

EFS Systems on a Linux Base: Channel Adapters

Functional description

FLEX-ES definitions

Control Unit emulation



Bill Ogden

Redbooks



International Technical Support Organization

EFS Systems on a Linux Base: Channel Adapters

Archived

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

First Edition (March 2005)

This edition applies to Version 7.0.2 of FLEX-ES, a product of Fundamental Software, Inc., of Fremont, California.

© Copyright International Business Machines Corporation 2005. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
Preface	vii
Authors	vii
Become a published author	vii
Comments welcome	viii
Chapter 1. Introduction	1
1.1 ESCON quick review	3
Chapter 2. FSI channel adapters	5
2.1 Serial channel adapters	6
2.2 Parallel channel adapters	7
2.3 The rpm installation	11
2.4 The eprog program	12
2.4.1 Firmware upgrades	12
2.5 Linux messages	13
Chapter 3. Configuration as a channel	15
3.1 Configurations	18
3.2 Working example	20
3.3 Logical control unit paths	23
Chapter 4. Configuration as a CU	25
4.1 Control unit path numbers	28
4.2 Multiple control units	29
4.2.1 Shared DASD with emulation	31
Chapter 5. FLEX-ES releases	33
5.1 Release 6.2.18	33
5.2 Release 7.0.2	33
Appendix A. FAQ	35
Related publications	39
IBM Redbooks	39
Other publications	39
How to get IBM Redbooks	39
Help from IBM	39
Index	41

Archived

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®
@server®
Redbooks (logo) ™
z/Architecture™
z/OS®
z/VM®

zSeries®
CUA®
DB2®
ESCON®
IBM®
PartnerWorld®

ibm.com®Redbooks™
Redbooks (logo)™
S/370™
S/390®
ThinkPad®

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

Preface

This IBM® Redbook briefly describes channel adapters and functions available for use with FLEX-ES. These channel adapters connect FLEX-ES systems through parallel channels and serial (ESCON®) channels to typical S/390® I/O devices, including disk drives. New FLEX-ES support, starting with release 7.0, provides more general channel support than was available in earlier releases and includes usage of new serial channel adapters.

This document is primarily directed to participants in the IBM internal ITSO/EFS project and to IBM PartnerWorld® for Developers members with relevant interests. It is not intended as an introduction to FLEX-ES and readers are assumed to be generally familiar with FLEX-ES definitions and operations as well as general S/390 terminology.

Authors

Bill Ogden is a retired IBM Senior Technical Staff Member, who still works on his favorite projects with the International Technical Support Organization, Poughkeepsie Center. He specializes in entry-level S/390 and z/OS® systems for new users, developers, and educational usage. Bill has been with the ITSO since 1978.

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

Carl Ross, who helped with several aspects of practical ESCON implementation.

Scott Carter, who, in addition to being an eagle-eyed proofreader, provided much help with understanding control unit functions.

Gary Ehemann, who helped in many areas and especially with details about multiple paths within the channel protocol.

David MacMillan, who helped with an excellent set of initial definition examples and operational descriptions.

In addition, we thank **Ken Trowell**, IBM Poughkeepsie, for helping us obtain a better understanding of control unit logical paths.

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 will team with IBM technical professionals, Business Partners, and clients.

Your efforts help increase product acceptance and client satisfaction. As a bonus, you will 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 e-mail 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

Introduction

This IBM Redbook briefly describes channel adapters and functions available for use with FLEX-ES. This is not an introduction to FLEX-ES, and we assume that you are generally familiar with FLEX-ES definitions and operations. You can find background material in the following IBM Redbooks:

- ▶ *EFS Systems on a Linux® Base: Getting Started*, SG24-7007-01
- ▶ *EFS Systems on a Linux Base: Additional Topics*, SG24-7008
- ▶ *z/VM® on an EFS Base: Getting Started*, SG24-7009

FLEX-ES and the channel adapters described in this redbook are products of Fundamental Software, Incorporated, of Fremont, California.¹ The company is commonly known as FSI. Their products are available through their business partners. The formal documentation for the channel adapters described in this redbook is in FSI documentation, primarily in the most recent versions of the following two documents:

- ▶ *System Programmer's Guide*, FSIMM300
- ▶ *Resource Language Reference*, FSIMM310

The following three categories of FSI channel adapters are available for use with FLEX-ES:

- ▶ **Serial Channel Adapter (SCA)**

There are two versions of this adapter, one with a single ESCON channel and one with four ESCON channels. Only the single-channel version was available at the time of writing. (ESCON is an IBM term. A more generic name, *Serial Channel*, is used with these adapters. However, these adapters are for use only with ESCON devices.)

- ▶ **Parallel Channel Adapter (PCA)**

Two versions of this adapter are available, one with a single channel and one with three channels.

- ▶ **ICA**

This adapter provides six telecommunication interfaces and functions like the Integrated Communications Adapter (ICA) available in earlier S/390 systems.

¹ You can find more information about Fundamental Software, Inc., (FSI) at <http://www.funsoft.com>.

This redbook concentrates on z/OS usage. ICA interfaces are not supported by z/OS and are not further discussed here. All three channel adapters are available for both the UnixWare and Linux versions of FLEX-ES. This document describes channel use with the Linux version of FLEX-ES, although the general discussion applies to both versions. The serial channels and parallel channels are collectively referenced as *physical channels* to clearly distinguish them from the emulated FLEX-ES channels.²

The parallel channel adapter has been available for several years. Until recently, it was used only with the UnixWare version of FLEX-ES and required somewhat restrictive memory management definitions. Both the serial (ESCON) and parallel channel adapters are now supported by both versions of FLEX-ES (UnixWare and Linux) and do not require special memory definitions within FLEX-ES. This support is available starting with FLEX-ES release 7.0.³

FLEX-ES normally emulates a S/390 (or a zSeries® machine or a S/370™, and so forth). The FSI channel adapters connect standard mainframe I/O control units to the emulated S/390. FLEX-ES can also emulate control units and permit mainframes to use these control units via the channel adapters. Figure 1-1 illustrates the two different uses of FLEX-ES.

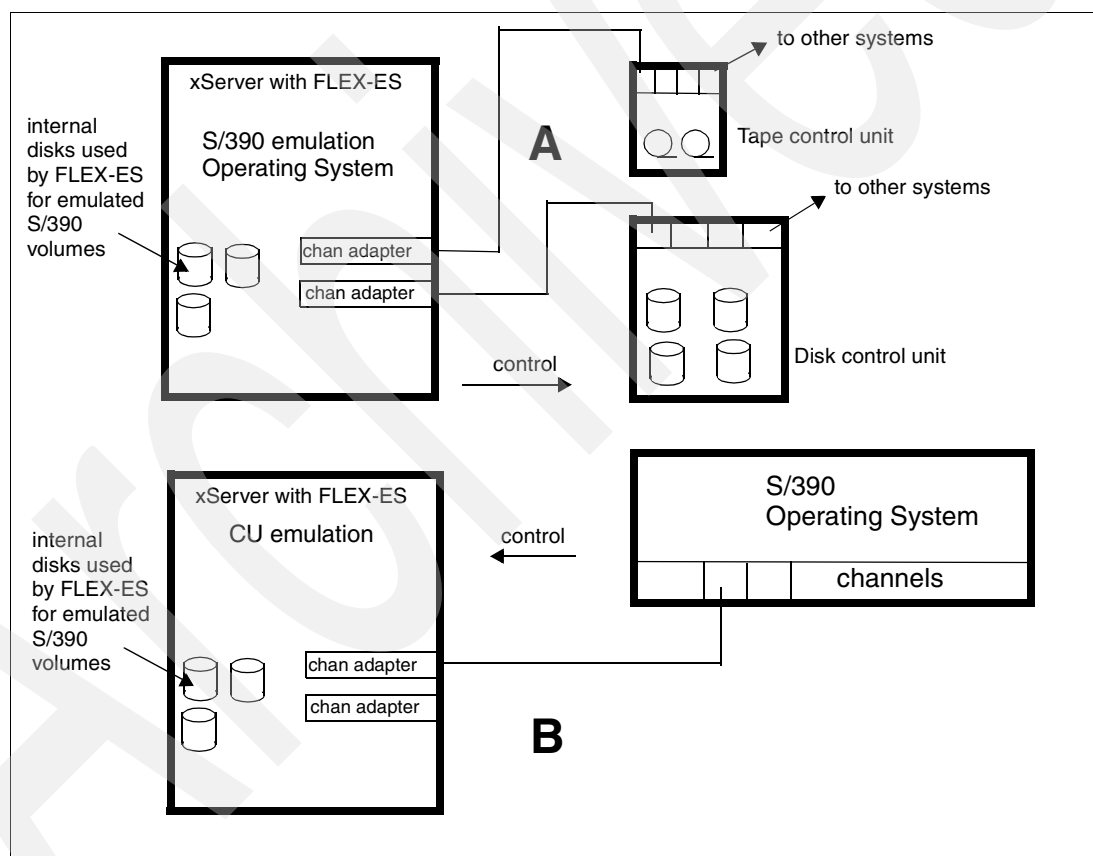


Figure 1-1 S/390 emulation versus host-connected control unit emulation

Configuration A represents typical FLEX-ES operation, including two channel adapters that access conventional S/390 control units and devices. In practice, these connections might be through ESCON Directors, but this is omitted in this illustration. Configuration B represents an alternative use of FLEX-ES as a S/390 disk control unit and disk volumes. The emulated disk volumes are typically 3390 volumes. In one case FLEX-ES is a processor and in the other

² The FLEX-ES emulated channels are types *local* and *network* (with subtypes of each).

³ This new level of FLEX-ES support also applies to the ICA adapter.

case it is a control unit and devices. The channels in these configurations are either parallel or serial channels.⁴

Both functions shown in the figure can be done at the same time. A FLEX-ES system running z/OS, for example, can access standard mainframe I/O devices through the physical channel adapters while also providing emulated devices for the mainframe to use. However, you cannot use a single FSI physical channel for both purposes in a given FLEX-ES environment.⁵ You can share emulated disk volumes that FLEX-ES emulated mainframes use with traditional mainframes through use of a channel adapter.

The FLEX-ES license must be authorized (via the license key file) in order to use the channel adapters.

1.1 ESCON quick review

We assume that most readers are interested in ESCON channels. (We recognize that many people still use parallel channels, but we assume that FLEX-ES users with parallel channels are already familiar with the earlier FLEX-ES support for the FSI parallel channel adapters.) The following brief overview of ESCON terminology and characteristics may be helpful.

An ESCON channel provides a *link* to a *control unit*. The link is a *fiber optic cable* with two optical fibers (one for each direction). The two unidirectional paths are not exploited for full duplex operation. *Frames* are sent over the link. There are link-level frames (to manage the link) and device-level frames (primarily for data). The largest data frame is 1024 bytes plus an 11-byte header. The channel and control units automatically block and unblock data into these frames.

Most ESCON fiber optic cables have an ESCON connector (*SC connector*) on each end of the cable. Newer systems, such as IBM zSeries machines, use an *MTRJ connector* for ESCON channels. The MTRJ connector is much smaller than the SC connector and an *adapter cable* (SC on one end and MTRJ on the other end) is used to interface with older ESCON cables. The FSI adapter cards use MTRJ connectors and an appropriate adapter cable is provided with each FSI adapter.

ESCON is physically a *point-to-point* link mechanism. Physically separate control units cannot be daisy chained together in the manner that parallel channel control units are chained. However, a physical control unit may have multiple logical control units internally, and a single ESCON channel can access these multiple control units. Multiple logical control units in one physical control unit are common for 3990-like devices.

Control units have addresses in the range 0 - 15 (decimal), usually expressed as a single hexadecimal digit. The devices connected to the control unit have addresses in the range 0 - 255, usually expressed as two hexadecimal digits. These two types of addresses are typically written as the *CUADD* and *UNITADD*. By convention, the UNITADD is the first unit address that the control unit uses and is assumed to increment sequentially for all additional devices that the control unit operates.

An ESCON link can be connected to an *ESCON Director*. Other *ports* on the Director may be connected to control units or other channels. These are all point-to-point connections. The Director may logically switch the link connection to any one of the ports on the Director, but the physical ESCON connections are always point-to-point. Each port on an ESCON Director

⁴ This is a change from earlier parallel channel implementation in which DASD connections were not supported.

⁵ The FLEX-ES definitions set the environment. To change this, stop FLEX-ES operations and restart with different definitions.

has a *Link Address* (often known as a *port address*). This is a one-byte address with 256 possible values.

ESCON runs at *200 megabits* per second. Data bytes are automatically expanded to 10 bits/byte to provide additional control and error recovery information. The bit rate of an ESCON channel is *20 megabytes* per second.⁶ The actual effective speed is less. The ESCON protocol includes idle packets between data frames. In practice, the operation of I/O devices and the host performance in starting the required channel programs tend to be the limiting factors for performance.

You can configure ESCON channels as several *channel types*, including the following:

- ▶ CNC, which provides normal operation of the ESCON channel.
- ▶ CTC, which provides a control unit function allowing two channels to be connected together.
- ▶ CVC, which is used with an ESCON to parallel converter.
- ▶ CBY, which is used with an ESCON to parallel converter for byte multiplexing control units.

The FSI serial adapter (ESCON) supports only CNC-type channel operation. You can use it in a channel-to-channel connection; however, the “other” channel must provide the CTC logic.

In practice, most ESCON channels are connected through ESCON Directors. To access a device in this environment, the host system must specify the following:

```
host channel number (CHPID) ---> output link on ESCON Director ---> CUADD--->UNITADD
```

That is, the channel number, the Director link address for the control unit, the control unit address, and the device address must be provided. Multiple paths may exist to a control unit and device, and each path may have a different channel and link address. In a S/390 or zSeries system, this addressing information exists in the *IOCDS*. The complete address path (or multiple paths) is embodied in the *device number* (sometimes known as the *software address*) that the operating system uses.

FLEX-ES systems do not have an IOCDS and the same addressing information specified in an IOCDS must be provided in the FLEX-ES definitions.

⁶ Early S/390 systems had ESCON channels that operated at a maximum of 10 megabytes per second.

FSI channel adapters

FSI channel adapters are PCI adapters and should be usable with any supported FLEX-ES system that has appropriate PCI adapter slots. There is no design limitation on the number of physical channels that you can use with FLEX-ES.

2.1 Serial channel adapters

Figure 2-1 shows a single-channel SCA card. (Not shown is the MTRJ to ESCON adapter cable (2 meters long) that is supplied with it.)

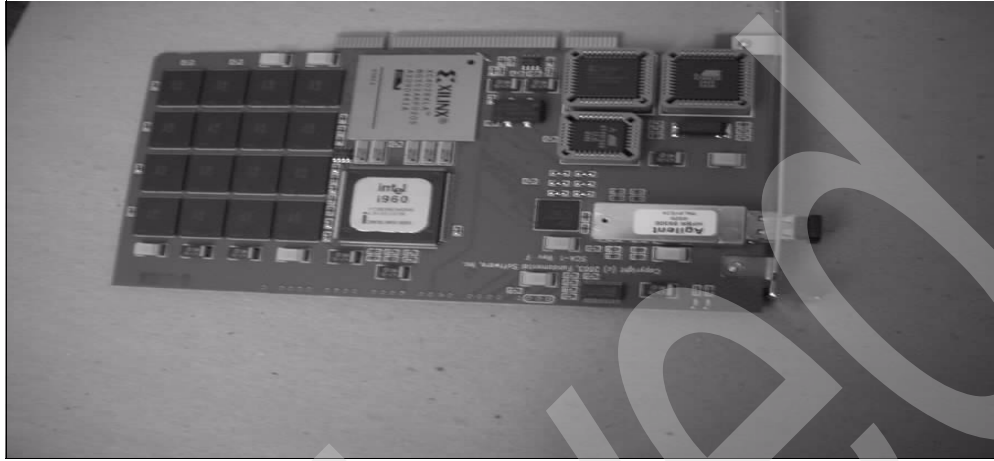


Figure 2-1 Single-channel SCA

The serial channel adapter is designed for a 3.3 volt PCI bus slot but can tolerate a 5 volt slot. It uses a 33 MHz bus interface and forces the bus to that speed if it is nominally a faster bus. This means that any other adapters on the same bus run at 33 MHz. If the host server has more than one PCI bus, then it is best to separate fast adapters (such as a ServeRaid adapter or the FSI 3-channel parallel adapter) onto a different bus.

Note: This is a half-length PCI adapter and requires a corresponding slot in the server used for FLEX-ES.

In general, these adapters fit in any PCI slot (with the required length) and as many adapters may be used as there are slots to hold them. Figure 2-2 shows the MTRJ to ESCON adapter cable that is included with the adapter.

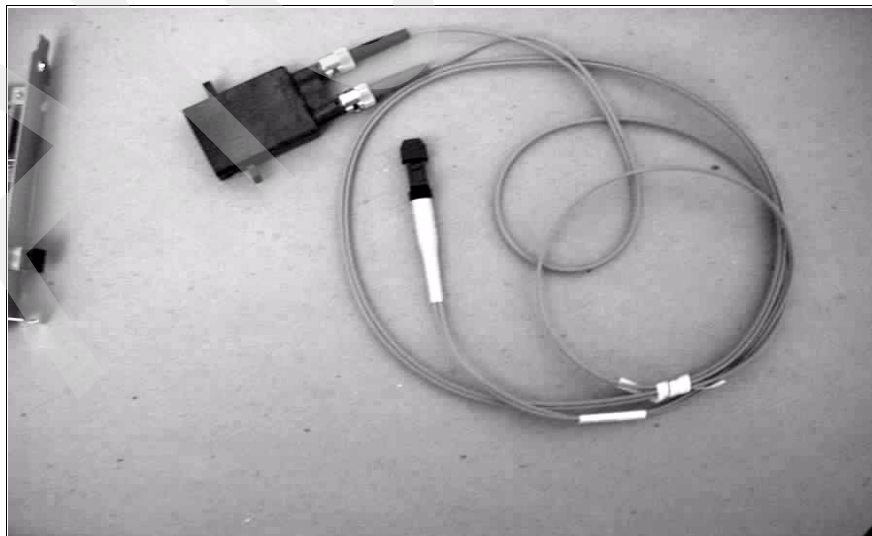


Figure 2-2 MTRJ to ESCON adapter cable

2.2 Parallel channel adapters

Parallel channels are defined in FLEX-ES as *blockmux*, *bytemux*, or *selector* channels. The bytemux and selector options are used with older devices in specialized situations and are not further described in this document.

Figure 2-3 shows a single-channel PCA and the associated connectors, clearly depicting the standard bus and tag connectors.

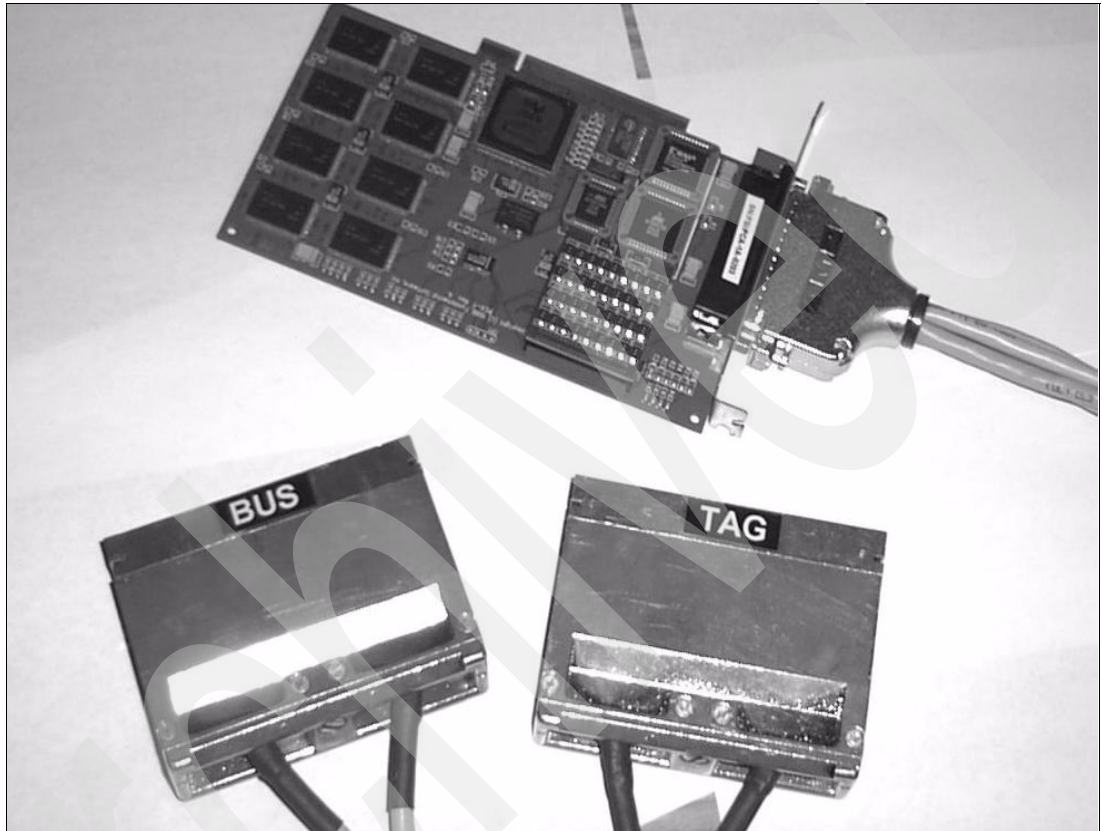


Figure 2-3 Single-channel parallel adapter and cables

The parallel channel characteristics are approximately the same as the IBM parallel channels. In a classical configuration, up to eight control units can be daisy chained together, provided maximum cable lengths are not exceeded. The maximum cable lengths depend on the cable types and the control units involved. Consult the standard IBM documentation for details.

The cable is an important part of the PCA. It matches a 62-pin D-shell connector on the PCA and provides standard parallel bus and tag connectors on the other end.

Figure 2-4 on page 8 shows a three-channel PCA.

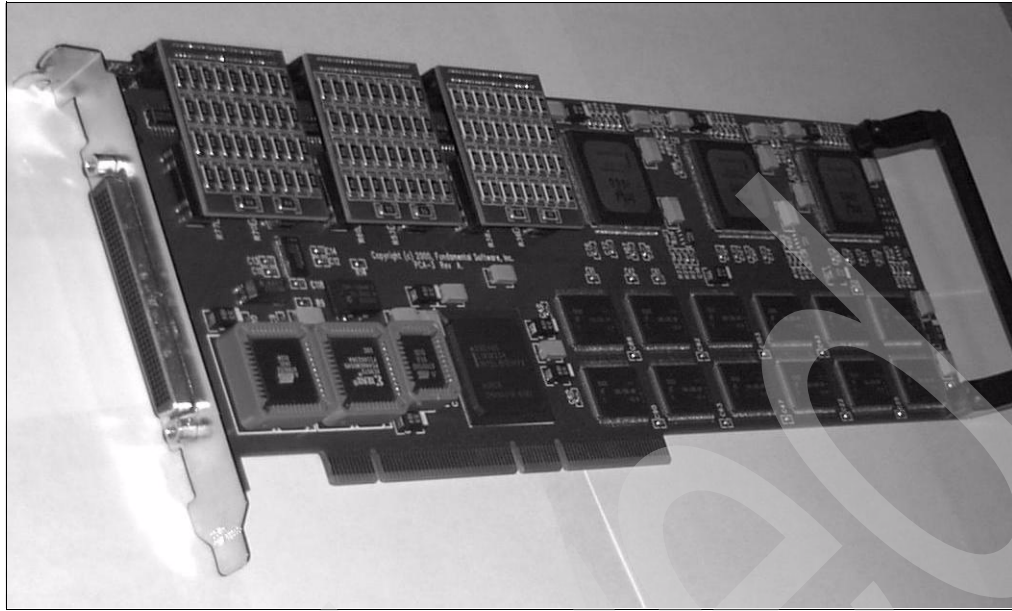


Figure 2-4 Three-channel parallel channel adapter

The adapter cable set for a three-channel adapter is shown in Figure 2-5.

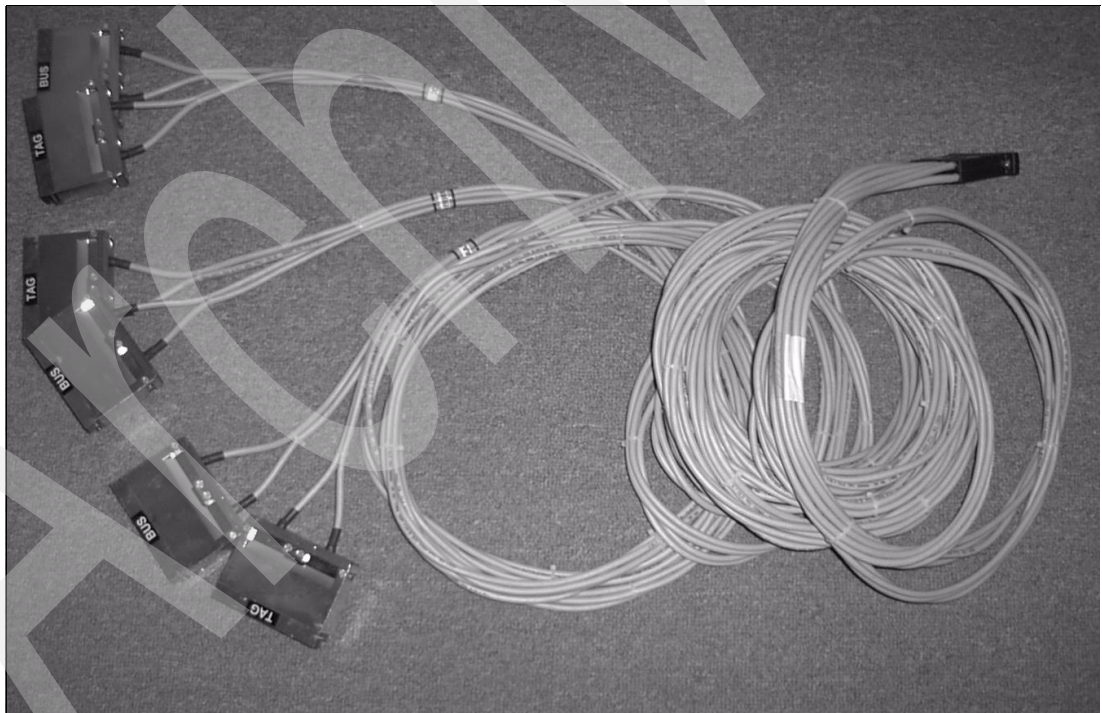


Figure 2-5 Cable set for three-channel parallel adapter

The “finger space” and cable space around the PCI adapters becomes significant, as shown in Figure 2-6 on page 9, and practical planning is needed. Parallel cables are heavy, so we recommend that you do not suspend them from the D-shell connectors on the adapter card.



Figure 2-6 Multiple parallel channel connections

There are many ways to approach cable management. FSI offers the strain relief device shown in Figure 2-7 on page 10. While not a high-technology solution, it is practical and helps solve the problem.



Figure 2-7 FSI strain relief device

The computer for this redbook project is shown in Figure 2-8 on page 11. The position of the LED on one of the serial adapters is noted in this illustration. The LEDs (on each adapter) are important for determining the identity of each serial adapter if more than one is present.

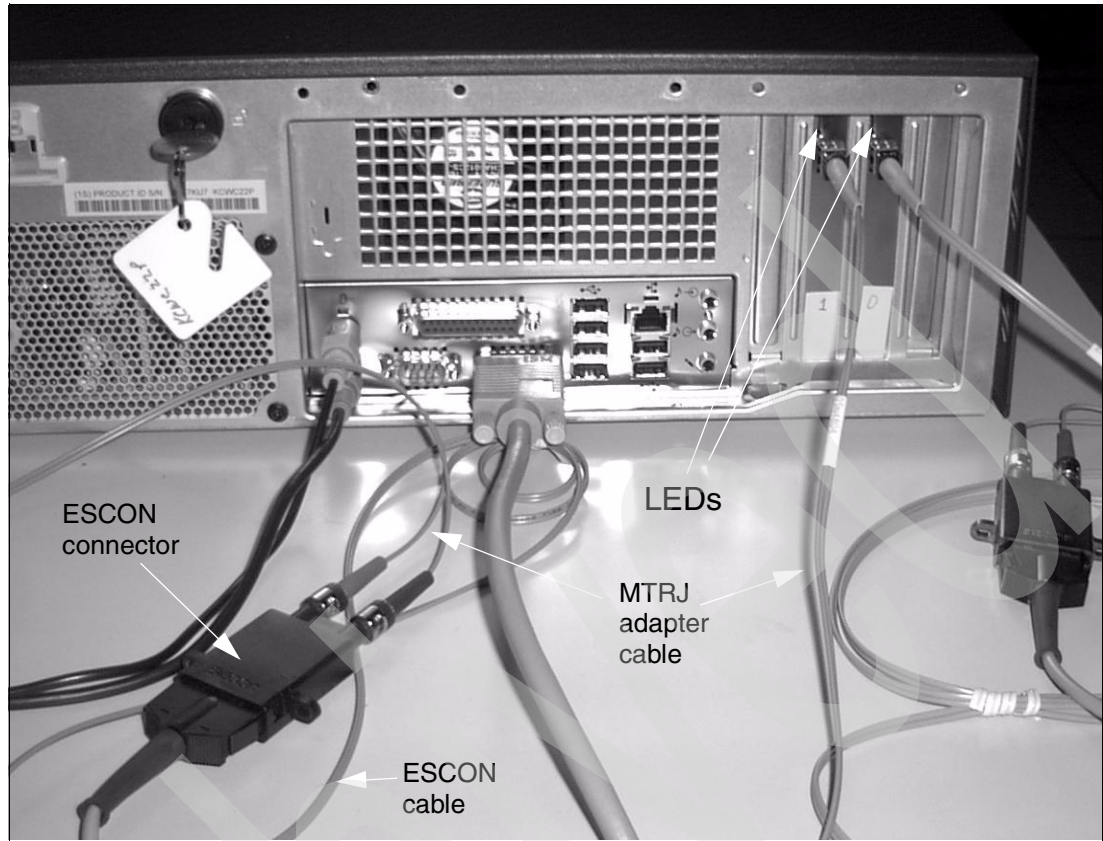


Figure 2-8 FLEX-ES system used for this redbook project

2.3 The rpm installation

FLEX-ES is distributed as several rpm files that are installed with **rpm** commands. To use the channel adapters, install the **chpbt** rpm and **memmgr** rpm files in addition to the three rpm files familiar to current Linux FLEX-ES users. Example 2-1 illustrates the installation process if we assume FLEX-ES release 7.0.1 is on a CD.

Example 2-1

```
(insert the CD and let RedHat automount it)
# ls /mnt/cdrom                                verify CD contents
# cd /tmp                                       untar into /tmp
# tar xvf /mnt/cdrom/flexes-7.0.1.i386.rpm.tar
flexes-7.0-1.i386.rpm
msgmgr-7.0-1.i386.rpm
memmgr-7.0-1.i386.rpm
chpbt-7.0-1.i386.rpm
fsiica-7.0-1.i386.rpm
ftlib-7.0-i386.rpm
# rpm -e ftlib                                remove old FLEX-ES release
# rpm -e msgmgr
# rpm -e flexes
# rpm -i flexes-7.0-1.i386.rpm                 install new release
# rpm -i msgmgr-7.0-1.i386.rpm
# rpm -i memmgr-7.0-1.i386.rpm
# rpm -i chpbt-7.0-1.i386.rpm
# rpm -i fsiica-7.0-1.i386.rpm
```

```
# rpm -i ftlib-7.0-1.i386.rpm
(install appropriate FLEX-ES license)
(reboot to pick up new msgmgr)
(recompile FLEX-ES definitions)
```

The **memmgr** and **chpbt** modules are needed only if an FSI channel adapter is used. The **fsiica** modules are needed only if an FSI ICA adapter is used.

2.4 The eprog program

The **eprog** program is included with FLEX-ES. Use it from *root*. Following is the usage:

```
eprog [-c <chnum>] <file>          # write firmware to EPROM and reconfigure
eprog [-c <chnum>] -b              # blink the LED for this channel
eprog [-c <chnum>] -q              # query channel
eprog [-c <chnum>] [-f] -r [<file>] # read firmware from EPROM to file
```

At the Linux level the physical channels are accessed through `/dev/chpbt/chn` special devices, starting with `ch0`, `ch1`, and so forth. These special device files are created when the FLEX-ES resource manager is started. The **eprog** command can determine which physical channel adapter corresponds to a particular channel number, as in Example 2-2:

Example 2-2

```
# eprog -c 1 -b                      (blink the LED for channel 1)
```

This command causes an LED on the specified channel adapter to blink for about 20 seconds. The LED is behind the back plate of the adapter, as identified in Figure 2-8 on page 11, and some care is needed to see it clearly.

Example 2-3 shows the output produced when using **eprog** to query a serial channel adapter.

Example 2-3

```
# eprog -c 0 -q                      (query channel 0)
Xilinx loaded: Y
File: lcy4315a.bin (10 Nov 2004)
Type: SCA-1
Location: bus=3 dev=9
```

The file name displayed represents the firmware (“microcode”) level on the adapter.

2.4.1 Firmware upgrades

FLEX-ES release 7.0.1 requires serial channel adapters to be at firmware level `lcy4350a` or later. The adapter in the previous example was not at the correct level. Firmware code is packaged in simple files that you can send via e-mail or download with ftp. The size of the code is in the 75K-100K range. You can install new code with **eprog**, as follows.

```
(copy the new firmware file to the current Linux directory)
(assume our new file is named lcy4350a.bin)
# eprog -c 0 lcy4350a.bin            (flash channel 0 to the new level)
  Stopping i960: finished
  Erasing EPROM: finished
  Writing EPROM: finished            (counts through about 325 steps for update)
  Reconfiguring: finished
# eprog -c 0 -q                      (query channel 0 for verification)
```

```
Xilinx loaded: Y
File: lcy4350a.bin (15 Dec 2004)
Type: SCA-1
Location: bus=3 dev=9
```

As with firmware updates for most device, it is critical that the system not be interrupted while the update installation is in progress. If the update is interrupted, it may be necessary to return the adapter to FSI for repair.

2.5 Linux messages

When booting Red Hat Linux 9.0, with the default kernel or with the 2.4.25 kernel built from the FSI-provided source tree, a number of messages similar to the following appear in the boot log:

```
WARNING: loading /usr/flexes/modules/memmgr.o will taint the kernel: forced load
WARNING: Kernel-module version mismatch
/usr/flexes/modules/chpbt.o was compiled for kernel version 2.4.18-14
WARNING: loading /usr/flexes/modules.memmgr.o will taint the kernel: no license
```

These messages are normal, and you can ignore them. You can also ignore messages about a mismatch between the kernel and the compiler level used for the FLEX-ES msgmgr module.

The following messages near the end of the boot-up process. They indicate that the indicated FSI channel adapters were properly detected.

```
chpbt: FSI PCI channel count = 2
chpbt: Bus 3 Device 9 Base e9000000 Length 1000000
chpbt: Bus 3 Device a Base ea000000 Length 1000000
chpbt: FLEX-ES channel driver initialization complete
```

Archived

Configuration as a channel

The FLEX-ES definitions for the channels are straight forward. Please note that the descriptions here apply only to FLEX-ES release 7.0 and later. A careful examination of the following sample definitions should provide most of the information needed:

```

system chan1:
instset(esa)                # examples in ESA mode
...
channel(0) local             # FLEX-ES emulated channel
channel(1) blockmux chpbt0   # parallel channel
channel(2) serial chpbt1     # ESCON channel
...
cu devad(0x700,2) path(0) resource(cu3174)      # 2 devices
cu devad(0x560,1) path(1) unitadd(0x00)        # 1 device
cu devad(0x300,32) path(2) cuadd(0x00) unitadd(0x00) link(0xc1) # 32 devices
...
end chan1

resources chan1:
...
chpbt0: blockmux
/dev/chpbt/ch0
end chpbt0
..
chpbt1: serial
/dev/chpbt/ch1
end chpbt1
..
cu3174: cu 3174
interface local(1)
device(00) 3278 mstcon
device(01) 3278 L701
end cu3174
..
end chan1

```

The **channel** definitions for physical channels are simple. They define the channel type (usually *blockmux* or *serial*) and the name of stanzas in the resources section of the

definition. The names of the stanzas (chpbt0 or chpbt1 in the example) are arbitrary. The resource stanzas specify the Linux name of each channel in the format /dev/chpbt/chx.

Every control unit FLEX-ES uses, whether emulated or real, must be precisely defined to FLEX-ES. The device count in the **cu** definition *must* match the number of devices configured in the control unit. This number may be more than the number of devices actually attached to the control unit. For example, a tape control unit might have four tape drives attached but configured for 16 in the control unit. In this case the **cu** statement *must* specify 16 devices.

Important: This requirement is important! An incorrect **cu** definition for physical channels can lead to obscure problems later.

Additional information is needed in the **cu** statements for physical channels. The exact combinations depend on the configuration to be used. Following is the complete set of optional parameters:

```
PCA:  unitadd(0x..) [interlocked] [shared] [sharedb] [streaming(20, 30, or 45)]
SCA:  cuadd(0x.) unitadd(0x..) [link(0x..)]
```

Example 3-1 Parameter examples

cu devad(0x480,8) path(1) unitadd(0x80) interlocked shared	#PCA
cu devad(0x500,16) path(1) unitadd(0x90) streaming(45)	#PCA
cu devad(0x300,16) path(2) unitadd(0x00) cuadd(0x0)	#SCA
cu devad(0x300,16) path(2) unitadd(0x00) cuadd(0x0) link(0xc1)	#SCA

The following rules apply for the parameters:

- ▶ The **interlocked**, **shared**, **sharedb**, and **streaming** parameters are needed for various IBM parallel channel devices. Specific parameters for common devices are suggested in *Resource Language Reference*, FSIMM310. (The streaming parameters 20, 30, and 45 indicate channel transfer speeds of 2.0, 3.0, and 4.5 MB/sec.)
- ▶ The **unitadd** parameter for parallel devices is two hex digits and generally combines the hardware address of the control unit (first digit) and the device address (second digit). The exact usage may vary depending on the type of control unit.
- ▶ The **unitadd** (two-hex digits) and **cuadd** (one-hex digit) for serial devices are the corresponding hardware values for the control unit and devices. (These values can often be obtained from an IOCP listing.)
- ▶ The **link** parameter is used only when the connection is through an ESCON Director. The parameter (two-hex digits) is the Director port number connected to the target control unit.
- ▶ The **devad** address (first parameter, three or four-hex digits) is the device number that the operating system sees running under FLEX-ES. In ESA mode this device number has no relationship with the hardware control unit addresses or device addresses. All the descriptions in this document assume the use of ESA mode.
- ▶ The **devad** count (second parameter, usually entered as a decimal number) *must* reflect the number of devices supported by the control unit. This may differ from the number of devices actually attached to the control unit.

Please note that these parameters must match the “real” parameters of the “real” devices. You can be creative with addresses for FLEX-ES emulated devices; however, that is not the case for the unitadd, cuadd, link, and device count parameters. These parameters reflect actual hardware characteristics of external units. The devad address parameter, which sets the address used by software under FLEX-ES, is not determined by external hardware and can be anything desired by the user.¹

¹ This statement assumes we are in XA mode where the device number (“software address” or “address”) is not related to hardware addresses.

It is important to understand that two different address spaces are involved in the **cu** definition for real channels and devices:

- ▶ The **devad** parameter specifies the device numbers (often referenced as “addresses” or “software addresses”) that the operating system uses. A FLEX-ES user can assign whatever device numbers the operating system needs. The hardware **unitadd**, **cuadd**, or **link** values do not control the device numbers.
- ▶ The **unitadd**, **cuadd**, and **link** parameters specify hardware-assigned values and must match the external control unit connected to the channel.

3.1 Configurations

Figure 3-1 illustrates several alternative configurations involving multiple SCA channels, multiple FLEX-ES instances, multiple disk control units, and an ESCON Director.

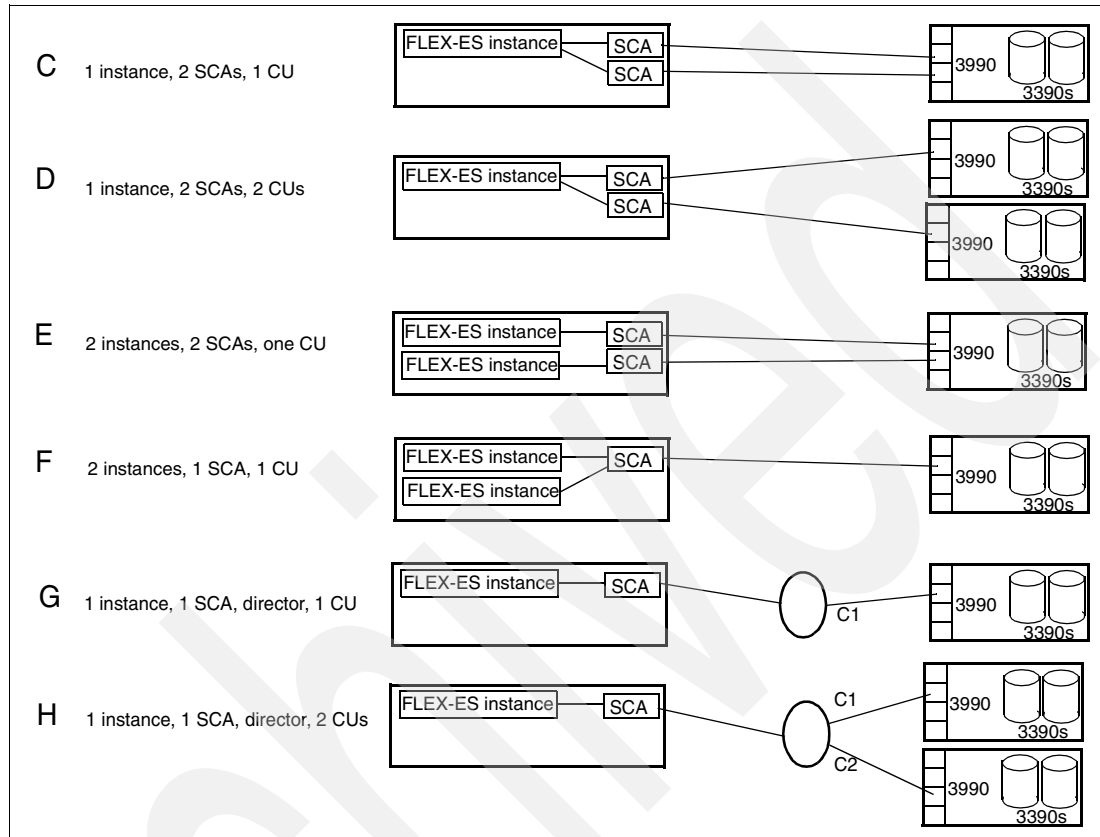


Figure 3-1 Selected alternative configurations

We made a few assumptions when describing the alternative configurations. We assume the CU address is 0x'0' (which is very common), the first device address is 0x'00' (also common), and that 16 units are defined in the control unit. For our examples, we assume the operating system running under FLEX-ES uses addresses in the 300 or 400 ranges for these units. The following descriptions provide only the relevant channel definition statements and `cu` definition statements.

The alternative configurations in Figure 3-1 on page 18 include the following:

- **C.** This configuration illustrates multiple paths to the same devices. *This function is not supported by the current FLEX-ES release.* The definition syntax is:

```
channel(1) serial s1
channel(2) serial s2
cu devad(0x300,16) path(1,2) unitadd(0x00) cuadd(0x0)
```

Again, *this configuration is not currently supported.* We illustrate it here because it represents one of the most common questions about FLEX-ES channel adapters.

- **D.** This simply provides separate channels for separate disk control units. The definitions might be:

```
channel(1) serial s1
channel(2) serial s2
cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0)
cu devad(0x400,16) path(2) unitadd(0x00) cuadd(0x0)
```

- **E.** This example has two FLEX-ES instances running independent z/OS images with shared DASD. The definitions might be:

```
System1:
  channel (1) serial s1
  cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0)
System2:
  channel (1) serial s2
  cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0)
...
s1: serial
/dev/chpbt/ch0
end s1
s2: serial
/dev/chpbt/ch1
end s2
```

Note that we elected to use the same disk addresses (starting at 300) in both z/OS systems. This is typical but not required.

- **F.** In this case we have two FLEX-ES instances sharing one SCA channel and one disk control unit. The definitions might be:

```
System1:
  channel (1) serial s1
  cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0)
System2:
  channel (1) serial s1
  cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0)
...
s1: serial
/dev/chpbt/ch0
end s1
```

This is somewhat equivalent to an MIF configuration in a zSeries machine. There is no requirement that the device numbers (0x300) be the same in the two systems.

- **G.** This is a simple use of an ESCON Director. The definitions might be:

```
channel(1) serial s1
cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0) link(0xC1)
```

- **H.** In this example we have one SCA channel going through an ESCON Director to access two control units. The definitions might be:

```
channel(1) serial s1
```

```
cu devad(0x300,16) path(1) unitadd(0x00) cuadd(0x0) link(0xC1)
cu devad(0x400,16) path(1) unitadd(0x00) cuadd(0x0) link(0xC2)
```

Remember that in an ESA-mode system (or zArchitecture system) the “software address” (which is determined by the devad operands) is completely independent of the “hardware addresses” in the unitadd or cuadd operands.

The diagrams used for the examples in this document include additional control unit interfaces. (Figure 3-1 on page 18 shows four interfaces, but this is an arbitrary illustration.) You can connect these interfaces to other systems (including other FLEX-ES systems) in the same manner that control units are shared with multiple S/390s. This is “business as usual” for system configurations.

3.2 Working example

We decided to connect a FLEX-ES system with two serial adapters to part of the ITSO floor system. The ITSO systems programmer sent us the following example extract from the IOCP used in all the ITSO floor systems.

Example 3-2 Example extract from the IOCP

```
CNTLUNIT CUNUMBR=2500,PATH=((CSS(0),7A,7E),(CSS(1),7A,7E)), *
      UNITADD=((00,064)),LINK=((CSS(0),C5,DE),(CSS(1),C5,DE)),*
      CUADD=0,UNIT=3990
CNTLUNIT CUNUMBR=2501,PATH=((CSS(0),7B,7F),(CSS(1),7B,7F)), *
      UNITADD=((00,064)),LINK=((CSS(0),D0,9D),(CSS(1),D0,9D)),*
      CUADD=0,UNIT=3990
IODEVICE ADDRESS=(2500,064),CUNUMBR=(2500,2501),STADET=Y, *
      UNIT=3390
CNTLUNIT CUNUMBR=2540,PATH=((CSS(0),7A,7E),(CSS(1),7A,7E)), *
      UNITADD=((00,064)),LINK=((CSS(0),C5,DE),(CSS(1),C5,DE)),*
      CUADD=1,UNIT=3990
CNTLUNIT CUNUMBR=2541,PATH=((CSS(0),7B,7F),(CSS(1),7B,7F)), *
      UNITADD=((00,064)),LINK=((CSS(0),D0,9D),(CSS(1),D0,9D)),*
      CUADD=1,UNIT=3990
IODEVICE ADDRESS=(2540,064),UNITADD=00,CUNUMBR=(2540,2541), *
      STADET=Y,UNIT=3390
CNTLUNIT CUNUMBR=2580,PATH=((CSS(0),7A,7E),(CSS(1),7A,7E)), *
      UNITADD=((00,064)),LINK=((CSS(0),C5,DE),(CSS(1),C5,DE)),*
      CUADD=2,UNIT=3990
CNTLUNIT CUNUMBR=2581,PATH=((CSS(0),7B,7F),(CSS(1),7B,7F)), *
      UNITADD=((00,064)),LINK=((CSS(0),D0,9D),(CSS(1),D0,9D)),*
      CUADD=2,UNIT=3990
IODEVICE ADDRESS=(2580,064),UNITADD=00,CUNUMBR=(2580,2581), *
      STADET=Y,UNIT=3390
CNTLUNIT CUNUMBR=25C0,PATH=((CSS(0),7A,7E),(CSS(1),7A,7E)), *
      UNITADD=((00,064)),LINK=((CSS(0),C5,DE),(CSS(1),C5,DE)),*
      CUADD=3,UNIT=3990
CNTLUNIT CUNUMBR=25C1,PATH=((CSS(0),7B,7F),(CSS(1),7B,7F)), *
      UNITADD=((00,064)),LINK=((CSS(0),D0,9D),(CSS(1),D0,9D)),*
      CUADD=3,UNIT=3990
IODEVICE ADDRESS=(25C0,064),UNITADD=00,CUNUMBR=(25C0,25C1), *
      STADET=Y,UNIT=3390
CNTLUNIT CUNUMBR=2600,PATH=((CSS(0),7E,7F),(CSS(1),7E,7F)), *
      UNITADD=((00,128)),LINK=((CSS(0),84,0F),(CSS(1),84,0F)),*
      UNIT=3990-6
IODEVICE ADDRESS=(2600,128),CUNUMBR=(2600),STADET=Y,UNIT=3390
CNTLUNIT CUNUMBR=3800,PATH=((CSS(0),46,47),(CSS(1),46,47)), *
```

```

UNITADD=((00,256)),LINK=((CSS(0),C4,C4),(CSS(1),C4,C4)),*
CUADD=0,UNIT=3990-6
CNTLUNIT CUNUMBR=3801,PATH=((CSS(0),4A,4B),(CSS(1),4A,4B)),*
UNITADD=((00,256)),LINK=((CSS(0),C4,C4),(CSS(1),C4,C4)),*
CUADD=0,UNIT=3990-6
IODEVICE ADDRESS=(3800,256),CUNUMBR=(3800,3801),STADET=Y,*
UNIT=3390

```

We were also provided with the following information:

- ▶ One FLEX-ES serial channel connects to Director AF port 21 and the other serial channel connects to Director AB port FB.
- ▶ The 25xx devices are connected to switch AB port D0 and switch AF port 9D
- ▶ The 26xx devices are connected to switch AF port 8E.
- ▶ The 38xx devices are connected to switch AB port C4.

Figure 3-2 illustrates Example 3-2 on page 20:

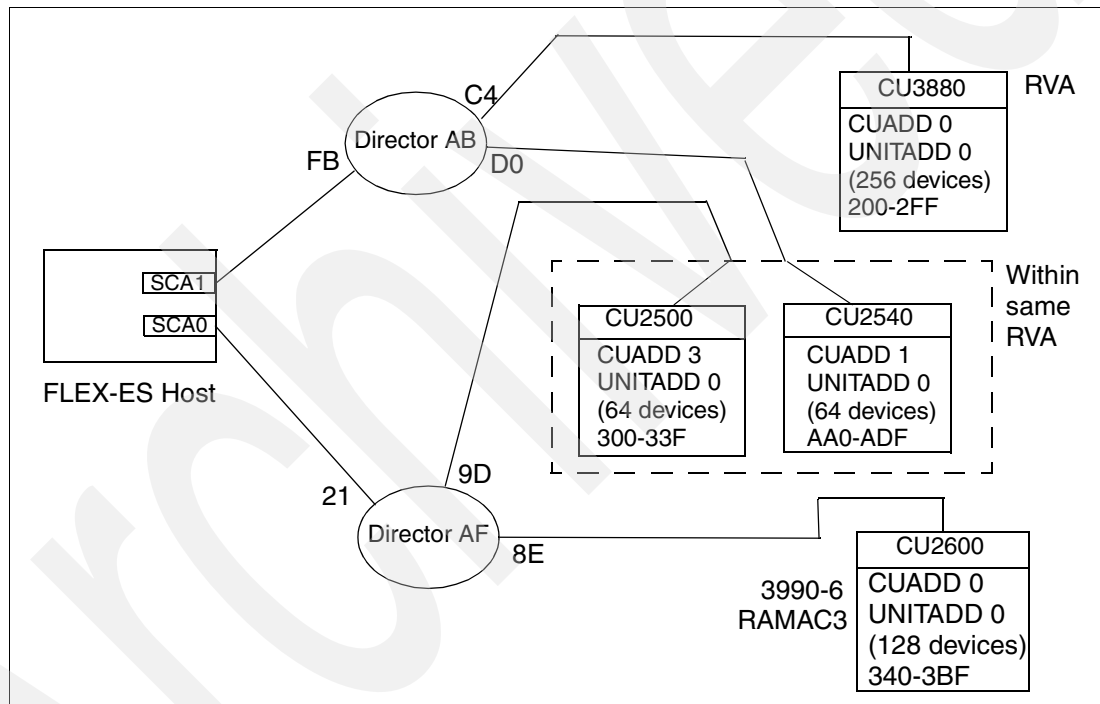


Figure 3-2 Graphic depiction of example

We started with the smallest set of FLEX-ES definitions that we could use, as follows:

```

system R15s:
memsize(524288)
cachesize(4096)
instset(z)
cpu(0)
channel(0) local
channel(2) local
channel(5) serial sca0 # switch AF port 21
cu devad(0x340,128) path(5) unitadd(0x00) cuadd(0x00) link(0x8E)
cu devad(0xA80,1) path(2) resource(CU3990)
cu devad(0x700,3) path(0) resource(CU3174)
end R15s

```

```

resources R15s:
sca0:
serial /dev/chpbt/ch0
end sca0

CU3990: cu 3990
interface local(1)
device(00) 3390-3 /s390/SARES1      # our z/OS volume
end CU3990

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

end R15s

```

We used the single-volume z/OS system (volser SARES1) that is supplied with the AD CD-ROM package. We connected a single serial channel to a single control unit. After IPLing, we could access volumes on control unit CU2600. This assured us that the channel and definitions were configured correctly. We next expanded the FLEX-ES definitions, as follows:

```

system R15:
memsize(524288)
cachesize(4096)
instset(z)
cpu(0)
channel(0) local
channel(1) local
channel(2) local
channel(5) serial sca0 # switch AF port 21
channel(6) serial sca1 # switch AB port FB
cu devad(0x300,64) path(5) unitadd(0x00) cuadd(0x03) link(0x9D)
cu devad(0x340,128) path(5) unitadd(0x00) cuadd(0x00) link(0x8E)
cu devad(0xAA0,64) path(6) unitadd(0x00) cuadd(0x01) link(0xD0)
cu devad(0x200,256) path(6) unitadd(0x00) cuadd(0x00) link(0xC4)
cu devad(0xA80,16) path(2) resource(CU3990)
cu devad(0xA90,4) path(2) resource(CU3991)
cu devad(0x700,4) path(0) resource(CU3174)
cu devad(0xE20,2) path(1) resource(CU3172)
cu devad(0x560,1) path(2) resource(CU3480)
end R15

resources R15:
sca0:
serial /dev/chpbt/ch0
end sca0

sca1:
serial /dev/chpbt/ch1
end sca1

CU3990: cu 3990
interface local(1)
device(00) 3390-3 /s390/Z5RES1
device(01) 3390-3 /s390/Z5RES2
device(02) 3390-3 /s390/OS39M1
device(03) 3390-3 OFFLINE
device(04) 3390-2 OFFLINE

```



```

device(05) 3390-3 OFFLINE
device(06) 3390-3 OFFLINE
device(07) 3390-3 /s390/Z5USS1
device(08) 3390-3 OFFLINE
device(09) 3390-2 OFFLINE
device(10) 3390-3 OFFLINE
device(11) 3390-3 OFFLINE
device(12) 3390-3 /s390/SARES1
device(13) 3390-3 OFFLINE
device(14) 3390-3 OFFLINE
device(15) 3390-3 /s390/Z5USS2
end CU3990

```

```

CU3991: cu 3990
interface local(1)
device(00) 3390-3 OFFLINE
device(01) 3390-1 OFFLINE
device(02) 3390-1 /s390/WORK01
device(03) 3390-1 /s390/WORK02
end CU3991

```

```

CU3174: cu 3174
interface local(1)
device(00) 3278 mstcon
device(01) 3278 L701
device(02) 3278 L702
device(03) 3278 L703
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 R15

```

The AD CD-ROM system (IPLed from Z5RES1) did not have enough 3390 addresses for our configuration; instead, we used HCD to build a new IODF with additional 3390 devices at addresses 200-2FF. After building the new IODF, we modified the LOADCS entry in SYS1.IPLPARM to use the new IODF and then IPLed with it. This provided addressability for all the disks on our serial channels.

3.3 Logical control unit paths

We were unable to access CU2500 and CU2540, which is shown in Figure 3-2 on page 21. Simple **VARY ONLINE** commands indicated no physical paths existed to devices on these control units. Two z/OS commands can be especially useful in this situation:

```

d m=dev(AA0)
  DEVICE 0AA0 STATUS=OFFLINE
  CHP                                05

```

```

DEST LINK ADDRESS      00          <--- note this value
PATH ONLINE            Y
CHP PHYSICALLY ONLINE  Y
PATH OPERATIONAL       N
PATHS NOT VALIDATED
ds p,aa0
UNIT DTYPE  M CNT  VOLSER  CHPID=PATH STATUS
RTYPE  SSID  CFW TC  DFW  PIN DC-STATE  CCA  DDC  AUT  CU-TYPE
0AA0,3390,  F,000,          ,05=L
*****SYMBOL DEFINITIONS*****
F=OFFLINE  <PHYSICALLY UNAVAILABLE

```

We found the diagnostic information from the three commands (**VARY ONLINE**, **D M=DEV**, and **DS p**) to be confusing. The **d m** command indicated that the channel and path were online but the indicated link address was incorrect (00 when it should be 9D). Two commands indicated the units had no physical connections. The **ds** command provided no information about the control unit.

With some detective work we determined that this combination of information indicates that no *logical* paths were available to the control units. Control units have a maximum number of logical paths (unique connections) that may be used. Each unique LPAR and CHPID combination accessing a control unit uses a separate logical path. Once a logical path is used it is typically not released until the associated LPAR is deactivated. An installation with many LPARs (on one or more systems) can exhaust the logical paths available on a control unit.

We found that the **D M=DEV** command does not display the link address when running under FLEX-ES.

Example 3-3 shows possible complete output from the **d m=dev** and **ds p** commands, with a properly working channel, control unit, and device:

Example 3-3 Possible complete output from the commands

```

12.19.02          d m=dev(200)
12.19.02          IEE174I 12.19.02 DISPLAY M 567                      C
DEVICE 0200      STATUS=ONLINE
CHP          06
DEST LINK ADDRESS 00
PATH ONLINE    Y
CHP PHYSICALLY ONLINE Y
PATH OPERATIONAL Y
MANAGED        N
MAXIMUM MANAGED CHPID(S) ALLOWED: 0
DESTINATION CU LOGICAL ADDRESS = 00
CU ND          = 009393.002.IBM.91.000001325025.0040
DEVICE NED = 003390.A38.IBM.91.000001325025.0000
00- 12.19.15      ds p,200
12.19.15          IEE459I 12.19.15 DEVSERV PATHS 570                  C
UNIT DTYPE  M CNT  VOLSER  CHPID=PATH STATUS
RTYPE  SSID  CFW TC  DFW  PIN DC-STATE  CCA  DDC  ALT  CU-TYPE
0200,33903 ,0,000,IODFPK,06=+
339034 00F8 Y YY. YY.  N  SIMPLEX  00  00          3990-3
*****SYMBOL DEFINITIONS *****
O = ONLINE          + = PATH AVAILABLE

```

Configuration as a CU

You can use a FLEX-ES system as channel-attached control unit(s) for a S/390 or zSeries machine. This is sometimes known as the FLEX-ES *control unit behavior*.

For example, the FLEX-ES system can emulate a number of 3990 control units and associated 3390 disks. In this case the FLEX-ES definitions need not include a *system* definition; instead, only *resources* definitions are needed if the only use of FLEX-ES is to provide this emulation. If FLEX-ES S/390 instance(s) and host-connected control unit emulation are both used, then both system and resource definitions are required.

Considering only host-connected control unit emulation, a simple definition might be as follows:

```
resources cut1:

cu3990a: cu 3990
interface serial sca0 addr(0x00) cuadd(0) path(0)
device(00) 3390-3 /dev/raw/300
device(01) 3390-3 /dev/raw/301
end cu3990a

sca0: chanpath
/dev/chpbt/ch0 serial
end sca0

end cut1
```

This definition corresponds to Figure 4-1 on page 26.

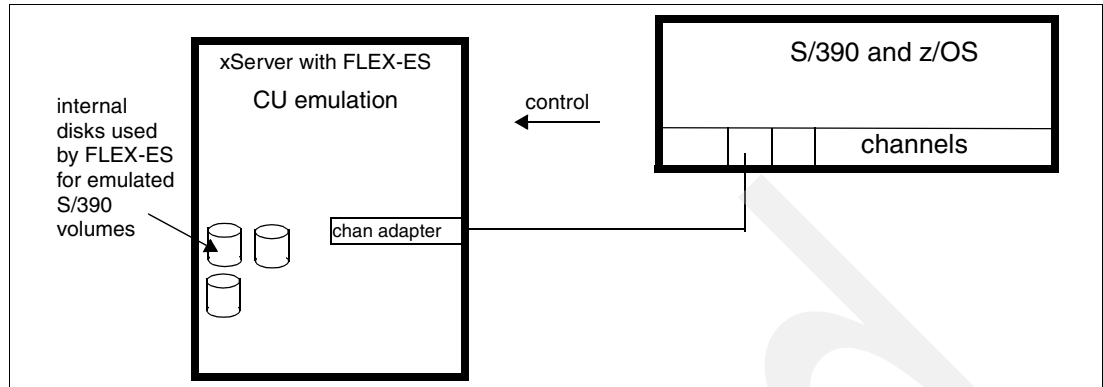


Figure 4-1 Simple host-connected control unit emulation

This definition has the following special characteristics:

- ▶ The **cu** interface statement includes **addr** and **cuadd** operands that provide the hardware control unit address and device address that the channel and host system sees. (The **addr** parameter applies to the first device. The **addr** value increments by one for each additional device defined for the control unit.)
- ▶ The interface parameters include the name of a **chanpath** definition. This is a resource that specifies the Linux name of the channel (**/dev/chpbt/ch0** in the example) and the type of channel (serial, meaning ESCON).
- ▶ The definition uses raw device definitions for the emulated disk drives, which is recommended for performance. However, you can also use simple Linux files.
- ▶ We explain the **path** parameter in a later section. It is *not* the same as a path parameter in a **cu** definition statement.
- ▶ No addresses (“software addresses,” device numbers) exist in the definitions. The IOCDS on the host system defines the device numbers, which are relevant only to the host system.

Figure 4-2 represents a larger use of emulated control units.

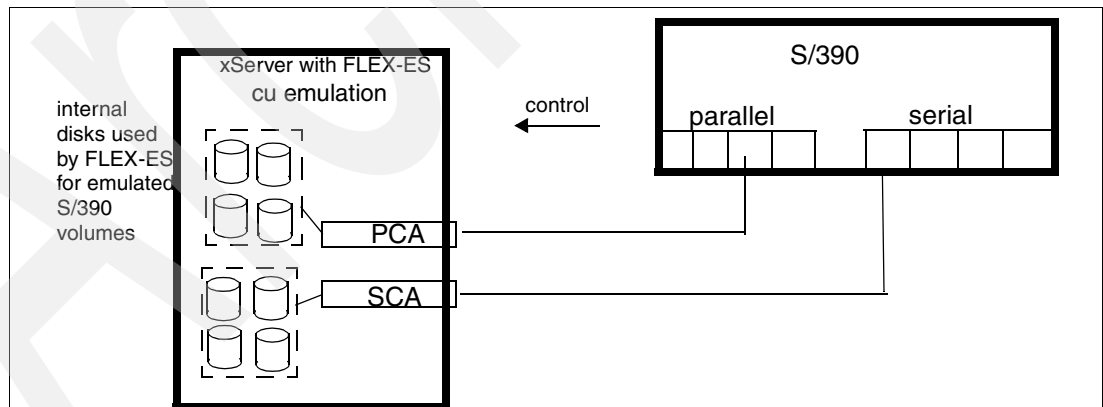


Figure 4-2 Logical Control Units

resources curesources:

...

cu3990: cu 3990

interface parallel pca0 addr(0xa0) streaming(45)

device(00) 3390-3 /dev/raw/3A0

device(01) 3390-3 /dev/raw/3A1

device(02) 3390-3 /dev/raw/3A2

```

device(03) 3390-3 /dev/raw/3A3
end cu3990

cu3990a: cu 3990
interface serial sca0 addr(0x00) path(0) cuadd(0)
interface serial sca0 addr(0x00) path(1) cuadd(0)
interface serial sca0 addr(0x00) path(2) cuadd(0)
interface serial sca0 addr(0x00) path(3) cuadd(0)
device(00) 3390-3 /dev/raw/3B0
device(01) 3390-3 /dev/raw/3B1
device(02) 3390-3 /dev/raw/3B2
device(03) 3390-3 /dev/raw/3B3
end cu3990a

pca0: chanpath
/dev/chpbt/ch0 blockmux
end pca0
sca0: chanpath
/dev/chpbt/ch1 serial
end sca0
end curesources

```

We have one PCA and one SCA. Notice the following points:

- ▶ Parallel channels are defined as blockmux in the channel resource.
- ▶ Multiple interfaces are defined for a control unit.
- ▶ The **path** parameter in the interface statements refer to a logical control unit path through the ESCON channel. Do not confuse this with path parameters in FLEX-ES *system* definitions.
- ▶ If we also wanted to use a FLEX-ES S/390 instance, we could add a **local** interface (or **network** interface) to the resource definitions (with corresponding **channel** and **devad** system definitions).
- ▶ Only one logical control unit path is defined for the parallel channel, and four logical control unit paths are defined for the serial channel.
- ▶ The “address names” for the raw devices (such as 3A0, 3A1, 3B0, 3B1, and so forth) are simply arbitrary names with no inherent meaning. They are not communicated to the host as addresses. Good planning might result in these being the device numbers used on the host, but this is not required.

4.1 Control unit path numbers

FLEX-ES emulated control units (such as a 3990 control unit with associated 3390 disk volumes) can handle a maximum of 32 separate logical control unit paths. For typical FLEX-ES systems (but without channel adapters), this number is not a concern. An unusually complex FLEX-ES environment, with multiple S/390 instances and additional network-connected FLEX-ES systems, might include a resource definition similar to the following.

```
cu399X: cu 3990
interface local(2)
interface network(3)
device(00) 3390-3 /s390/XXXXXX
...
```

This means there are two local **cu** definitions pointing to this emulated control unit (probably from two S/390 instances being emulated on the local machine) and three remote (network-connected) **cu** definitions pointing to this emulated control unit. In this case, five of the 32 maximum interface paths to the control unit are being used.

Consider a zSeries system with two LPARs, as illustrated in Figure 4-3.

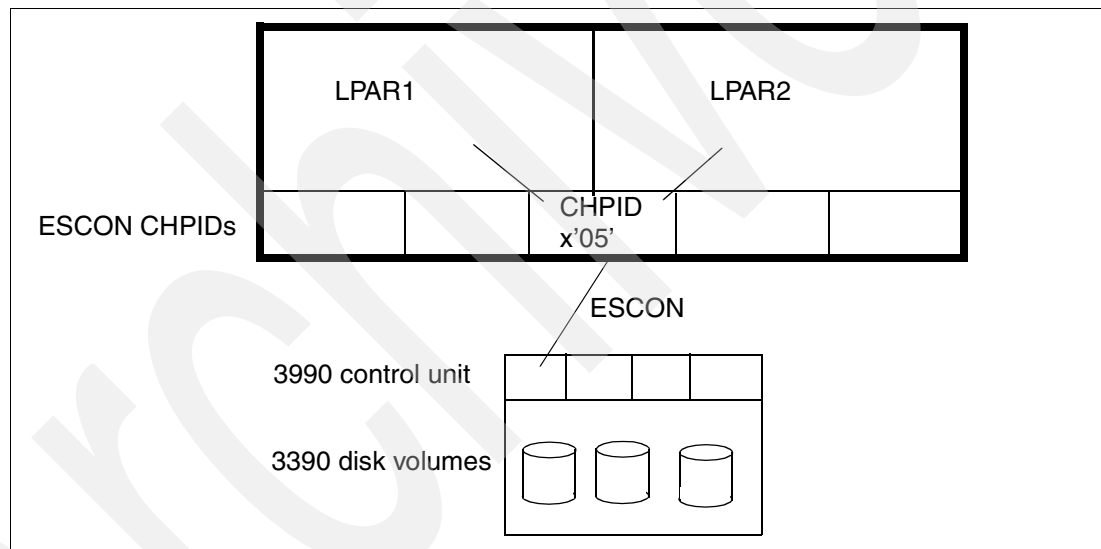


Figure 4-3 Control unit used by multiple LPARs

This configuration, in which multiple LPARs share a single CHPID link to a control unit, is commonly known as a MIF or EMIF configuration.¹ In this case the control unit must keep track of which LPAR requested an I/O operation so that data and interrupts are returned to the correct LPAR. ESCON frames sent from a host channel automatically include an LPAR identifier in every frame.² The frames, by the time they arrive at the control unit, also include the ESCON Director port number of the originating CHPID if the connection is through a Director. The combination of the source director port number and the LPAR number provides a unique “return address” or *path* for data and control signals. Modern control units can keep track of many such paths.

¹ MIF is for Multiple Interface Facility and EMIF is for Extended Multiple Interface Facility. For practical purposes both terms mean the same thing. There is now a tendency to consolidate the usage as MIF and drop usage of EMIF.

² This is properly known as the EMIF or MIF identifier. These have a one-to-one correspondence to LPAR numbers. These numbers are assigned in slightly different ways on different S/390 and zSeries processors.

FLEX-ES emulated control units can handle up to 32 such paths. If the emulated control unit is used by FLEX-ES S/390 images (local and network connected), these paths count toward the maximum. If no FLEX-ES images exist then up to 32 paths may be specified for host usage of the emulated control unit (assuming we are discussing a parallel or serial channel).

Each combination of an LPAR number (MIF number) and an ESCON port number (if used) constitutes a different path.

Path numbers are in the range 0-31 and are arbitrarily assigned within this range. However, a given path number is used only once for an emulated control unit. The following is incorrect:

```
cu3990a: cu 3990
interface serial cupbt0 cuadd(0) path(0) addr(0x00)
interface serial cupbt0 cuadd(0) path(0) addr(0x00) # WRONG!
device(00) 3390-3 OFFLINE
end cu3990a

cupbt0: chanpath
/dev/chpbt/ch0 serial
end cupbt0
```

In this case the same path number is used twice for the same control unit. If the second interface statement is changed to path(1), the definition is correct and produces two logical paths to the control units on the same channel. This is appropriate, for example, if two LPARs are using the control unit.

Each path definition consumes a small amount of storage within FLEX-ES and adds a very small amount of overhead to channel operations. If you have multiple LPARs (or multiple host systems) accessing a FLEX-ES emulated control unit, define a reasonable number of paths and do not worry if your estimate is slightly high. If you do not provide enough paths, any system attempting to use the control unit (after earlier systems have occupied all the available paths) will find the control unit unavailable.

Once an LPAR and Director path use a path, that path remains dedicated to that usage. Paths are not released dynamically. A path is released when the owning LPAR is deactivated or by certain operator commands affecting the channel.

4.2 Multiple control units

Parallel channels may be connected to several control units ("daisy chained"), while serial channels provide only point-to-point connections. However, this does not prevent a serial channel from addressing several control units, provided they all use the same serial connection. This is illustrated in Figure 4-4 on page 30.

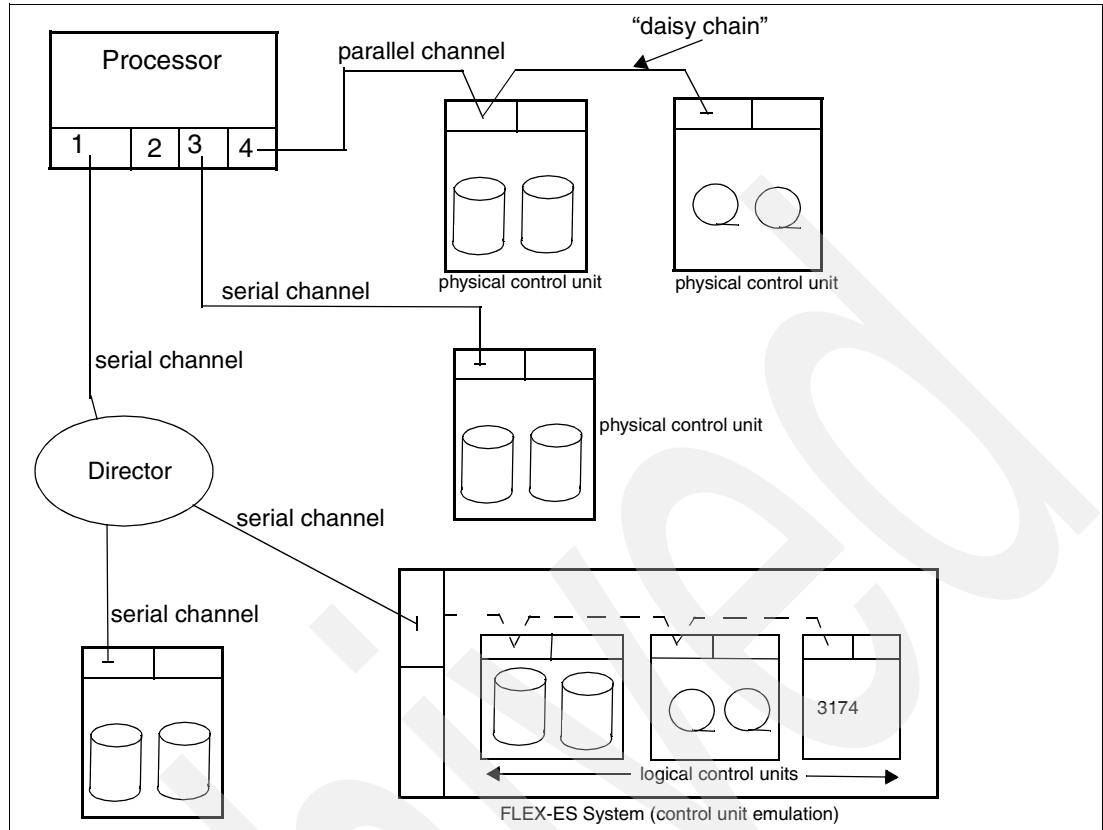


Figure 4-4 Point-to-point and daisy chaining

In Figure 4-4, channels 1 and 3 are serial (ESCON), and channel 4 is a parallel channel.³ The parallel channel is connected to a disk control unit and then “daisy chained” to a tape control unit. You can connect FLEX-ES parallel channels in the same way as traditional mainframe parallel channels.

Serial channels are point to point. They cannot be daisy chained. An ESCON Director can allow a serial channel to connect to multiple control units, but only one control unit is logically connected at any instant.⁴ The connection from the processor channel to a port in a Director is one point-to-point connection. A connection from an ESCON Director port to a control unit is another point-to-point connection. FLEX-ES serial channels operate in the same manner, providing a point-to-point connection to a control unit or Director.

A FLEX-ES system providing CU emulation might have a single physical channel adapter (serial or parallel). FLEX-ES can provide multiple control unit emulation through this single connection. Figure 4-4 illustrates a (single) serial connection to a FLEX-ES system that is emulating three control units. In a sense, the emulated FLEX-ES control units are daisy chained together. Each emulated control unit must have a different CU address, of course.

³ For the purposes of this discussion we are ignoring the fact that parallel channels are not available for later zSeries machines.

⁴ This is an over simplification of potentially complex multiplexing capabilities of serial channels, of course.

The definitions for the FLEX-ES portion of this environment (ignoring any ESCON Director, for simplicity) might be as follows:

IOCP statements:

```
CNTLUNIT CUNUMBR=0450,PATH=(1),UNIT=3990,UNITADD=((00,32)),CUADD=0
CNTLUNIT CUNUMBR=0451,PATH=(1),UNIT=3480,UNITADD=((00,32)),CUADD=1
CNTLUNIT CUNUMBR=0452,PATH=(1),UNIT=3174,UNITADD=((00,32)),CUADD=2
```

FLEX-ES resource definitions:

```
CU3990A: cu 3990
interface serial esconcu0 cuadd(0x00) addr(0x00) path(0)
device(00) 3390-3 /dev/raw/300
...
end CU3990A

CU3480A: cu 3480
interface serial esconcu0 cuadd(0x01) addr(0x00) path(0)
device(00) 3480 OFFLINE
...
end CU3480A

CU3174A: cu 3174
interface serial esconcu0 cuadd(0x02) addr(0x00) path(0)
device(00) 3278 L900
...
end CU3174A

esconcu0:
chanpath /dev/chpbr/ch0 serial
end esconcu0
```

In Figure 4-5 on page 32 the stanza names (such as CU3480A) are arbitrary names. Notice that the three FLEX-ES emulated control units are connected to the same serial channel adapter (esconcu0) and each has a different cuadd parameter. The path parameters in the interface statements relate only to the control unit with the interface statement. They are not related to the IOCP path parameter in any way.

4.2.1 Shared DASD with emulation

FLEX-ES emulated control units are shared between a FLEX-ES S/390 instance (or more than one instance) and a host system, as illustrated in Figure 4-5 on page 32. In this case we have a FLEX-ES S/390 image running z/OS, for example, and a zSeries processor running z/OS. Both are sharing a DASD control unit that happens to be a FLEX-ES emulated control unit. This emulated control unit has two (logical) interfaces.

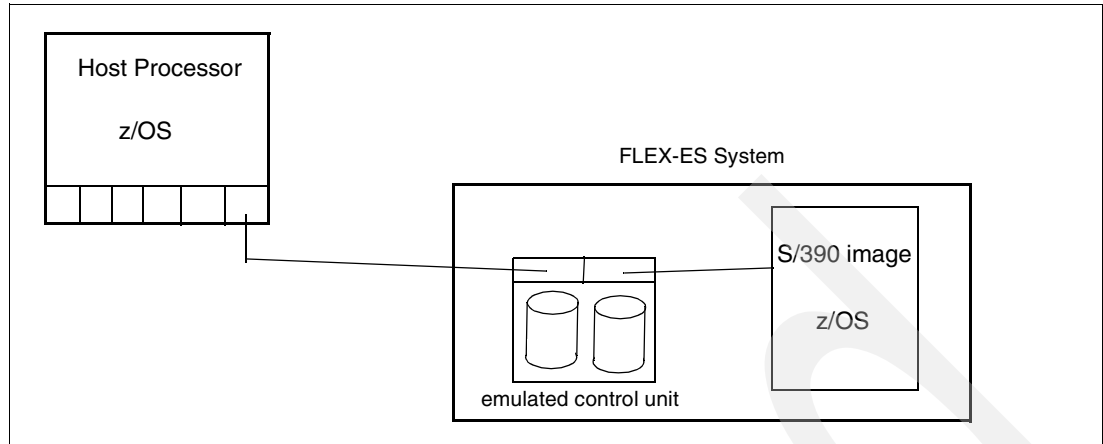


Figure 4-5 FLEX-ES and host sharing

This is a classical shared DASD situation and both z/OS systems define the control unit devices as “shared”. There is no GRS connection in this simple illustration, so the workloads on the two z/OS systems should be manually coordinated to prevent data set corruption.

The FLEX-ES definitions for this scenario might contain the following.

```
channel(1) local
cu devad(0x300,2) path(1) resource(CU3990B)
...
...
CU3990B: cu 3990
interface local(1)
interface serial esconcu1 cuadd(0x00)
device(00) 3390-3 /dev/raw/300
device(01) 3390-3 /dev/raw/301
end CU3990B
```

FLEX-ES releases

At the time of writing, there were two current FLEX-ES releases: 6.2.18 and 7.0.2. Release 7.0.1 (or later) is required for the channel adapter usage described in this document.

5.1 Release 6.2.18

This is the general-usage release available at the time of this writing. The significant enhancement in this release (and in the previous 6.2.17 release) is the availability of the z/Architecture™ Long-Displacement Facility. This facility provides a large number of new zSeries instructions. Currently, the only known user of these instructions is DB2® Version 8. However, future z/OS releases and future program product releases are expected to expand the use of these instructions.

You can find details about the new instructions in recent *Principles of Operation* manuals.

This FLEX-ES release also includes a number of operational enhancements for various SCSI tape drives.

5.2 Release 7.0.2

This release incorporates the improvements in earlier releases and adds the following:

- ▶ Release 7.0.2 now supports the z/Architecture ASN-and-LX Reuse Facility (starting with release 7.0.1).
- ▶ Release 7.0.2's major FLEX-ES redesign supports the FSI channel adapters under Linux. Earlier support of the parallel channel adapter (PCA) under UnixWare involved a restrictive memory management technique. This was removed, and no special memory management is needed when using channel adapters. The external effects of this redesign appear minor but a large amount of FLEX-ES internal rework was involved.
- ▶ Release 7.0.2 provides additional minor improvements for various SCSI tape drives.
- ▶ Release 7.0.2 removed the restriction about "hot plugging" the FLEX-ES dongle (in releases 6.2.18 and 7.0.1).

- ▶ Release 7.0.2 contains options to create AWS-compatible faketape files, which were extended, and several faketape utility programs can now compress/decompress these files. Another AWS-related option allows the user to specify the segment size to be used that can range from 1 to 64K-1 with a default value of 4096.
- ▶ Release 7.0.2 has minor command and utility improvements. Two examples of the improvements are:
 - Ranges of device numbers may be used in FLEX-ES CLI commands such as **display devstate A80-A88**.
 - A new -v option for the ckddump program causes it to display the volser of the emulated CKD volume.

Channel firmware levels required for this release are lcy5044a.bin (for SCAs) and cux5043c.bin (for PCAs).

FAQ

Q: Can I use my existing FSI parallel channel adapters with this new software?

A: Yes.

Q: Do I need to update the firmware in my existing channel adapters?

A: Yes.

Q: After I perform the firmware updates, can I use the adapter with my previous FLEX-ES release?

A: No. However, the firmware update process also produces a file containing your existing firmware code. You can use this file to “downgrade” your firmware level, so you can use the adapter with your earlier FLEX-ES release.

Q: Should I leave my FLEX-ES dongle connected to my USB port when I am not running FLEX-ES?

A: If it is convenient, leave it connected. A dongle contains a battery that is good for many years, whether the dongle is connected or not. However, if the dongle is connected to a system (with power on the system) then there is no load on the internal battery.

Q: Must I connect the FLEX-ES dongle before powering up my system?

A: In general, the answer is no. You can hot plug the dongle into the system before starting the FLEX-ES resource manager. However, FLEX-ES releases 6.2.18 and 7.0.1 do not accept hot plugging the dongle. A bypass is available if hot plugging is required. FLEX-ES release 7.0.2 accepts hot plugging.

Q: Can I install my channel adapters in any PCI slot?

A: Yes, but you should do a little planning. If your system has multiple PCI busses available, separate slow adapters (such as the 33 MHz serial channel adapter) from fast adapters (such as the ServeRaid adapters). If your system has only one PCI bus available, you have no choice.

Q: Do I need a different FLEX-ES license to use release 7.0.1 or 7.0.2? To use a channel adapter?

A: Your FLEX-ES license must be at least at the level of the FLEX-ES release you use. To use release 7.0.2, your license must be at 7.0.2 or a later level. To use channel adapters, your FLEX-ES license must be enabled for channel adapters.

Q: Can I use FSI channel adapters with FLEX-ES releases earlier than 7.0.1?

A: Not with Linux-based FLEX-ES systems. (FLEX-ES systems based on UnixWare can use parallel channel adapters with earlier FLEX-ES releases.)

Q: Does the same FLEX-ES license work for both Linux and UnixWare?

A: No.

Q: Must I use z/OS release 1.6 with FLEX-ES release 6.2.18 or 7.0.2?

A: No. There is no defined relationship between a z/OS level and a FLEX-ES release. However, if any program in your z/OS system (such as some functions within DB2 Version 8) uses long-displacement instructions then you should use FLEX-ES 6.2.17 or later.

Q: How can I determine my FLEX-ES license level and expiration date?

A: Use the command `d release` while at the `flexescli>` prompt.

Q: Must I use the version of X3270 that is distributed with FLEX-ES?

A: For a remote user (connected through the FLEX-ES Terminal Solicitor), the answer is no. For local X3270 sessions that are started automatically by the shell script that is used to start a S/390 instance, the answer is more complicated. You need an X3270 program that can accept logical terminal names from the calling parameters, such as L701 in the following example:

Example 0-1

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

The FSI-provided X3270 program can do this. We were told that the latest “official” X3270 program (available from various Web sites) can also accept logical terminal names from the calling parameters (using a different syntax for the calling parameters). Verify that any X3270 program you download has this capability, and if it will be started with a command such as the one shown in Example 0-1.

Q: Will 3390s connected through ESCON channels be faster than emulated 3390 drives?

A: We found the opposite: the emulated drives seem to be faster. However, this is partly a matter of cache effectiveness and different workloads can produce different results.

Q: Do I need a full FLEX-ES license and dongle to do only control unit emulation?

A: Yes.

Q: Do I need a special FLEX-ES license to do control unit emulation?

A: To configure the channel adapter as a CU, your FLEX-ES license must be enabled for this.

Q: Can the same SCA be used to access external control units (with an operating system running under FLEX-ES) and to provide external host access to FLEX-ES emulated control units? That is, can a channel perform both ways at the same time?

A: No. A FLEX-ES channel adapter cannot provide access to external control units and external access to FLEX-ES emulated control units at the same time. Two channels are needed.

Q: Can ESCON adapters from other vendors be used with FLEX-ES? Can other IBM ESCON adapters be used? Can the P/390 ESCON adapters be used?

A: No.

Q: Can PAV be used with the ESCON channels?

A: We did not try PAV. We assume it can be used when z/OS is running on the FLEX-ES

system and is connected to a properly configured control unit that supports PAV. PAV is not supported by FLEX-ES emulated control units.

Q: Can a serial adapter card be used in a ThinkPad® docking station?

A: No. (We tried this. The problem appeared to be power management for the adapter card in the docking station. We did not spend much time investigating the problem.)

Q: Should I install the latest FLEX-ES release immediately?

A: Possibly. If you need the long-displacement instructions or if you purchased channel adapters then upgrade now. If you extensively use SCSI tape drives, you might consider upgrading soon. If you have a stable system and do not fall into one of these categories then there is no urgent reason to upgrade to the new releases.

Q: I need to copy a new FLEX-ES license key from a diskette to my notebook but my notebook has a USB-connected diskette drive. How do I do this with RedHat 9.0?

A: You can mount the diskette drive with the following command:

```
# mount -t msdos /dev/sda /mnt/floppy
```

You can then access diskette files through /mnt/floppy/xxx path names. However the msdos mount does not handle long file names. A better way is the following:

```
# cd /etc
# gedit mtools.conf          Change the configuration for the diskette tools
                             (change the following line as indicated:)
                             drive a: file="/dev/sda" exclusive mformat_only
                             (save the changed file)
                             (Attach the USB diskette drive and insert a diskette)
# mdir a:                    Check the diskette directory
# mcopy a:ibminternalxxx.6.2.18.key /var/adm/flexes/.flexeslicense
                             (Using the correct file name from your diskette, of course.)
```

Q: How can I copy a whole 3390 volume from an emulated drive to a "real" drive?

A: One way is to use the z/OS ADRDSSU utility program; for example:

```
//OGDEN1 JOB 1,OGDEN,MSGCLASS=X
// EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//IN DD UNIT=3390,DISP=SHR,VOL=SER=111111
//OUT DD UNIT=3390,DISP=SHR,VOL=SER=222222
//SYSIN DD *
        COPY INDD(IN) OUTDD(OUT) ALLDATA(*) ALLEXCP COPYVOLID ADMINISTRATOR
/*
```

This will copy volume 111111 to volume 222222 and change the volser of the target volume to 111111. After the copy is complete the target volume is placed offline to prevent a duplicate volser on the system. The two volumes may be emulated volumes, "real" volumes, or one of each.

Q: Can I copy an IPL volume?

A: Yes; use the parameters shown in the previous example.

Q: How do I reset *all* the options for a FakeTape device? [This question does not apply to ESCON-attached devices.]

A: You can use a CLI **mount** command that includes a null string for the options. For example:

```
flexes> d devstate 181
Filename: /tmp/nothing State: OPEN, READY, READ-WRITE, BEGINNING-OF-TAPE
Options: maxwritesize=100,scsitimeout=420,createmode=664

flexes> mount 181 /tmp/nothing ''          <==== note the null string
```

```
flexes> d devstate 181
Filename: /tmp/nothing State: OPEN, READY, READ-WRITE, BEGINNING-OF-TAPE
Options: scsitimeout=420,createmode=664
```

Remember that these parameters apply to FakeTape emulated devices on disk *and also to SCSI tape drives*. In particular, the maxwritesize parameter limits the amount of data written to a SCSI tape devices as well as limit the size of an emulated tape (FakeTape) on disk.

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 39. Note that some of the documents referenced here may be available in softcopy only.

- ▶ *EFS Systems on a Linux Base: Getting Started*, SG24-7007-01
- ▶ *EFS Systems on a Linux Base: Additional Topics*, SG24-7008
- ▶ *z/VM on an EFS Base: Getting Started*, SG24-7009

Other publications

These Fundamental Software, Inc., publications are also relevant as further information sources:

- ▶ *FLEX-ES System Programmer's Guide*, FSIMM300
- ▶ *FLEX-ES Resource Language Reference*, FSIMM310

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

Archived

Index

A

AD CD-ROM package 22
adapter cable 3
ADDRSSU utility program 37
ASN-and-LX Reuse Facility 33

B

blockmux 7
bus and tag connectors 7

C

CBY 4
channel types 4
channel-attached control unit(s) 25
chanpath definition 26
chpbt rpm 11
ckddump program 34
CNC 4
configuration examples 18
control unit behavior 25
Control unit path numbers 28
Control Units, multiple 29
CTC 4
cu definition 16
cu definitions 28
CU emulation 30
cu interface statement 26
CUADD 3
cuadd parameter 31
CVC 4

D

devad address 16
devad count 16
device number 4, 16
device-level frames 3
diskette drive 37
dongle, FLEX-ES 35

E

EMIF configuration 28
emulated FLEX-ES channels 2
eprog program 12
ESA mode 16
ESCON connector 3
ESCON Director 3, 16
ESCON Director port 28, 30
ESCON quick review 3

F

fiber optic cable 3

firmware 35
firmware level 12
FLEX-ES definitions 15
FLEX-ES dongle 35
FLEX-ES license 35
FLEX-ES license key 37
FLEX-ES releases 33
flexes rpm 11
Frames, ESCON 3
fsiica rpm 11
ftlib rpm 11
Fundamental Software, Incorporated 1

G

GRS connection 32

H

HCD, for new IODF 23
host-connected control unit emulation 25

I

ICA 1
instances, multiple FLEX-ES 19
Integrated Communications Adapter (ICA) 1
IOCDs 4, 26
IPL volume 37

L

LED, on SCA 10, 12
license key 37
license, FLEX-ES 35
Link Address 4
link parameter 16
link to a control unit 3
link-level frames 3
Linux 2
Linux messages 13
local interface 27
Logical control unit paths 23
Long-Displacement Facility 33

M

memmgr rpm 11
MIF configuration 28
msgmgr rpm 11
MTRJ 6
MTRJ connector 3
multiple paths 19

N

network interface 27

P

Parallel Channel Adapter 7
parallel channel adapter 35
Parallel Channel Adapter (PCA) 1
parallel channel characteristics 7
path definition 29
path parameter 26–27
path parameters 31
PAV, with FLEX-ES 36
PCA 1, 27
PCI slot 35
physical channels 2
physical path to devices 23
point-to-point link 3
port address 4
ports, Director 3
Principles of Operation manual 33

R

raw device definitions 26
Redbooks Web site 39
 Contact us viii
rpm Installation 11

S

SC connector 3
SCA 1, 6, 27
SCSI tape drives 33
Serial Channel Adapter (SCA) 1
Serial Channel Adapters 6
ServeRaid adapter 6
Shared DASD 31
software address 4
strain relief device 9

T

ThinkPad docking station 37

U

UNITADD 3
unitadd parameter 16
UnixWare 2

X

X3270, versions 36

Z

z/Architecture ASN-and-LX Reuse Facility 33
z/Architecture Long-Displacement Facility 33
z/OS commands 23



EFS Systems on a Linux Base: Channel Adapters



Redbooks

Functional description

FLEX-ES definitions

Control Unit emulation

This IBM Redbook briefly describes channel adapters and functions available for use with FLEX-ES. These channel adapters connect FLEX-ES systems through parallel channels and serial (ESCON) channels to typical S/390 I/O devices, including disk drives. New FLEX-ES support, starting with release 7.0, provides more general channel support than was available in earlier releases and includes usage of new serial channel adapters.

This document is primarily directed to participants in the IBM internal ITSO/EFS project and to IBM PartnerWorld for Developers members with relevant interests. It is not intended as an introduction to FLEX-ES and readers are assumed to be generally familiar with FLEX-ES definitions and operations as well as general S/390 terminology.

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

SG24-6498-00

ISBN 0738492442