kernel the default kernel by changing the *default* value in /boot/grub/grub.conf.)

In practice, you can probably use your msgmgr module, regardless of these messages, provided you had a clean compile and link when you installed msgmgr. To try this, you can force the loading of msgmgr:

```
# cd /usr/flexes/modules
# insmod -f msgmgr.o
```

CU will not start

If the resource manager starts without errors but you receive error messages about control units when you start **flexes** (usually via a shell script), you should consider the following:

- The resource names in the FLEX-ES system definition must match the names in the resource definitions. Check *carefully* for typos. We frequently see, for example, CU3390 when you meant to type CU3990, or similar errors.
- *Carefully* check the Linux file names in the resource definitions. The files must exist *and be in the right format*. For example, if the DEVICE statement in the resource definition says 3390-3, it must point to a Linux file that emulates a 3390-3. If the file emulates a 3390-2, FLEX-ES does not start.
- Check the ownership of all the files used by the resource definitions. These must be owned by userid *flexes* and groupid *flexes*. You can check ownership with a command such as **`ls -al /s390`**. You can change ownership (working as *root*) with a command such as **`chown flexes:flexes /s390/WORK02`**.
- If you used our naming conventions, verify that directory /s390 is owned by *flexes*.

x3270 problems

The following error message means that you are starting the wrong copy of x3270:

```
Error: app-defaults version mismatch: want 3.2.18, got 3.1.1.0
```

See 4.1.3, “Location of x3270 program” on page 41, for more information.

Other errors

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

```
$ cd /usr/flexes/rundir
$ cfcomp defR14A                          (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 7.6, “x3270 client” on page 71, 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.

FLEX-ES trace files

FLEX-ES writes trace files that are intended only for use by FSI. However, these sometimes contain clues that may be useful. Consider the following:

```
$ cd /var/adm/flexes                        (All the trace files begin here)
$ ls -al                                    (Always use the -al switches)
  drwxrwxr-x    2   flexes  flexes         4096 June 12  12.02  CU3990
  drwxrwxr-x    2   flexes  flexes         4096 June 12  12.02  CU3991
  drwxrwxr-x    2   flexes  flexes         4096 June 12  12.02  S14B
    (and other files, including .flexeslicense)
$ cd S14B                                   (Move to one of the directories)
$ ls -al
  -rwxrwxr-x    2   flexes  flexes          514 June 12  12.02  devtr.1115.0
  -rwxrwxr-x    2   flexes  flexes           96 June 12  12.02  .stderr.0.1.1483
   (and other files, such as "." and "..")
$ dtprint devtr.1115.0                      (Display the trace)
$ cat .stderr.0.1.1483                      (If the length is not zero)
```

There are several important points in this example:

- The FLEX-ES trace files (or directories) are in /var/adm/flexes.
- /var/adm/flexes usually contains directories with the names of your system (S14B) or your resources (CU3990). The exact names depend on the names in your FLEX-ES definitions.
- Always use the **ls -al** command when looking in these directories. Many file names begin with a period and are not shown by a simple **ls** command.
- Many files (usually .stderr files) may have zero length. You can ignore these.
- You can **cat** any .stderr files with length greater than zero.
- You must use the **dtprint** command to view device trace (devtr) files.

If you attempted to start FLEX-ES many times, there may be many of these trace files and directories. You can delete them all (but be careful not to delete the .flexeslicense file). The time and date stamps can help you determine which are the most recent files.

Dongle problems

We found that the FSI dongle works only in the lower USB connector of a ThinkPad docking station.[4]

There have been rare cases of FSI dongle failures. These have fallen into two categories:

- A partial failure, which is associated with the message "`Cannot write to dongle`."
- Complete failure in which a Linux **cat /proc/bus/usb/devices** command does not display the presence of the FSI dongle. An additional check is to connect the dongle to a Microsoft Windows system. If Windows detects the dongle and offers to install a driver for it, this means the dongle is probably working. (There is no Windows driver for the FSI dongle, so you cannot go too far down this path.)

7.10 VTAM APAR

If you use XCA or OSA with SNA over Ethernet and you are using a multiprocessor system then you should investigate APAR OA03594. This problem and fix is not unique to FLEX-ES, but can occur on any system with more than one CPU and a fast XCA or OSA-1 function.

[4] We used a T23 ThinkPad. The situation may be different for other hardware combinations.

7.11 Security

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 main 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 consoles[5] to specific IP addresses and not through the Terminal Solicitor. This is done by specifying an IP address in the FLEX-ES resource 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.

7.12 RAS discussion

Is a ThinkPad EFS or xSeries EFS system as reliable as a “real” S/390? *No.* A reliability discussion can quickly become extended and open-ended and we do not attempt to explore all the potential discussion points. We can summarize our views this way:

- An xSeries 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 production EFS system is based on a high-quality server and can be expected to exhibit much better reliability than a typical desktop PC. Contributing factors are:
  - ECC memory (or other types of correcting 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
  - 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 is used only for S/390 emulation.
  - Thorough physical inspection of the hardware by the Business Partner involved

    While this sounds trivial, it contributes to stability.

[5] We use the term “master console” loosely here to mean any z/OS operator console.

  – 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 mobile computer, but this does not compete with an xSeries 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 an xSeries EFS system provides more than adequate reliability within its place in the hierarchy of servers. A ThinkPad EFS system is not normally intended as a production system and should not be compared with the RAS characteristics of production systems.

7.13 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 userid “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 (Spacebar to advance page)
                               (Ctrl-c 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 -tvf mytar.tar                       List files contained in "my.tar"
tar -xvf mytar.tar                       Extract all the files in the tar file
tar -zxvf mytar.tar.gz                   Extract from compressed 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?
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
umount /dev/cdrom               Unlock CD drive so you can remove CD
```

Multiple consoles, sessions, screens

Linux offers multiple consoles on the PC 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 GUI 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. Gnome offers *gedit*. This is much like a simple PC text editor and very unlike `vi`. It is suitable for almost all the text editing we required while installing and customizing FLEX-ES. The Red Hat Linux distribution includes a number of editors, and you can select the ones you like best.

8

FAQs

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. You may also consider the use of a USB hub to attach multiple USB devices, including the FSI dongle.

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 probably need a new license key file, but can use your existing dongle. A license key file may be used with the FLEX-ES release that corresponds to the license key level or with any earlier release.

Q: How do I display my FLEX-ES release number?
A: Here are three different ways:

```
# rpm -qa flexes               (from a Linux command prompt)
flexes> d release              (from a flexes> prompt)
# resadm -V                    (if the resource manager is running)
```

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 can 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 direct access storage device (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 queuing 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.

Q: I understand that a LAN MAC address is used to control FLEX-ES licenses in some cases. Can I use this method instead of a dongle?
A: FLEX-ES licenses for systems based on Linux normally require the use of 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 it is what we used, and the majority of EFS users have Ethernet LANs.

Q: How many Ethernet adapters should I have?
A: One should be sufficient. 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 do not see any difference between the two methods.

Q: Are 4mm tapes compatible between my MP3000 and my xSeries 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 has 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 main storage as large as possible (without triggering Linux swapping) or should I define some expanded storage at the expense of main storage?
A: Provided that you do not exceed the maximum of approximately 2 GB permitted for main storage, we think that server memory is better used as S/390 main 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 main storage, expanded storage, and a disk cache. If you want to define the largest possible S/390, you should read Chapter 2 in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008, for a discussion of various tradeoffs.

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 may indicate a practical maximum of something like three or four on a single panel.

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 may prefer to work through the Terminal Solicitor because it is operational even if z/OS TCP/IP is not working. You can use odd-sized 3270 sessions through z/OS TCP/IP and through the Terminal Solicitor (starting with FLEX-ES version 6.2.10).

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 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 FSI Business Partner from which you purchased the system. We suggest you should upgrade only when a later release contains a feature you need or when your FSI Business Partner recommends an upgrade. Remember that you need a new license key file for a new release.

Q: Where can I order a printed copy of the FLEX-ES manuals?
A: Consult your FSI Business Partner for this. FSI does not routinely produce printed manuals.

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 operation and 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 (at the same time), we suggest that 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 an xSeries EFS system, we believe a UPS is a good idea. (You may need to configure the BIOS options correctly for a ThinkPad in a docking station to continue operation (on the battery) if the AC power fails.)

Q: How do I order an EFS system? What if I am not a PWD member?
A: You do not need to be a PWD member to order an EFS system (although you do not have access to AD CD-ROM systems or some of the other S/390 operating system packages mentioned in this document). The FSI Web site has a contact point for locating 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: 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 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: PWD members should ask their Business Partner if a new license key file is needed. You may 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. For example, your basic FLEX-ES license may contain a *speed limit* based on your S/390 software license.

Q: What is the difference between swapping and paging?
A: None, in these discussions.

Q: Various people mention different MIPS ranges for EFS systems. What is correct?
A: 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 60+ 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 rebuild the Linux kernel, you need to reinstall the FLEX-ES msgmgr package and you need to make a special effort to make the USB dongle interface work. You *must* use a kernel that matches the kernel source tree present when msgmgr was installed.

Q: Can I use DHCP to obtain a LAN IP address?
A: You can do this for Linux TCP/IP, but not for z/OS TCP/IP. Your EFS system is essentially a S/390 server, not a client, and you may 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 TCP/IP. If you are sharing an Ethernet adapter with Linux, you also need to change the IP address in the FLEX-ES resource stanza that defines the emulated 3172. To change the Linux IP address, use:

```
(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: See the appropriate chapter in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008, for more information about printing.

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. Newer ThinkPads use less power and have better batteries, so the duration may be longer.

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 may 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 xSeries has four processors, with one licensed for S/390 emulation. Can I also license more processors for S/390 emulation? Will this make a faster system? How much faster?
A: Yes, you can license more processors for S/390 emulation. You cannot *dedicate* all processors for this purpose, because this would leave no processor for other required Linux functions and auxiliary FLEX-ES processes.

Remember that enabling more processors 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 has fewer context switches and avoids 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 or processors?
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 Linux processes and are dispatched on the "other" Pentium processors.

Q: I have an older S/390 and am interested in changing to an xSeries EFS machine. I currently use seven channels on my machine. Do I need this many channels on the xSeries 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 xSeries EFS configurations.

Important: Remember that no channels, parallel or ESCON, are available for Linux-based FLEX-ES systems at the time of writing. The comments here are based on experience with UNIX-based FLEX-ES systems and are likely to apply to Linux-based systems if they become available.

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 xSeries 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 cables 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 server 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 or you could construct your own. In any event, this number of cables cannot simply hang from the back of the server. You should talk with your Business Partner about your plans.

Q: How much electrical power does your xSeries EFS system use? How much does it cost to run all the time?
A: This varies with system configurations, of course. We informally measured the current used by our Netfinity 5100 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 to 240 watts. At typical U.S. electricity prices, it costs 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) substantially increases the power required.

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 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 /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: Can I upload from diskettes using the IND$FILE 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 on a diskette, you can use:

```
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 contents 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
```

Then scroll to the third record. This is 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 terminates **ckddump** output. If you want to see only the volser, use:

```
$ ckddump /my/linux/files/AAA 0 0 | grep VOL1
```

Q: You said that Parallel Sysplex is not supported. Can I run basic sysplex, using shared DASD for coupling data sets?
A: We have had reports of a user or users running and testing base sysplex on FLEX-ES 6.1.15 (or later) using emulated CTC connections between the emulated S/390 instances, but it can be very complex and is not supported for production use.

Q: Can I run z/OS under z/VM? Can I emulate sysplex functions under z/VM?
A: Yes and yes (but only for basic susplex).

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 5.1.2, “Installation tasks” on page 48). New AD systems are not installed often enough to worry much about the installation speed, so we usually overwrite old volumes (for example, rename Z4RES1 to S4RES1 and overwrite it) and ignore any performance issues.

Q: Looking through the Red Hat list of services, I see something named FLEXES09 that is enabled. The comments are about token ring MAC addresses. I do not have token ring. Do I need this?
A: This is part of FLEX-ES support for SNA on token ring. It is a small component that must run (if needed) during system startup. Leave it as it is.

Appendix A. Sample FLEX-ES operation files

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.

Definitions for the Dallas DEMOpkg

The following FLEX-ES definition file was used to load and IPL the first five volumes of the Dallas z/OS 1.4 DEMOpkg. This is a subset of the FLEX-ES definition samples included on the CDs. Also, the CDs contained separate files for the system definition and the resource definitions. We combined these into a single definition file to be more consistent with our other examples.

```
system DDS:
memsize(524288)
cachesize(4096)
instset(z)
tracesize(512)
cpu(0)
channel(0) local
channel(1) local
channel(2) local
channel(3) local
channel(4) local
cu devad(0x300,5) path(2) resource(RSC39902)
cu devad(0x460,4) path(3) resource(RSC31741)
cu devad(0x470,4) path(3) resource(RSC31742)
cu devad(0x5DE,2) path(4) resource(RSC3088)
end DDS

resources DDR:
RSC31741: cu 3174
   interface local(1)
   device(00) 3278 Tso1
   device(01) 3278 Tso2
   device(02) 3278 Tso3
   device(03) 3278 MstrCon
end RSC31741

RSC31742: cu 3174
   interface local(1)
   device(00) 3278 Tso4
   device(01) 3278 Tso5
   device(02) 3278 Tso6
   device(03) 3278 Tso7
end RSC31742

RSC39902: cu 3990
   interface local(1)
   device(00) 3390-3 /s390/dmtres
   device(01) 3390-3 /s390/dmtcat
   device(02) 3390-3 /s390/dmtos1
   device(03) 3390-3 /s390/dmtos2
   device(04) 3390-3 /s390/dmtos3
end RSC39902

RSC3088: cu 3172
   options 'ipaddress=192.168.0.111'
   interface local(1)
   device(00) 3172 eth0
   device(01) 3172 OFFLINE
end RSC3088

end DDR
```

The following shell script was used to start operation. It opens three 3270 sessions on the Linux desktop:

```
flexes DDS.syscf
x3270 -model 4 -port tn3270 localhost:MstrCon &
x3270 -model 4 -port tn3270 localhost:Tso4 &
x3270 -model 4 -port tn3270 localhost:Tso5 &
flexescli localhost DDS
```

Minimal AD system definitions

Our system and resource definitions (both in the same file) used for the four volumes required to IPL the z/OS Release 1.4s AD CD-ROM system were as follows:

```
system S14A:
memsize(524288)
cachesize(4096)
instset(z)
cpu(0)
channel(0) local
channel(1) local
channel(2) local
cu devad(0xA80,8) path(2) resource(CU3990)
cu devad(0x700,5) path(0) resource(CU3174)
cu devad(0xE20,2) path(1) resource(CU3172)
end S14A

resources R14A:
CU3990: cu 3990
interface local(1)
device(00) 3390-3 /s390/S4RES1
device(01) 3390-3 /s390/S4RES2
device(02) 3390-3 /s390/OS39M1
device(03) 3390-3 OFFLINE
device(04) 3390-3 OFFLINE
device(05) 3390-3 OFFLINE
device(06) 3390-3 OFFLINE
device(07) 3390-3 /s390/S4USS1
end CU3990

CU3174: cu 3174
interface local(1)
device(00) 3278 mstcon
device(01) 3278 L701
device(02) 3278 L702
device(03) 3278 L703
device(04) 3278 L704
end CU3174

CU3172: cu 3172
interface local(1)
options 'ipaddress=192.168.0.111,adapternumber=0'
device(00) 3172 eth0
device(01) 3172 OFFLINE
end CU3172

end R14A
```

The file names shown, such as S4RES1, are typical for the AD 1.4s release. The previous AD 1.4 release typically used names such as Z4RES1.

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 L701) 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). 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.)

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 3172 definition is not required for a minimal system. You could omit it and the corresponding cu definition. However, many users want to establish their z/OS TCP/IP connectivity, even when working with a minimal AD system and the 3172 is needed for this.

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.

After using **cfcomp** to compile the above definitions, we used this shell script to start operation. (A **resadm -s R14A.rescf** command must be issued first.)

```
flexes S14A.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 &
flexescli localhost S14A
```

Full AD system definitions

The following definitions assume that all the AD CD-ROM volumes have been loaded. The definitions also assume that three scratch volumes have been created (WORK01, WORK02, WORK03) and that a tape drive may be used. (The tape drive could be a real SCSI drive or a FakeTape drive.) More 3270 terminals are defined for use by remote users via the Terminal Solicitor. Note that we used system name S14B and resource name R14B so we could have the minimal and full definitions in the same directory.

The file names shown, such as S4RES1, are typical for the AD 1.4s release. The previous AD 1.4 release typically used names such as Z4RES1. In these definitions we assume that all the S/390 emulated volumes fit in one Linux file system (/s390). If this is not the case, you need to modify the file names as needed.

```
system S14B:
memsize(524288)
cachesize(4096)
instset(z)
cpu(0)
```

```
channel(0) local
channel(1) local
channel(2) local
cu devad(0xA80,14) path(2) resource(CU3990)
cu devad(0xA90,3) 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 S14B

resources R14B:
CU3990: cu 3990
interface local(1)
device(00) 3390-3 /s390/S4RES1
device(01) 3390-3 /s390/S4RES2
device(02) 3390-3 /s390/OS39M1
device(03) 3390-3 /s390/S4DB21
device(04) 3390-2 /s390/S4CIC1
device(05) 3390-3 /s390/S4DIS1
device(06) 3390-3 /s390/S4DIS2
device(07) 3390-3 /s390/S4USS1
device(08) 3390-3 /s390/S4DIS3
device(09) 3390-2 /s390/S4IMS1
device(10) 3390-3 /s390/S4WAS1
device(11) 3390-3 /s390/S4WAS2
device(12) 3390-3 /s390/SARES1
device(13) 3390-3 /s390/S4DIS4
end CU3990

CU3991: cu 3990
interface local(1)
device(00) 3390-1 /s390/WORK01
device(01) 3390-1 /s390/WORK02
device(02) 3390-1 /s390/WORK03
end CU3991

CU3174: cu 3174
interface local(1)
device(00) 3278 mstcon
device(01) 3278 L701
device(02) 3278 L702
device(03) 3278 L703
device(04) 3278 L704
device(05) 3278 L705
device(06) 3278 L706
device(07) 3278 L707
device(08) 3278 L708
device(09) 3278 L709
device(10) 3278 L70A
device(11) 3278 L70B
device(12) 3278 L70C
device(13) 3278 L70D
device(14) 3278 L70E
device(15) 3278 L70F
device(16) 3278 L710
device(17) 3278 L711
device(18) 3278 L712
device(19) 3278 L713
end CU3174
```

```
CU3172: cu 3172
interface local(1)
options 'ipaddress=192.168.0.111'
device(00) 3172 eth0
device(01) 3172 OFFLINE
end CU3172

CU3480: cu 3480
interface local(1)
device(00) 3480 OFFLINE
end CU3480

end R14B
```

There are many arbitrary choices in these definitions. You need to examine the definitions and customize them to your purposes.

This shell script is just like the first one, except that it uses system name S14B instead of S14A.

```
flexes S14B.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 &
flexescli localhost S14B
```

Customized Offerings Driver definitions

We used the following FLEX-ES definitions and shell script with the Customized Offerings Driver:

```
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(0x0A1,1) 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              #address 320
device(01) 3390-3 /s390/TEMP02              #address 321
end CU3990

CU3174: cu 3174
interface local(1)
device(00) 3278 LF00                        #address F00
device(01) 3278 LF01
device(02) 3278 LF02
```

```
end CU3174

CU3175: cu 3174
interface local(1)
device(00) 3178 mstcon                      #address 0A1
end CU3175

CU3490: cu 3490
interface local(1)
device(00) 3490-E /dev/sg1                  #address 390
end CU3490

end RCOD
```

The following is the shell script we used to start the system:

```
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
```

z/VM OMA/2 definitions

The following provide a minimum system definition for the installation of z/VM from OMA/2 CDs:

```
system VMS:
memsize(524288)
cachesize(4096)
instset(z)
tracesize(512)
cpu(0)
channel(1) local
channel(2) local
channel(3) local
cu devad(0x400,2) path(1) resource(CU3990)
cu devad(0x600,3) path(2) resource(CU3174)
cu devad(0x500,1) path(3) resource(CU3422)
end VMS

resources VMR:
CU3174: cu 3174
interface local(1)
device(00) 3278 L600                        #address 600
device(01) 3278 L601
device(02) 3278 L602
end CU3174

CU3990: cu 3990
interface local(1)
device(00) 3390-3 /s390/VM1000              #address 400
device(01) 3390-3 /s390/VM2000
end CU3990
```

```
CU3422: cu 3422
interface local(1)
device(00) 3422 OFFLINE                   #address 500
end CU3422

end VMR
```

We used the following shell script with these definitions:

```
flexes VMS.syscf
x3270 -model 3 -keyboard pc -port tn3270 localhost:L600 &
x3270 -model 3 -keyboard pc -port tn3270 localhost:L601 &
flexescli localhost VMS
```

Definition statements

A *system* keyword is required in the system definition section. The name following the keyword is the name of the syscf created when the definition is compiled. A colon must follow the name.

memsize expresses kilobytes of storage and the largest permitted value is 2097152.

essize is for expanded storage and is expressed in megabytes. Furthermore, this parameter must be a multiple of 16.

instset is either *esa* or *z* for the systems we discuss in this document. An esa parameter causes 31-bit S/390 emulation and a z parameter enables 64-bit capable zSeries emulation.

cachesize is expressed as kilobytes of S/390 storage. We suggest using 4096 for this parameter. This cache is used internally by FLEX-ES.

tracesize is 512 unless you are directed to a different size by FSI or your Business Partner.

cpu and *channel* numbers are arbitrary, but usually start with zero. You cannot have more CPUs defined than you have licensed FLEX-ES processors.

The *dedicated* keyword on a `cpu` statement causes FLEX-ES to reserve an Intel processor solely for S/390 emulation. This improves performance. You must have at least one Intel processor that is not dedicated.

Additional information about FLEX-ES definitions can be found in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008.

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 about ordering these publications, see “How to get IBM Redbooks” on page 100. Note that some of the documents referenced here may be available in softcopy only.

- *S/390 Partners in Development: OS/390 (and z/OS) New User’s Cookbook*, SG24-6204
- *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008

Other publications

These publication *z/VM Installation Guide*, GC24-5992, is also relevant as a further information source. You may also want to refer to the following publications from Fundamental Software, Inc.:

- *FLEX-ES Technical FAQ,* FSIMM040
- *FLEX-ES Planning Guide,* FSIMM100
- *FLEX-ES Operator’s Guide*, FSIMM200
- *FLEX-ES CLI Language Reference*, FSIMM210
- *FLEX-ES System Programmer’s Guide*, FSIMM300
- *FLEX-ES Resource Language Reference*, FSIMM310

Online resources

These Web sites and URLs are also relevant as further information sources:

- Fundamental Software, Inc.:

  `http://www.funsoft.com`
- Sistina home page

  `http://www.sistina.com`

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

U

V

X

Z

IBM Redbooks

EFS Systems on a Linux Base: Getting Started

(0.2”spine)
0.17”<->0.473”
90<->249 pages

EFS Systems on a Linux Base: Getting Started

Planning for ThinkPad and Netfinity bases

Installation, customization, operation

Common techniques

This IBM Redbook describes the basic installation of FLEX-ES (a product of Fundamental Software, Inc., Fremont, California) with several z/OS operating system packages. Much of the material is in cookbook format. More advanced techniques and options are described in *S/390 PartnerWorld for Developers, ITSO/EFS Project EFS Systems on a Linux Base: Additional Topics*, SG24-7008. These books are intended primarily for members of the IBM PartnerWorld for Development organization and for internal IBM users of the ITSO/EFS package.

This second edition includes the latest updates to the first edition of this book. It also contains more information about using Red Hat 9.0 with FLEX-ES Release 6.2.14.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7007-01 ISBN 0738499404