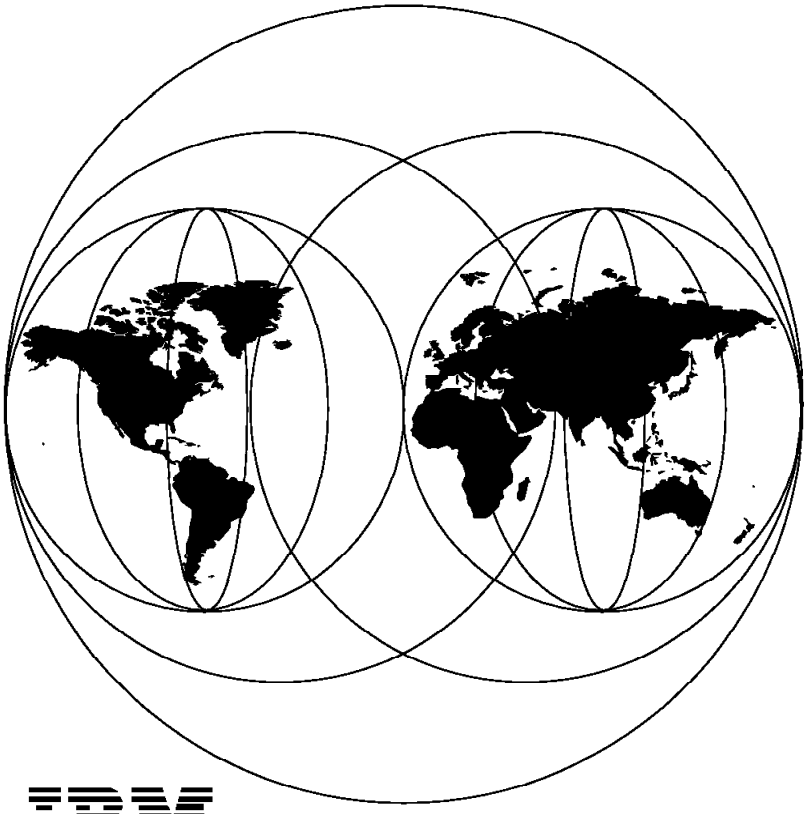
International Technical Support Organization

SG24-4589-00

RISC/6000 to Mainframe
Using S/370 Channel Connections

January 1996



IBM

International Technical Support Organization
Austin Center

International Technical Support Organization

SG24-4589-00

RISC/6000 to Mainframe
Using S/370 Channel Connections

January 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information under “Special Notices” on page xiii.

First Edition (January 1996)

This edition applies to Block Multiplexer and to Enterprise System Connection adapter support, which are part of AIX Version 3.2.5.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

An ITSO Technical Bulletin Evaluation Form for reader's feedback appears facing Chapter 1. If the form has been removed, comments may be addressed to:

IBM Corporation, International Technical Support Organization
Dept. 632B Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

Abstract

This document provides examples for connecting RS/6000 systems to Virtual Machine (VM) and Multiple Virtual Storage (MVS) systems via Block Multiplexer and Enterprise Systems Connection (ESCON) channels. Examples are given using TCP/IP, SNA, and Client Input Output/Sockets (CLIO/S).

This document was written for customers and system engineers who plan to connect RS/6000 systems to mainframes that use channel architectures. Some knowledge of AIX, VM and MVS is assumed.

(199 pages)

Contents

Figures

Tables

Special Notices

This publication is intended to help customers and system engineers install and configure AIX Channel Connection Products. The information in this publication is not intended as the specification of any programming interfaces that are provided by AIX, VM or VMS. See the PUBLICATIONS section of the IBM Programming Announcement for AIX, VM and MVS for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (VENDOR) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

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

ACF/VTAM
Advanced Function Printing
AIX/6000
APPN
ADSTAR
AIX
AIXwindows
AT

C/370
CICS/MVS
CICS/6000
DATABASE 2
DB2/6000
ESA/390
ESCON
IMS
MVS/ESA
MVS/XA
PowerPC
POWERserver
Print Services Facility
RISC System/6000
S/370
SAA
SP
SP2
System/390
VM/XA
CICS
CICS/VM
CUA
DB2
ESA/370
ESCON XDF
IBM
Micro Channel
MVS/SP
NetView
PowerPC Architecture
POWERstation
RACF
RS/6000
S/390
Scalable POWERparallel Systems
SP1
SQL/DS
VM/ESA

The following terms are trademarks of other companies:

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Windows is a trademark of Microsoft Corporation.

NFS	Sun Microsystems, Inc.
SCSI	Security Control Systems, Inc.

Other trademarks are trademarks of their respective companies.

Preface

This document is intended to provide a set of examples for AIX and mainframe users to assist in the connection of RISC/6000 systems to mainframes via Block Multiplexer and Enterprise System Connection (ESCON) channels.

This document is intended for customers and system engineers who intend to use RS/6000 channel connections to mainframes.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Channel Connections”

  This chapter contains a description of the book′s contents, a summary of channel products and the required software needed to drive them.

- Chapter 2, “MVS TCP/IP V3R1M0 Overview”

  This chapter is an introduction to MVS and mainframe TCP/IP terminology and function. While not being exhaustive, it provides a sufficient understanding of TCP/IP connectivity, what is needed to achieve it and what to ask of the MVS/VM system programmers prior to starting the connectivity.

- Chapter 3, “RISC System/6000 S/390 ESCON Channel Connection”

  This chapter describes the steps necessary for ESCON connectivity to both VM and MVS systems via the RISC System/6000.

- Chapter 4, “ RS/6000 S/370 Block Multiplexer Channel Connection”

  This chapter describes the steps necessary for BLKMUX connectivity to both VM and MVS systems via the RISC System/6000.

- Chapter 5, “SNA Channel Connectivity Using AIX SNA Server/6000”

  This chapter describes the steps necessary for SNA connectivity to both VM and MVS systems over BLKMUX and ESCON via RISC System/6000. SNA and TCP/IP are achieved over the same channel.

- Chapter 6, “ IBM Client Input Output/Sockets (CLIO)”

  This chapter describes the CLIO/S product and the steps necessary to set up connectivity to MVS via BLKMUX and ESCON via the RISC System/6000. We comment on bandwidth achieved and the rate-determining factors in defining the achievable speed across the complete data pipe.

- Chapter 7, “VM Host Connection”

  This chapter describes the step taken to connect AIX systems to VM hosts.

- Chapter 8, “Throughput Figures and Tuning”

  This chapter quotes some performance figures that have been achieved in the labs with the various products. It should be used as a reference only and with a careful understanding of what the customer can possibly achieve.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document.

- *AIX SNA Server/6000 Channel Connectivity Feature User's Guide*, SC31-8081
- *Enterprise System Connection Adapter User's guide and Service Information*, SC23-2474-01
- *Block Multiplixer Channel Adatper User's Guide and Service Information*, SC23-2427-02
- *Enterprose Systems Architecture/390 ESCON I/O Interface*, SA22-7202-02
- *AIX Version 3.2 System Management Guide: Communications and Networks*, GC23-2487

International Technical Support Organization Publications

- *TCP/IP Tutorial and Technical Overview*, GG24-3376-04
- *IBM TCP/IP V3R1 For MVS Implementation Guide*, GG24-3687-02
- *IBM TCP/IP Version 2 Release 2 For VM Installation and Interoperability:ecit, GG24-3624-02*

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, *GG24-3070.*

To get a catalog of ITSO redbooks, VNET users may type:

```
TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
```

A listing of all redbooks, sorted by category, may also be found on MKTTOOLS as ITSOCAT TXT. This package is updated monthly.

How to Order ITSO Redbooks

IBM employees in the USA may order ITSO books and CD-ROMs using PUBORDER. Customers in the USA may order by calling 1-800-879-2755 or by faxing 1-800-445-9269. Almost all major credit cards are accepted. Outside the USA, customers should contact their local IBM office. Guidance may be obtained by sending a PROFS note to BOOKSHOP at DKIBMVM1 or E-mail to bookshop@dk.ibm.com.

Customers may order hardcopy ITSO books individually or in customized sets, called BOFs, which relate to specific functions of interest. IBM employees and customers may also order ITSO books in online format on CD-ROM collections, which contain redbooks on a variety of products.

ITSO Redbooks on the World Wide Web (WWW)

Internet users may find information about redbooks on the ITSO World Wide Web home page. To access the ITSO Web pages, point your Web browser to the following URL:

`http://www.redbooks.ibm.com/redbooks`

Subscribing to Internet Listserver

IBM redbook titles/abstracts are now available through Internet E-mail via the IBM Announcement Listserver. With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. All it takes is a few minutes to set up a profile, and you can get news (in ASCII format) from selected categories.

To initiate the service, send an E-mail note to:

`announce@webster.ibmlink.ibm.com`

with the keyword `subscribe` *in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.*

To obtain more details about this service, employees may type the following:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
```

Note: *INEWS users can select RelInfo from the action bar to execute this command automatically.*

IBM employees may access LIST3820s of redbooks as well. The internal Redbooks home page may be found at the following URL:

`http://w3.itsc.pok.ibm.com/redbooks/redbooks.html`

Acknowledgments

This project was designed and managed by:

Al Mitchell
International Technical Support Organization, Austin Center

The authors of this document are:

Dr. Martin Cotter
IBM UK

Hyuk Kahng
IBM Korea

Toshio Asami
IBM Japan

This publication is the result of a residency conducted at the International Technical Support Organization, Austin Center.

Thanks to the following people for the invaluable advice and guidance provided in the production of this document:

Barry Nusbaum
International Technical Support Organization, Raleigh Center

Rich Avery
RISC System/6000 Division
IBM US

Bill Flint
RISC System/6000 Division
IBM US

Bernie Slovenski
RISC System/6000 Division
IBM US

Joe Casey
RISC System/6000 Division
IBM US

Jim Wenzlaff
RISC System/6000 Division
IBM US

Bret Bidwell
RISC System/6000 Division
IBM US

Bill Richardson
RISC System/6000 Division
IBM US

Marcus Brewer
Editor, ITSO Austin Center

Chapter 1. Channel Connections

Connecting RS/6000s to mainframes via 370 channel architectures can be a challenging task. It is rare to find an individual with mainframe O/S (VM and MVS), RS/6000, TCP/IP, and SNA skills. This book is an attempt to document these connections which were accomplished in the laboratory during a residency at ITSO Austin.

There has been a remarkable worldwide growth in the use of RS/6000 technology. This growth is coupled with a trend in which many customers want to retain their base mainframe technology. This evolution demands that the networking between these systems must match the enterprise needs. It must appear seamless to the applications running between the two architectures and have the bandwidth to cope with the transfer of huge amounts of data stored on mainframe Direct Access Storage Devices (DASD) and tape.

The window allowed for moving this data can be very short due to pressures involved with running day-to-day business operations. The only way it can be achieved dynamically is by using channel products.

There is also the need to utilize channel bandwidth to back up and restore data from RISC/6000s to mainframe-type devices, such as DASD and large, automatic tape drives, because the customers typically desire to absorb UNIX back-ups into their present storage-management hierarchy.

1.1 Our Experience

Our objective is to provide an understanding of how to link the two differing architectures while trying to give the mainframe or AIX specialists a physical base on which to install their applications when the connectivity is adequately set up. In doing so, we have taken a few shortcuts in order to keep this documentation readable in size. In this respect, we have tried to go out of our way to facilitate a simple flow of exactly what to do and where to get the information required when contemplating the connection.

That said, having connected various systems an innumerable number of times in the Austin laboratory, it becomes relatively facile if, as part of your team, you have an AIX systems specialist and a mainframe systems programmer.

1.2 Channel Products Overview

These components are available for connecting RS/6000s to mainframes.

1.2.1 Block Multiplexer Channel Adapter

The BLKMUX channel adapter (or Parallel Channel Adapter (PCA)) channel attaches to an ES/9000, 3090, 308X, or 4381 mainframe system. It effectively provides the functions required for the RS/6000 to be a control unit to the host.

1.2.2 ESCON Control Unit Adapter

The Enterprise System Connection (ESCON) Control Unit Adapter provides functions similar to the BLKMUX adapter, but attaches to an ESCON channel of an ES/9000 or ESCON-capable 3090 mainframe system.

1.2.3 S/370 Channel Emulator/A Adapter

The S/370 Channel Emulator/A Adapter provides the facility to attach selected Mainframe-type I/O devices to an IBM RISC System/6000.

1.2.4 S/390 ESCON Channel Emulator

The S/390 ESCON Channel Emulator provides channel functions similar to the S/370 Channel Emulator, but uses ESCON channel technology. ESCON-capable tape control units and tape libraries are attachable.

1.3 Hardware and Software Requirements for Channel Products

The following section describes the hardware and software requirements for installation of RISC System/6000 channel products.

1.3.1 Hardware Requirements

Block Multiplexer Channel Adapter (FC 2755)

- FC 2757 - Adapter cable
- FC 2758 - Interface assembly
- Channel bus and tag cables (cable group 0185)

S/370 Channel Emulator/A Adapter (FC 2759)

- The adapter cable, included in FC 2759
- Channel bus and tag cables

ESCON Control Unit Adapter (FC 2756)

- FC 6506 - Fan for RISC System/6000 model 7013
- ESCON fiber optic cabling/connectors

S/390 ESCON Channel Emulator (FC 2754)

- FC 6506 - Fan for RISC System/6000 model 7013
- ESCON fiber optic cabling/connectors

1.3.2 Software Requirements

Block Multiplexer Channel Adapter (FC 2755)

- AIX/6000 Version 3.2 or above

  Unless specified all the work involved is on AIX/6000 3.2.5
- FC 5055 - TCP/IP Only Device Driver
- FC 4760 - TCP/IP and SNA Device Driver
- VM TCP/IP 2.2 or later (for TCP/IP with VM)
- MVS TCP/IP 2.2 or later (for TCP/IP with MVS)

S/370 Channel Emulator/A Adapter (FC 2759) For Printer Support

- AIX/6000 Version 3.2.2 AND PTF U405589 or above
- 5765-140 IBM Print Services Facility/6000

S/370 Channel Emulator/A Adapter (FC 2759) for tape support

- AIX/6000 Version 3.2.2 AND PTF′s U405589, U403146 or above
- 5799-QDA - IBM Parallel Channel Tape Attachment/6000

ESCON Control Unit Adapter (FC 2756)

- AIX/6000 Version 3.2.3 or above
- Feature 5056 - extended support
- Feature 4761 - (For SNA driver as well)
- VM/ESA or MVS/ESA on the mainframe
- VM TCP/IP 2.2 or later (for TCP/IP with VM)
- MVS TCP/IP 2.21 or later (for TCP/IP with MVS)

S/390 ESCON Channel Emulator (FC 2754)

- AIX/6000 Version 3.2.3 or above
- The following are included with the adapter:
  - Run-time diagnostics
  - S390 (Low-Level) device driver
  - MTDD (High-Level) device driver

1.4 The RISC System/6000 Hardware and Software Configuration Test

The following section describes the channel connectivity and software products associated with the test system.

1.4.1 Hardware

Our network is reasonably complex because numerous RS6Ks are involved and attached to two separate mainframes. Please refer to Figure 1 on page 4 for an overall view of the hardware connectivity. All the RS6K channel product connectivity is shown in this diagram. The RS/6000 was a 990 with 128 MB of memory.

The host had two BLKMUX and two ESCON adapters for connectivity to both VM and MVS separately and FDDI connectivity to a separate RS6K.

These systems and networks were frequently reconfigured and renamed to be used for different types of connectivity. A summary of the attributes for the 990 is shown in A.9, “LSCFG of Test RS/6000.” on page 161. This is output from the command: `> lscfg`

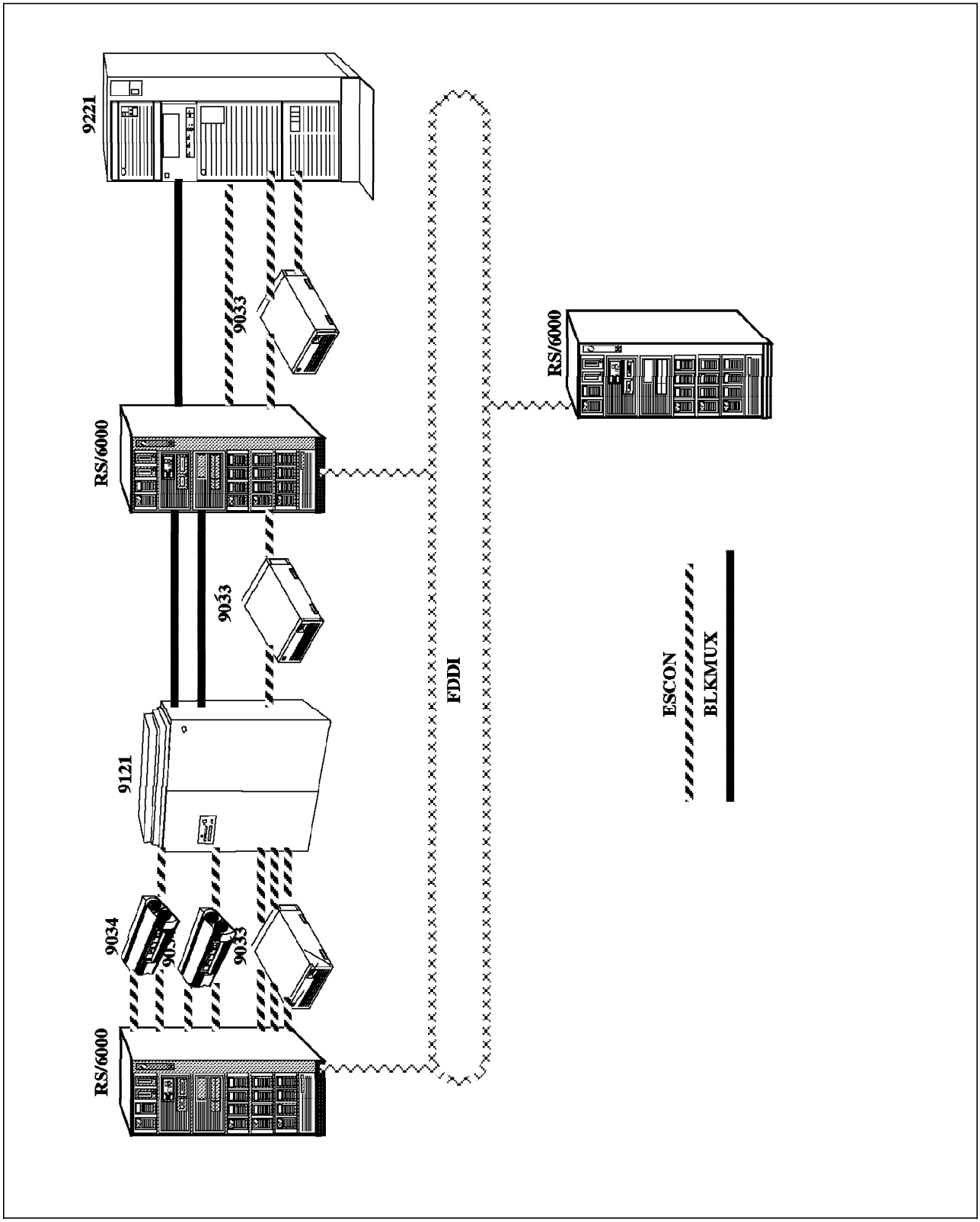


Figure 1. Our network

Chapter 2. MVS TCP/IP V3R1M0 Overview

This overview provides a brief understanding of what is required for the implementation of TCP/IP connectivity to the MVS host. This enables the reader to determine what is required for their particular system and what information to request from MVS systems programmers. For a more detailed discussion of MVS TC/PIP installation and tuning on the mainframe, please refer to IBM TCP/IP V3R1 For MVS Implementation Guide.

2.1.1 Summary of the Austin Setup

As a summary and a reminder, the overall attachment in the Austin setup consists of:

1. IBM ES/9000 9121-320 Mainframe System
2. 9121-320 24 MIPS with 256 MB main memory
3. IBM MVS/ESA Version 4.3 Operating System
4. IBM TCP/IP Version 3.1.0 for MVS
5. IBM ESCON Director 9033
6. IBM RISC System/6000 Model 990
7. IBM AIX Version 3.2.5 for RISC System/6000

2.1.2 TCP/IP CTC and CNC Implementation in MVS

In MVS, all TCP/IP servers are located in separate address spaces, except the Telnet server which is imbedded to the TCP/IP System Address Space. For a direct connection, the channel type identifies the Block Multiplexer adapter or ESCON adapter. One can specify an ESCON Converter (for the Block Multiplexer adapter) or an ESCON Director (for the ESCON adapter). The channel-type definition for AIX can be one of three types:

- BL For Block Multiplexer direct connection
- CVC ESCON Converter (intermediate)
- CNC ESCON through an ESCON Director or direct attach

Channel to channel (CTC) directly attached networks link to the TCP/IP System Address Space via the TCP/IP drivers. The definition of the addresses inside the TCP/IP main datasets is the major task in the implementation of TCP/IP connectivity. This is shown in Figure 2 on page 6.

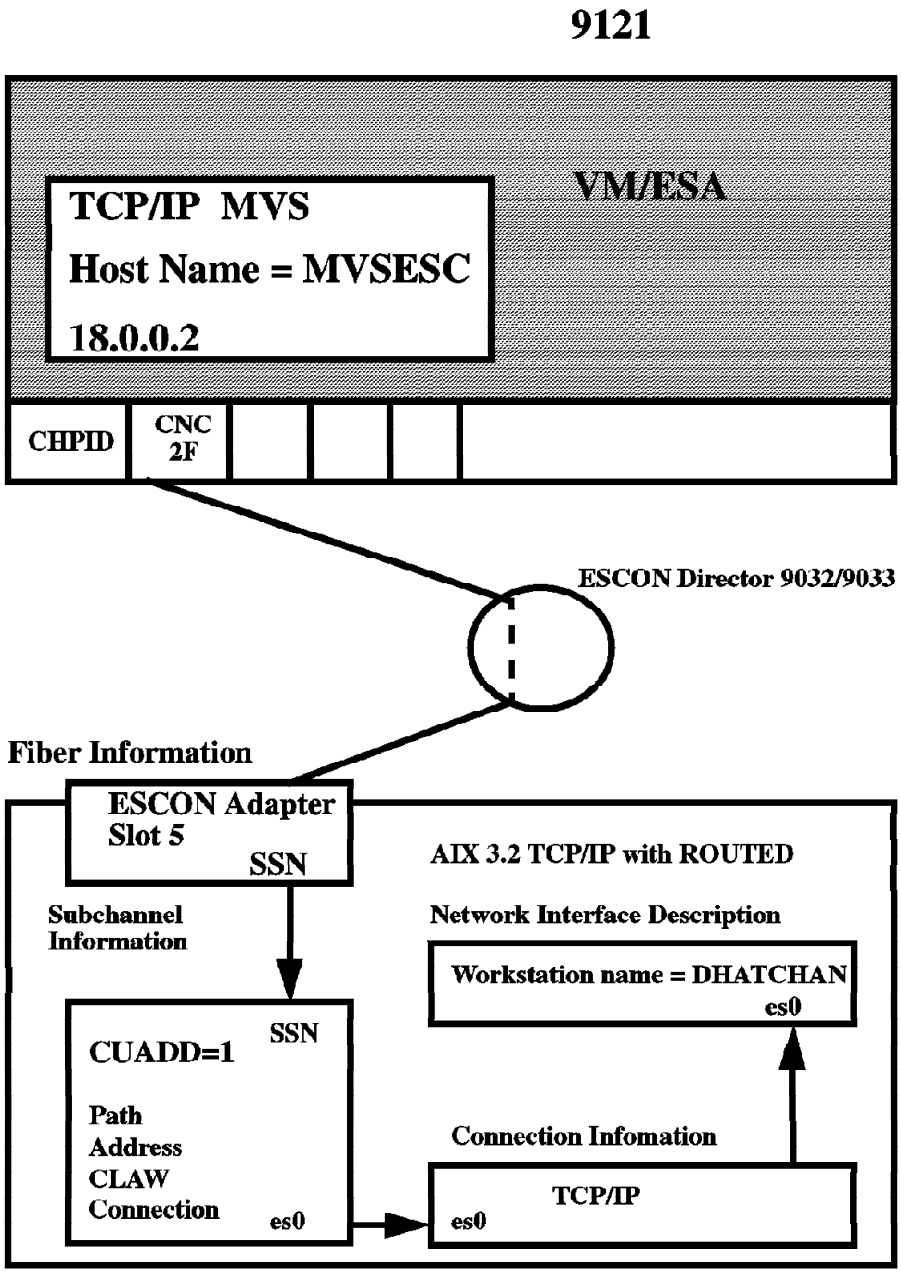


Figure 2. CTC Definitions for TCP/IP on MVS

2.1.3 HCD Definition

There are two methods for the 9121/9221 mainframes to define their associated I/O devices at the hardware level. The definitions are done with the I/O Configuration Program (IOCP) or Hardware Configuration Definition (HCD) program. The latter is dynamically configurable and does not necessitate a complete reboot of the system, whereas the IOCP requires a power-on reset. Since the HCD method was used for the Austin test suite, the RISC system must be set up as DYNAMIC=NO. If it is set up as a dynamic device, TCP/IP returns a condition code 3 message. Figure 3 is the HCD gen for most of the subchannel work we have used.

```
                                  CHANNEL PATH DETAIL REPORT                    TIME: 15:47  DATE: 95-08-04  PAGE E-   8
PROCESSOR ID FG9121   TYPE 9121     MODEL 320      CONFIGURATION MODE: LPAR  TOKEN: FG9121   95-08-04 14:58:44 IODF     IODF80
                           --- SWITCH ----                                    UNIT ADDR                 UNIT
                           ID  PR  CU  DYN   --- CONTROL UNIT ---  CU-          RANGE    -- DEVICE --  ADDR   DEVICE
CHPID  TYPE    SIDE  MODE      PN  PN  ID   NUMBER  TYPE-MODEL    ADD  PROTOCOL  FROM  TO   NUMBER,RANGE  START  TYPE-MODEL
-----  ------  ----  ----  ----------  ---  ------  ------------  ---  --------  ----  ---  ------------  -----  -------------
 0C    BL            DED                 00C1  3088                   S4        40   4F    0C40,16       40    3088
                                         00C2  RS6K                   S4        50   5F    0C50,16       50    RS6K
                           ------------------------------------------------------------------------------------------------
                                         0C60  3174                   S         60   6F    0C60,16       60    3278-2
                           ------------------------------------------------------------------------------------------------
                                         0C70  3174                   S         70   77    0C70,8        70    3174
                           ------------------------------------------------------------------------------------------------
                                         0C71  3791L                  S         78   7F    0C78,8        78    3791L
------------------------------------------------------------------------------------------------------------------------------
  2F    CNC           REC                 0184  SCTC          1              10   11    0910,2        10    SCTC
                           ------------------------------------------------------------------------------------------------
                                         0185  SCTC          2              12   13    0912,2        12    SCTC
                           ------------------------------------------------------------------------------------------------
                                         0186  SCTC          3              14   15    0914.2        14    SCTC
                           ------------------------------------------------------------------------------------------------
                                         0187  3174          4   S          16   16    0916          16    3174
                           ------------------------------------------------------------------------------------------------
                                         0188  SCTC          5              18   1F    0918,8        18    SCTC

ESOTERIC NAMES

-                                                 AFFINITY  ALLOCATION   ASSOCIATED
  NAME        NAME TYPE  VIO  TOKEN  PREF         INDEX    DEVICE TYPE  GENERICS                DEVICE NUMBER LIST
  ---------   ---------  ---  -----  ----  --------  -----------  ----------  ------------------------------------------------
  ------------------------------------------------------------------------------------------------------------------------------
  RS6K17      ESOTERIC             8300    FFFF      00004101     SCTC        0912 0913 0914 0915
  ------------------------------------------------------------------------------------------------------------------------------
  RS6K43      ESOTERIC             8389    FFFF      00004103     RS6K        0C50 0C51 0C52 0C53 0C54 0C55 0C56 0C57 0C58 0C59 0C5A
                                                                              0C5B 0C5C 0C5D 0C5E 0C5F
```

Figure 3. HCD gen listing

2.1.4 TCP/IP Base Datasets and Functions

The TCP/IP software is very flexible and relatively easy to install. At start up time, the configuration parameters for TCP/IP and all its services (server and client processes) are read from configuration datasets. All of these are supplied by IBM as a sample, but some of them **must** be modified in order to reflect the customer environment.

Once installed, one can customize the basic functions of TCP/IP by copying and editing these key datasets. They are:

- PROFILE.TCPIP
- TCPIP.PROCLIB
- TCPIP.DATA

- HOSTS.LOCAL
- FTP.DATA

Each mainframe site may have its own specific nomenclature for executing and changing TCP/IP function and the Austin test site changed these default names similarly. Table 1 shows the name changes and the purposes of these key datasets before we go on to explain the definitions more carefully:

Table 1. Configuration Files for TCP/IP on MVS. Austin Test Configuration Files for TCP/IP on MVS.

File Name	File name for main TCP/IP system on MVS	TCP/IP function	Purpose
PROFILE.TCPIP	SYS1.TCPPARMS(AUSHATA)	TCP/IP and Telnet	TCP/IP address space Configuration
TCPIP.PROCLIB	SYS1.PROCLIB(TCPIP)	Starting TCP/IP	Starting TCP/IP
TCPIP.DATA	SYS1.TCPPARMS(TCPDATA)	All server and client address spaces	• Major system parameters • Client configuration
HOSTS.LOCAL	TCPIP.V3R1M0.HOSTS(LOCAL)	TCP/IP	Hostname to Internet translation
FTP.DATA	SYS1.TCPPARMS(FTPDATA)	FTP clients	FTP client default configuration

Please see the following sections

A.2, “SYS1.TCPPARMS.AUSHATA” on page 138
A.4, “TCPIP.V3R1M0.HOSTS.LOCAL” on page 144
A.1, “SYS1.PROCLIB.TCPIP” on page 137
A.3, “TCPPARMS.FTPDATA” on page 143

for a listing of these individual files.

2.1.4.1 PROFILE.TCPIP

Before you start TCP/IP with the TCPIP.PROCLIB, the TCP/IP address space configuration parameters are entered in the PROFILE.TCPIP. The default name for this file is PROFILE.TCPIP, but in the Austin test setup, this was called SYS1.TCPPARMS(AUSHATA).

This dataset is broken up into five linking statements:

- DEVICE
- LINK
- HOME
- GATEWAY
- START

The following configuration parameters needed to be updated specifically for our work:

- TCP/IP buffer pool definitions

```
DATABUFFERPOOLSIZE          300 32768
LARGEENVELOPEPOOLSIZE       300 32768
```

- TCP/IP reserved port number definitions

- Device Statements:

  The main definitions changed for the work with MVS were:

```
DEVICE RPERF CLAW 910 MVSESC DHATRISC NONE 26 26 4096 4096 ; dhatrisk es0
DEVICE CA3 CLAW C50 MVSBLK1 MVSBLK NONE 26 26 4096 4096 ; dhatchan cat0
```

  where the first definition is for the ESCON adapter, and the second is for the Block Multiplexer adapter.

The device syntax is:

- RPERF and CA3

  Device names represented in the link statement.

- CLAW

  CLAW mode must be used for MVS.

- 910 and C50

  The I/O device address is defined in the HCD gen. This must be an even number, and TCP/IP always uses two devices: the even number for read and the odd number for write. A simple check at the MVS console to see whether these channels are varied on or not is wise prior to final testing for connectivity.

- MVSESC and MVSBLK1

  Definition of these adapters for the host system. These must match the RS/6000 definitions in /etc/hosts as defined via SMIT TCP/IP.

- DHATRISC and MVSBLK

  Name of the RS/6000 as defined for each channel-adapter type and connectivity.

- 26

  Represents the number of read and write buffers for TCP/IP. TCP/IP must have time to process the received data and append to the buffer before the channel program terminates. There should be enough write buffers to allow the busy TCP/IP connection to reuse the buffers without the channel program terminating. These buffers use real memory; so the size chosen affects the system performance!

- 4096

  This is the number of read and write buffers in bytes. This is the number that is set after the system was tuned.

- Link Statement

The link syntax is:

```
LINK PRF IP 0 RPERF
LINK CAT3 IP 0 CA3
```

- PRF and CAT3

  Unique assignee names

- IP and 0

  IP and 0 are keywords and must be placed here.

- RPERF and CA3

  These are the corresponding device names from the device statement.

- **HOME syntax**

  The HOME syntax is the local Internet address for the MVS host.

```
18.0.0.2      PRF
21.0.0.2      CAT3
```

- PRF and CAT3

  The link names specified in the link statement.

- The GATEWAY syntax refers to the statement:

```
GATEWAY
Network  First hop   Driver  Packet size  Subnet mask   Subnet value

    21      =         CAT3    4096            0
    18      =         PRF     4096            0
```

  This depends, of course, on the class of the IP address at your site.

- 21 and 18

  The network numbers, 21 and 18, refer to the IP address.

- =

  The equal sign (=) indicates that messages are routed directly via the link name.

2.1.4.2 TCPIP.DATA

TCPIP.DATA was copied, updated and renamed SYS1.TCPPARMS (TCPDATA) and is included in Appendix A.

The TCPIP.DATA configuration is read during initialization of all TCP/IP server and client functions. It contains the TCP/IP parameters available to address spaces in the MVS system that use TCP/IP functions. These parameters usually include, among others, the MVS hostname and the TCP/IP address space. If, for instance, the hostname is not included in the TCPIP.DATA, then system defaults to the SYS1.PARMLIB library where the IEFSSN00 member contains the hostname of the MVS system. This is included in Appendix A, “TCPIP and CLIO Configuration Files” on page 137. As is, the test setup hostname defaulted to the IEFSSN00 member.

2.1.4.3 HOSTS.LOCAL

HOSTS.LOCAL is similar in function to **/etc/hosts** in AIX. This dataset contains the names and the Internet addresses of the hosts that are used most often.

For convenience at a customer site, it may be wise to use a central domain name server which is defined in TCPIP.DATA with NSinterAddr statements. If none of the latter are present (as was the case in our testing), then the system defaults to the HOSTS.LOCAL file.

The Austin test system is in a constant state of flux because it is used constantly to test new products. For a more stable environment, using the domain server approach and putting the hostname of the MVS machine associated with the appropriate channel type (it would be convenient to have different ones for ESCON or BLKMUX) into the TCPIP.DATA dataset would be preferable.

2.1.4.4 MAKESITE

In order to change or introduce more host names, execute a special TCP/IP command called `makesite` for the changes to become effective. The `makesite` command reads the HOSTS.LOCAL dataset and builds two others:

- HOSTS.SITEINFO
- HOSTS.ADDRINFO

These two files are used by IP for hostname translation. HOSTS.SITEINFO translates hostnames to IP addresses, and HOSTS.ADDRINFO translates IP addresses to hostnames. They are executable and cannot be edited directly.

`Makesite` is run from a TSO command or as a batch job, and the sequence we used at Austin was as follows:

1. Rename the HOSTS.SITEINFO and HOSTS.ADDRINFO in backup datasets
2. Update the datasetprefix.HOSTS.LOCAL

   Our datasetprefix was TCPIP.V3R1M0.
3. Run `makesite`

   The output is written to

```
userID.HOSTS.ADDRINFO
userID.HOSTS.SITEINFO
```

4. Test the correctness of the changes by running the `testsite` command.
5. Rename UserID.HOSTS.ADDRINFO and UserID.HOSTS.SITEINFO to datasetprefix.HOSTS.ADDRINFO and datasetprefix.HOSTS.SITEINFO, respectively. Backups have already been made of these files.

2.1.4.5 TCP/IP PROCLIB

The TCP/IP PROCLIB was copied, updated and renamed as the SYS1.PROCLIB(TCPIP) dataset to match the requirements of the Austin test site. The details can be seen in Appendix A, “TCPIP and CLIO Configuration Files” on page 137 There are two main lines of JCL :

```
000082 //PROFILE  DD DISP=SHR,DSN=SYS1.TCPPARMS(AUSHATA)

000094 //SYSTCPD  DD DISP=SHR,DSN=SYS1.TCPPARMS(TCPDATA)
```

Where AUSHATA is the profile definition and TCPDATA contains the TCP/IP parameters.

2.1.4.6 Stopping and Starting TCP/IP at the MVS Console

Once everything has been installed correctly, TCP/IP can be started and stopped by issuing the following commands at the MVS console:

```
>S TCPIP
```

To start TCP/IP under MVS.

```
>P TCPIP
```

To stop TCP/IP.

Follow this with a simple ping from the RS/6000 to test the connectivity.

2.1.4.7 Introduction to FTP with MVS

In MVS, the FTP client/user interface is a line-mode interface implemented as a TSO command, and the user is in dialog with the FTP client protocol interpreter. The latter establishes a connection with the FTP server host (the RS/6000 in this case) which listens on port 21, the default control port. The TCPIP.PROFILE dataset (in our case AUSHATA) has the default port numbers set up. The data transfer takes place over another control connection, the data connection, which is established during the initial command exchange between the client and the server. Therefore, each FTP session requires two TCP ports on each TCP/IP host.

We have included further discussion of this in the chapter on CLIO/S where, after the test program FCFIDUM is run, more detail of the the port allocation for data transfer is part of the standard output. The channel bandwidth and connectivity was tested by transferring large files (> 25 MB) up and down to the mainframes. In order to do so, it is essential that FTP is configured correctly.

2.1.4.8 Configuring FTP in MVS

The MVS FTP server default SITE parameters are specified in the FTP server configuration dataset. For the client, the LOCSITE parameters are specified in either an installation-wide dataset or in a specific user-configuration dataset.

If the client is the MVS host, the FTP parameters are searched for by using the following hierarchy:

1. //SYSFTPD DD Explicit allocation in the PROCLIB JCL.
2. userdID.FTP.DATA
3. SYS1.TCPPARMS(FTPDATA) As chosen for the Austin site.
4. datasetprefix.FTP.DATA

For specific user ID requirements, use of dataset userID.FTP.DATA would be preferred.

2.1.4.9 Common Link Access to Workstation (CLAW)

TCPIP connection between RS/6000 and mainframe needs the use of Common Link Access to Workstation (CLAW); so a brief understanding of what CLAW does is relevant to our discussion.

CLAW is a low-level protocol that provides connectivity for TCP/IP running on the S/370 and the RS/6000. Each TCP/IP address space on the S/370 requires two dedicated subchannels for RS/6000 connectivity, and these must be defined if the channel connectivity over TCP/IP is to work. One subchannel is used for writing (the odd address), the other for reading (the even address). The fact that the channels are dedicated allows writing and reading of the data directly into the address space, thereby minimizing any interference.

CLAWs main functions assume that as major prerequisites, the S/370 channels should remain 100 percent busy for as long as the data transfer between S/370 and the RS/6000 occurs and that no S/370 I/O interrupts should occur as long as the CLAW user ID is not in an idle state. This facilitates the speed of transfer and the ease with which an application (such as CLIO) can use CLAW.

Chapter 3. RISC System/6000 S/390 ESCON Channel Connection

The Enterprise System Connection attachment is a high speed method of connecting devices to S/390 channel.

3.1 The Enterprise System Connection (ESCON)

The Enterprise System Connection (ESCON) attachment allows a RISC System/6000 to communicate with one or more system/390 hosts that use the ESCON architecture.

3.1.1 Overview

Our work involved the ESCON Control Unit Adapter only, but the ESCON Channel Emulator is also a part of the product overview. The latter is for attachment to ESCON-capable tape control units and tape libraries. The hardware is exactly the same for both products, and the feature code 2756 uses two microchannel slots.

As described in this chapter, the adapter can be defined in either the CLAW or the 3088 mode, both of which provide an API for applications written to this interface. The main advantages of fibre channel architecture are the increase in bandwidth to 10 MB/S or 17 MB/S, depending on which type of channel is installed, and the ease of installation due to the physical size of the cable and the excellent installation procedures. For this work, we used the 10 MB/S devices.

There are two methods of attachment:

- Direct attachment from the RS/6000 to the mainframe
- Via an 9033 ESCON Director (ESCD)

These are shown in Figure 4 on page 16.

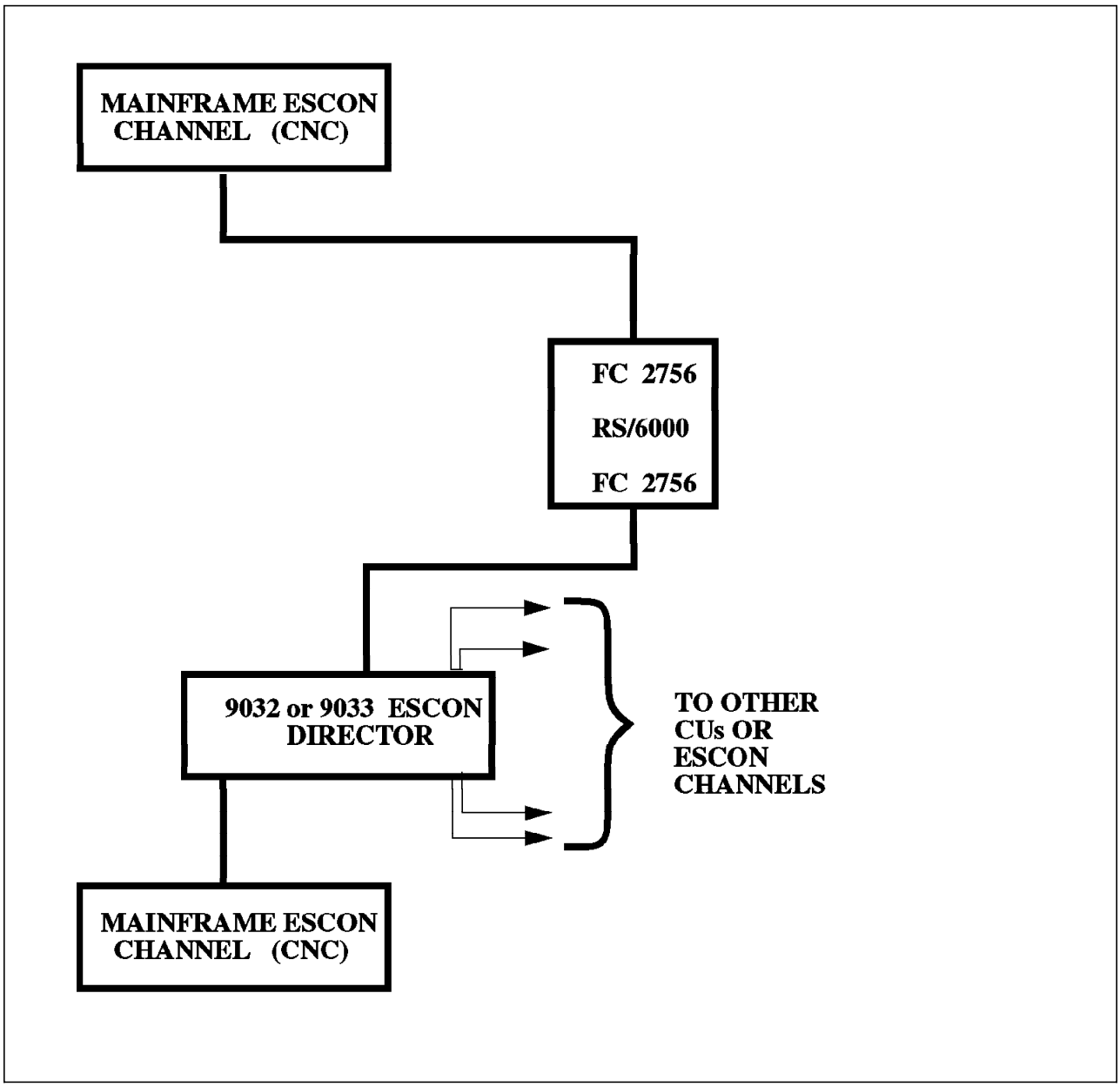


Figure 4. ESCON Configurations

Both configurations are included multiple times in the Austin test suite. The MVS link for this work was direct attach, and the VM link was via an ESCON Director (ESCD). There is little difference in the setup, except that particular attention must be taken for the subchannel definitions with respect to channel path through the ESCD.

When setting up the mainframe links, the major difficulties were in understanding the nomenclature between the two architectures and in ensuring that the definitions between the RS/6000 and the mainframe matched exactly. The architecture becomes more complex when ESCON Directors are being used, and the channel definitions have to be carefully thought through. We refer the reader to the particular definitions from the HCD gen on Figure 5 on page 21 that are

required for configuration and for information that is required prior to installing the hardware.

The channel attachment should be thought of as an attached processor rather than as a control unit. In this respect, the RS/6000 should be running an application that takes advantage of the ESCON bandwidth and architecture. One such application is CLIO, and we discuss this in detail in Chapter 6, " IBM Client Input Output/Sockets (CLIO)" on page 65.

The ESCON adapter supports links of up to three kilometers. When connected through two ESCDs and using the Extended Distance Facility (EXD) feature on the host and both directors, the link can be extended to up to 43 kilometers.

Section 3.1.2, "Planning Information Required" provides a default flow sequence that should be followed for the ESCON installation and testing. Most of the sections involve the use of the SMIT panels for set up and configuration, and we specifically refer to these in 3.1.3, "ESCON Channel Implementation" on page 19 for our own installation.

3.1.2 Planning Information Required

Prior to install, it is necessary to discuss the following items with the mainframe system programmers, and obtain the relevant information. For now, use this as a check list to determine who does which task. The individual terms are explained later in this chapter:

- Supported mainframes:
  - ES/9000-9021
  - ES/9000-9121
  - ES/9000-9221
  - 3090J
  - 9672 mainframe
- Software requirements:
  - AIX/6000 V3.2.3 or above
  - Feature 5056 (TCP/IP) or 4761 (TCP/IP and SNA) - device driver
  - VM/ESA or MVS/ESA on the mainframe
  - VM TCP/IP 2.2 or later (for TCP/IP with VM)
  - MVS TCP/IP 2.21 or later (for TCP/IP with MVS)
- Supported RS/6000 Systems
  - 7012 - Models 340 and higher, 1 adapter
  - 7013 - All models, 2 adapters
  - 7015 - All models, 2 adapters per I/O bus
  - 7030 - All models, 1 adapter
- Mainframe Channel ID (CHPID)
- Unit address and range
- I/O device addresses for the mainframe subchannels (and type)
- Link Address for the ESCON Director associated with the RS/6000

- ESCON Director ports where the host channel is plugged into
- Specific customer setup for TCP/IP files on MVS or VM
- If an application is used, the specific addresses
- RS/6000 and mainframe nodenames (Host name and workstation name)
- RS/6000 and mainframe IP addresses

3.1.2.1 Installation Sequence

The installation sequence is important in that the mainframe has to be IPL'd before the connectivity can be tested, and in the case of an IOCP gen system, a power-on reset is also required. The sequence should be followed:

- IOCDS Gen (IOCP or HCD) (HCD in the MVS mainframe example here)
- Mainframe operating system I/O gen
- Mainframe application (TCP/IP) configuration
- RS/6000 Adapter S/W install
- RS/6000 Adapter card install and cabling
- RS/6000 Adapter configuration via the SMIT interface (The fast path is via `smit esca`.)
  - Select **Devices**
  - Select **communications**
  - Select **ESCON Channel Adapter**
  - Configure and define the subchannels as follows:

3.1.2.2 Define the Subchannels

- For addresses, mode (3088/CLAW/3174-SNA)
- Select **Subchannel Definitions**
- Select **Add a Subchannel**
- Fill in the subchannel definition (see below for careful hints)
- Return to the ESCON Channel Adapter

3.1.2.3 Define Connections

- Select **Connection definitions**
- Select **Add a connection**
- Fill in as appropriate
- Return to the ESCON Channel Adapter

3.1.2.4 Define The Fibre Port

The definition of the fibre port ties in the subchannel addresses to the port with which they are associated.

- Select **Fibre definitions**
- Select **Add (or change if necessary) a Fibre**
- Fill in as appropriate
- Return to the ESCON Channel Adapter

3.1.3 ESCON Channel Implementation

The following sections outline in detail our specific setup for ESCON channels in the Austin test suite.

3.1.3.1 Installing and Configuring the ESCON Adapter

The ESCON adapters were configured in three phases:

1. Install the software.

   Installation of the 3088 emulation device driver

2. Shut the machine down, and install the ESCON adapter.
3. Reboot. The boot sequence should have already defined your ESCON interface. After the reboot, the ESCON director will be defined. It is not available until all definitions are complete.
4. Proceed to the definitions below.

3.1.3.2 ESCON Device Driver Installation

IBM supplies two drivers for the ESCON adapter for AIX 3.2.5. One is a combined SNA and IP driver, and the other is a TCP/IP only driver. If TCP/IP only is needed, then the TCP/IP driver should suffice. If SNA is needed, or will be needed in the future over the same channel concurrently, then the combined driver should be included in your system configuration. If your system is at AIX 4.1, you will get the combined driver by default.

- Feature code 4761 is the driver for SNA and TCP/IP. Order this feature code for both SNA and IP.
- Feature code 5056 is the TCP/IP-only driver. Order this driver if you want TCP/IP only.

The installation media should explain which driver you have; however, if the driver is already installed and you are just reconfiguring, a simple way of determining which driver is installed is to type the following commands at the RS/6000 command line:

```
dhatrisc:> smit blkmux
cd /etc/microcode
ls c38f.3174*
```

The combined driver has files of this type; the TCP/IP driver does not. If you have AIX 4.1, be aware that only the combined TCP/IP and SNA driver is supported.

3.1.3.3 Software Installation

The software installation for ESCON is from tape following the familiar SMIT install panels. At level 3.2.5, it is important to distinguish between the device drivers, as was discussed above.

3.1.3.4 Verification of Correct Install

To verify that you have the correct hardware and software, issue the following command to verify the ESCON level:

dhatrisk:># lslpp -h esc*.* The standard output for this should be:

```
Name
-----------------
Fix Id  Release         Status     Action     Date       Time   User Name

------ --------------- ---------- ---------- ---------- -------- --------

Path  /usr/lib/objrepos
escon.cuu
       03.02.0000.0000 COMPLETE   COMMIT     12/13/94   16:14:11 root

escon.usr
       03.02.0000.0000 COMPLETE   COMMIT     05/03/95   15:18:17 root
U 422416 03.02.0000.0000 COMPLETE   COMMIT     05/03/95   15:18:18 root
U 491144 03.02.0000.0000 COMPLETE   COMMIT     05/03/95   15:18:19 root


Path  /etc/objrepos
escondiag.obj
       03.02.0000.0000 COMPLETE   APPLY      03/22/95   10:58:09  root
```

Verify the microcode with the following commands:

```
> cd /etc/microcode
> ls c38f*.*
```

3.1.3.5 ESCON Device Driver Configuration

The MVS ESCON link was direct attached and did not pass through any ESCON Directors. For configuration of a ESCD, please refer to 3.1.5, “Setting Up the ESCD for VM TCP/IP over ESCON” on page 30.

The reader should try to become familiar with the contents of the HCD listing as it is not intuitive to the unfamiliar eye, and attention to detail is very important when setting up the subchannels.

Qualification of the Planning Information

Figure 5 on page 21 is a printout from the HCD gen listing the channel path detailed report:

```
                                  CHANNEL PATH DETAIL REPORT                    TIME: 15:47  DATE: 95-08-04  PAGE E-   8
PROCESSOR ID FG9121   TYPE 9121     MODEL 320      CONFIGURATION MODE: LPAR  TOKEN: FG9121   95-08-04 14:58:44 IODF     IODF80
                           --- SWITCH ----                                    UNIT ADDR                    UNIT
                           ID  PR  CU  DYN  --- CONTROL UNIT ---  CU-           RANGE    -- DEVICE --  ADDR   DEVICE
CHPID  TYPE    SIDE  MODE     PN  PN  ID   NUMBER  TYPE-MODEL    ADD  PROTOCOL  FROM  TO  NUMBER,RANGE  START  TYPE-MODEL
-----  ------  ----  ----  ----------  --  ------  ------------  ---  --------  ----  --  ------------  -----  -------------

 0C    BL            DED                 00C1  3088                 S4        40   4F   0C40,16       40    3088
                                         00C2  RS6K                 S4        50   5F   0C50,16       50    RS6K
                                    ---------------------------------------------------------------------------------------
                                         0C80  3174                 S         60   6F   0C60,16       60    3278-2
                                    ---------------------------------------------------------------------------------------
                                         0C70  3174                 S         70   77   0C70,8        70    3174
                                    ---------------------------------------------------------------------------------------
                                         0C71  3791L                S         78   7F   0C78,8        78    3791L
---------------------------------------------------------------------------------------------------------------------------
  2F    CNC           REC                 0184  SCTC          1              10   11   0910,2        10    SCTC
                                    ---------------------------------------------------------------------------------------
                                         0185  SCTC          2              12   13   0912,2        12    SCTC
                                    ---------------------------------------------------------------------------------------
                                         0186  SCTC          3              14   15   0914,2        14    SCTC
                                    ---------------------------------------------------------------------------------------
                                         0187  3174          4   S          16   16   0916          16    3174
                                    ---------------------------------------------------------------------------------------
                                         0188  SCTC          5              18   1F   0918,8        18    SCTC

ESOTERIC NAMES

-
                                        AFFINITY  ALLOCATION   ASSOCIATED
  NAME        NAME TYPE  VIO  TOKEN  PREF   INDEX    DEVICE TYPE  GENERICS                DEVICE NUMBER LIST
  ---------   ---------  ---  -----  ----  --------  -----------  ----------  ------------------------------------------------
  ------------------------------------------------------------------------------------------------------------------------
  RS6K17      ESOTERIC              8300    FFFF     00004101     SCTC        0912 0913 0914 0915
  ------------------------------------------------------------------------------------------------------------------------
  RS6K43      ESOTERIC              8389    FFFF     00004103     RS6K        0C50 0C51 0C52 0C53 0C54 0C55 0C56 0C57 0C58 0C59 0C5A
                                                                             0C5B 0C5C 0C5D 0C5E 0C5F
```

Figure 5. HCD gen listing

The channel type for ESCON is CNC, which is the ESCON channel. The switch number is the value specified for adapter connections through an ESCON Director. For HCD gen systems, it identifies the routing between the RS/6000 and the mainframe. For two ESCON directors, the dynamic director is specified. For IOCP, the SWITCH field contains this value, as can be seen in 3.1.5, “Setting Up the ESCD for VM TCP/IP over ESCON” on page 30.

3.1.3.6 Device Resolution for Channel Products 3088, RS6K and 3174

The channel path detailed report in the HCD listing is shown in Figure 5. HCD defined the CHPID number, the CHPID type, the Control Unit type and the device address for the control unit associated with this CHPID. The sense ID data is obtained from the RS/6000 and processed by the Unit Information Module, UIM, whis is designed to work with the HCD program. The RS/6000 emulates the 3088 channel to channel control unit. With the increase in the number of control units that emulate the 3088, the RS/6000 implemented it’s own control unit and device type of RS6K. If the MVS system returns an IOS message indicating a logical or physical unit mismatch when the RS/6000 is varied on, either the control unit type can be changed in HCD to RS6K, or the message can be ignored.

From the listing, one can read the headings where:

- CHPID for MVS is 2F
- CHPID TYPE is CNC
- CONTROL UNIT TYPE is SCTC
- Unit address and range 10 to 16

- Device Address 0910, 0914 (Note: 916 is reserved for SNA as defined by control unit type 3174.)
- No Link address for ESCON Director (default)
- CUADD parameter was 1, 2, 3, or 4, depending on the subchannel address

  The CUADD parameter or the virtual control unit address allows you to define multiple control units on one link or fibre channel.

Enter the following command at the RS/6000 prompt:

```
smitty esca
```

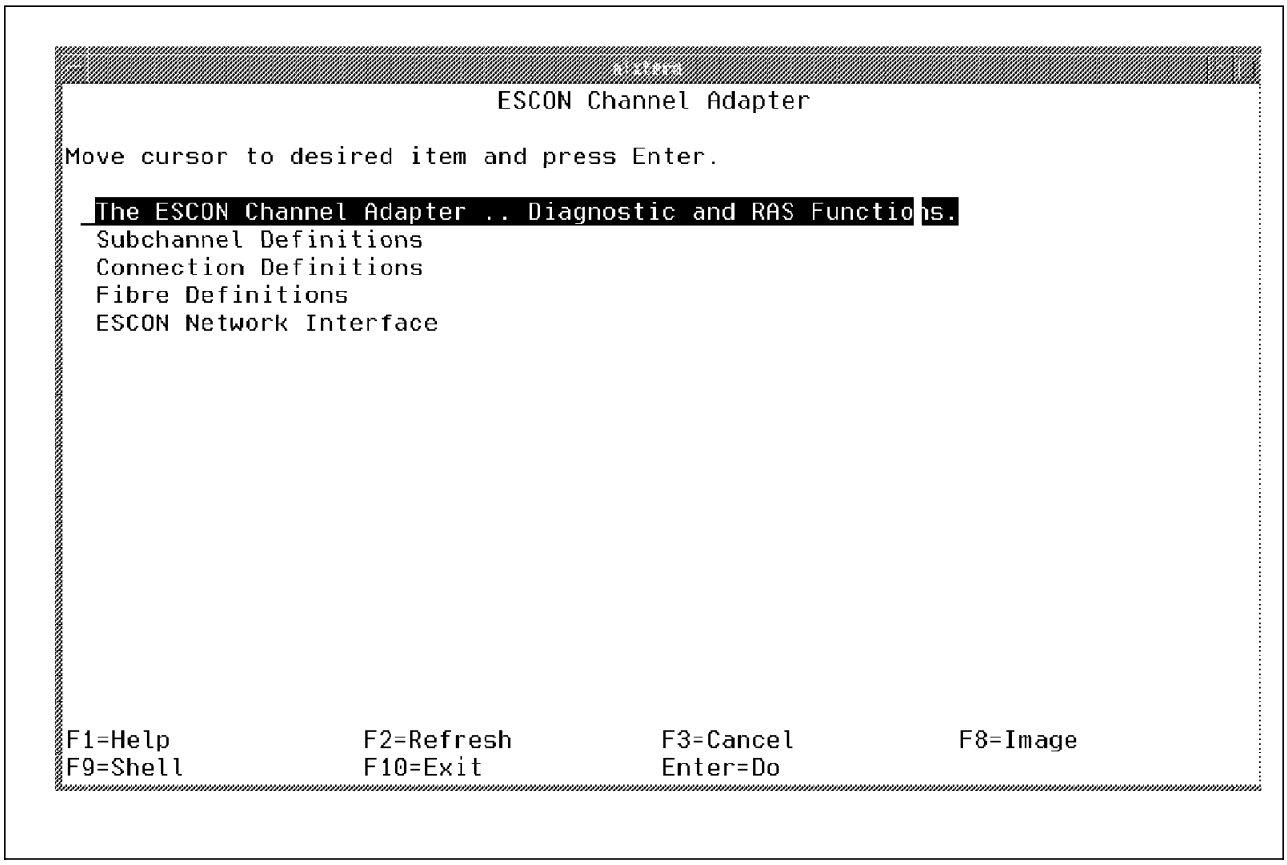


Figure 6. Defining an ESCON Channel Adapter

There are five configuration menus to follow after the one shown in Figure 6, and we will follow each of them for our particular environment.

3.1.3.7 ESCON Adapter Definitions

The reboot of the system after hardware install automatically creates an adapter definition. Two cards were installed in ′dhatrisc′, in slot 5 and slot 16, that were defined as escon0 and escon1, respectively. Please refer to A.9, “LSCFG of Test RS/6000.” on page 161.

3.1.3.8 Using SMIT to define an ESCON Channel Adapter

Enter the command:

```
smit chgesca
```

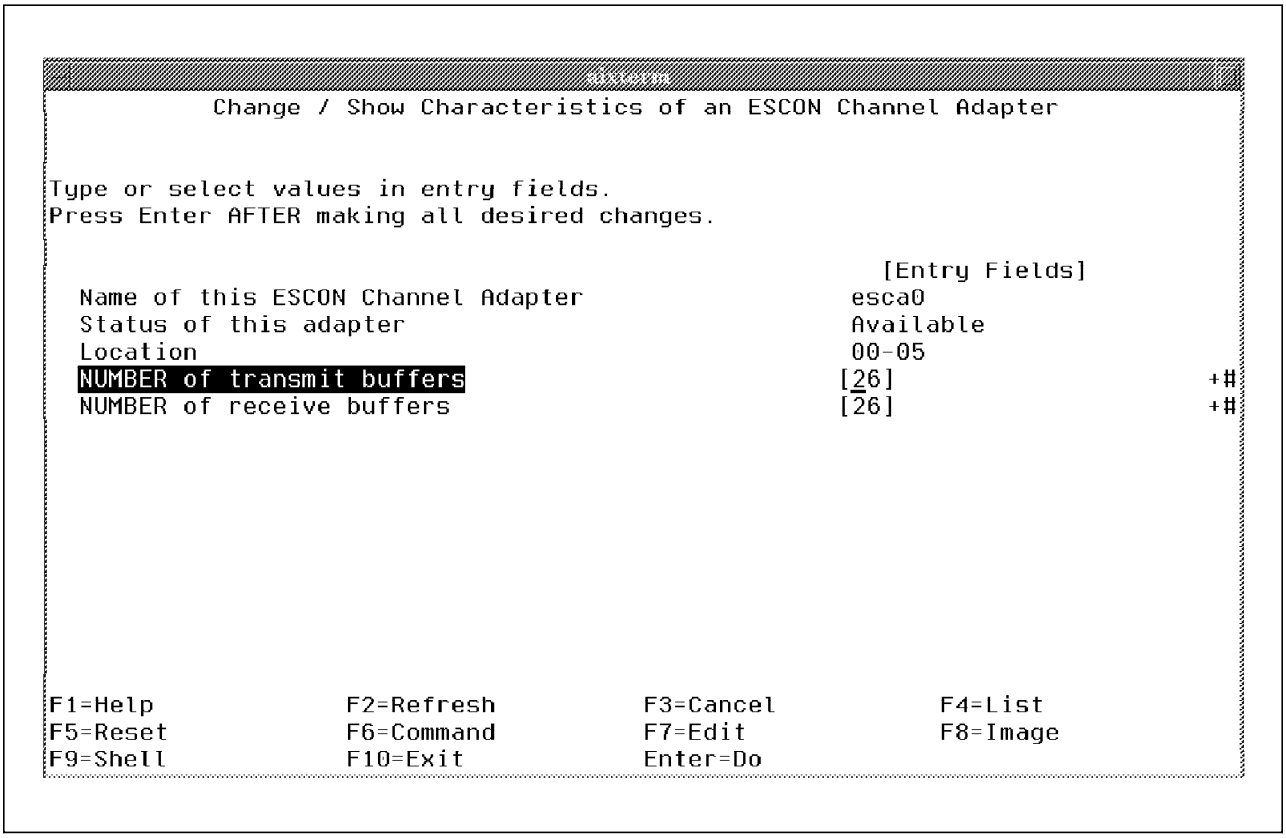


Figure 7. Definition of the ESCON Channel Adapter

As in Figure 7, the number of transmit buffers, 26, should match the number defined in the hardware definitions of the file PROFILE.TCPIP or those defined in A.2, “SYS1.TCPPARMS.AUSHATA” on page 138 in the host definitions. Each buffer is 4096 bytes. If data flows in one direction more than in the other, for instance to the RS/6000, then the number of receive buffers should be increased accordingly, to perhaps as many as 96.

There are two adapters installed, namely escon0 and escon1. They are configured for connection to the MVS mainframe and the VM mainframe, respectively. Use of the command `lsdev -Cc adapter` will also show the slot location and status of the channel adapters.

```
sio0   Available 00-00    Standard I/O Planar
scsi0  Available 00-00-0S Standard SCSI I/O Controller
scsi1  Available 00-08    SCSI I/O Controller
scsi2  Available 00-12    SCSI I/O Controller
fda0   Available 00-00-0D Standard I/O Diskette Adapter
sa0    Available 00-00-S1 Standard I/O Serial Port 1
sa1    Available 00-00-S2 Standard I/O Serial Port 2
tok0   Available 00-01    Token-Ring High-Performance Adapter
fddi0  Available 00-02    FDDI Primary Card, Single Ring Fiber
ppa0   Available 00-00-0P Standard I/O Parallel Port Adapter
chna0  Available 00-07    IBM S/370 Channel Emulator/A Adapter
chna1  Available 00-13    IBM S/370 Channel Emulator/A Adapter
cat    Available generic  370 Parallel Channel Adapter
escon0 Available 00-05    ESCON Channel Adapter
escon1 Defined   00-16    ESCON Channel Adapter
cat0   Available 00-04    370 Parallel Channel Adapter
cat1   Available 00-14    370 Parallel Channel Adapter
```

3.1.3.9 Defining ESCON Subchannels

Our system testing meant that we defined a reasonably complex subchannel scenario with definitions for CLAW, for the CLIO application and also for SNA. Subchannels for TCP/IP and SNA were also defined for connection to a VM host. At the time this redbook was written, there was no CLIO VM application available. An example of each of these definitions follows, with the overall connectivity definition in summary.

At the command line enter:

```
smitty escaaddSubd
```

As an example, we include one subchannel defined for our system, tcpip1, that is configured for CLAW channel connection to the MVS hosts in Figure 8 on page 25.

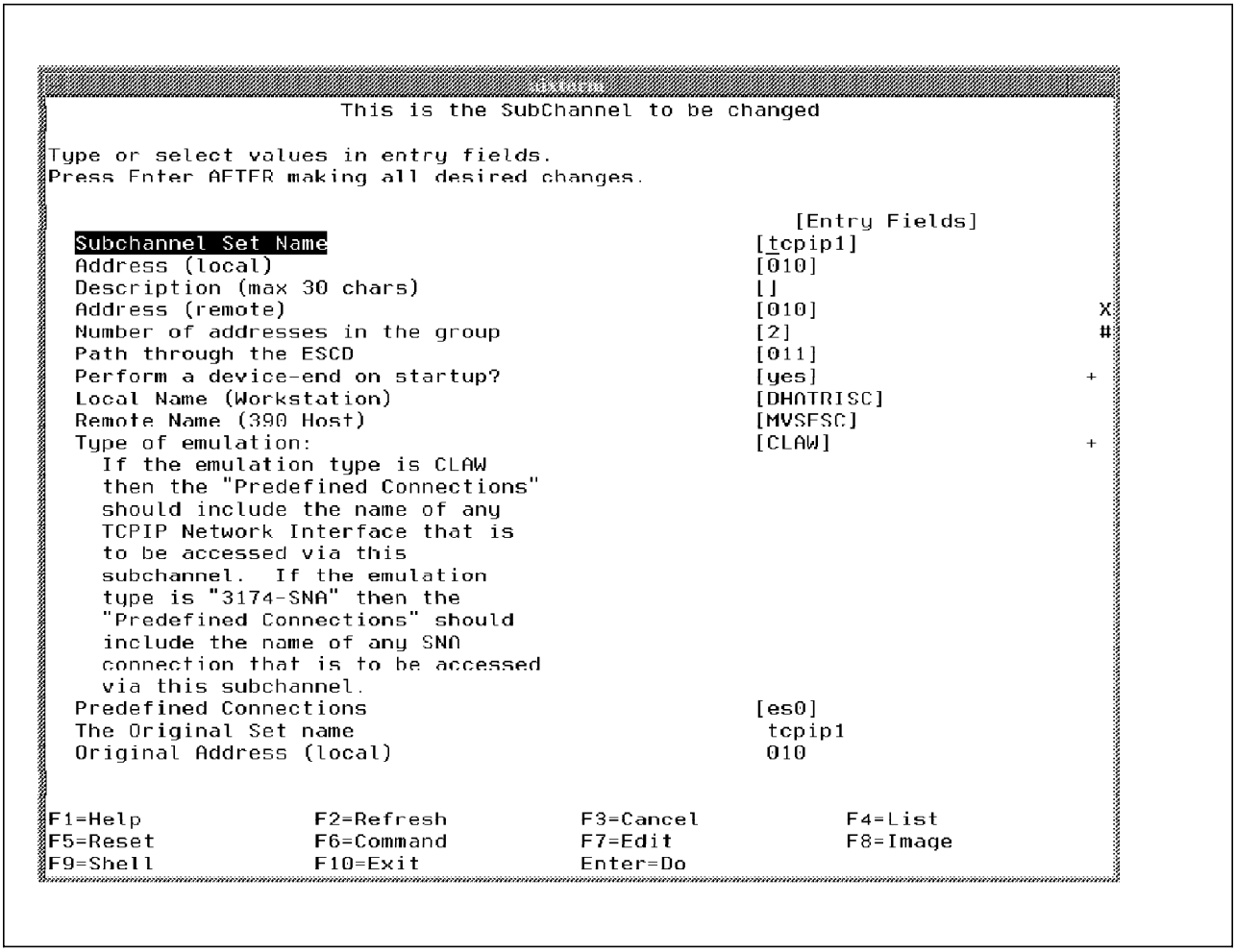


Figure 8. Defining ESCON Subchannels

The field definitions

- **Subchannel Set Name**

  This is a default name that should be consistent with the customer site. Note in the CLIO definitions that these were called CLIO′n′ for TCPIP TCPIP′n′, and for SNA, they were similarly defined.

  These names **must** match the fibre definition of the adapter.

  Note: For AIX V4, the name for a TCP/IP connection must be the interface name, es(n).

- **Address**

  This is the local address used by the RS/6000. It enables an application to define the subchannel to use. It should be equivalent to the remote address below but unique in its own right in the RS/6000 subchannel definitions. It is customary to go to the remote address defined, and then copy the value into the local definition, being mindful of the uniqueness.

- **Description**

  The description is not necessary and perhaps makes things more complex. We left it out as a rule.

- **Address (remote)**

This is a single subchannel address. It is one of the addresses specified in the HCD gen (or IOCP in our case with VM) in the UNITADD field, and for our example, it is the specification for the TCP/IP CLAW channels 910 and 911. Note that for CLAW, two subchannels have to be defined, and they MUST be started on an even channel.

- **Number of addresses in the group**

  The system default is 2, but this value is only needed for TCP/IP and CLAW. The SNA and CLIO subchannels were only 1.

- **PATH through the ESCD**

  The system default is 010, but this is the most complex of the subchannel parameters, and it needs some explaining:

  There are two parts to the parameter that have to do with the path through the ESCD and with the CUADD field of the HCD gen or host IOCP macro.

  The first two hex digits are the port number of the ESCD to which the host channel is connected. The example above defaults to **01** as there is no ESCD. The low order hex digit is the value specified in the CUADD definition in the host. If the CUADD parameter is not specified, the low order digit is zero. In example Figure 13 on page 31 with the VM system shows the value C4 which is related to the link address of the ESCD. If two directors are used, the host port address of the dynamic ESCD should be used.

- **Perform a device-end startup?**

  You should enter **yes** for TCPIP and CLIO and **no** for SNA. Please see Figure 39 on page 95 for a further explanation of this parameter with SNA and TCP/IP.

- **Local name**

  This is required for SNA and TCP/IP. For TCP/IP it MUST have the same value as the host value for the workstation name in the TCPIP.PROFILE. In our case, see A.2, “SYS1.TCPPARMS.AUSHATA” on page 138.

- **Remote Name**

  This is required for SNA and TCP/IP. For SNA, this must match the VTAM PU definition used in the host. For TCP/IP, again, it must match the TCPIP.PROFILE file, or in our case, see A.2, “SYS1.TCPPARMS.AUSHATA” on page 138.

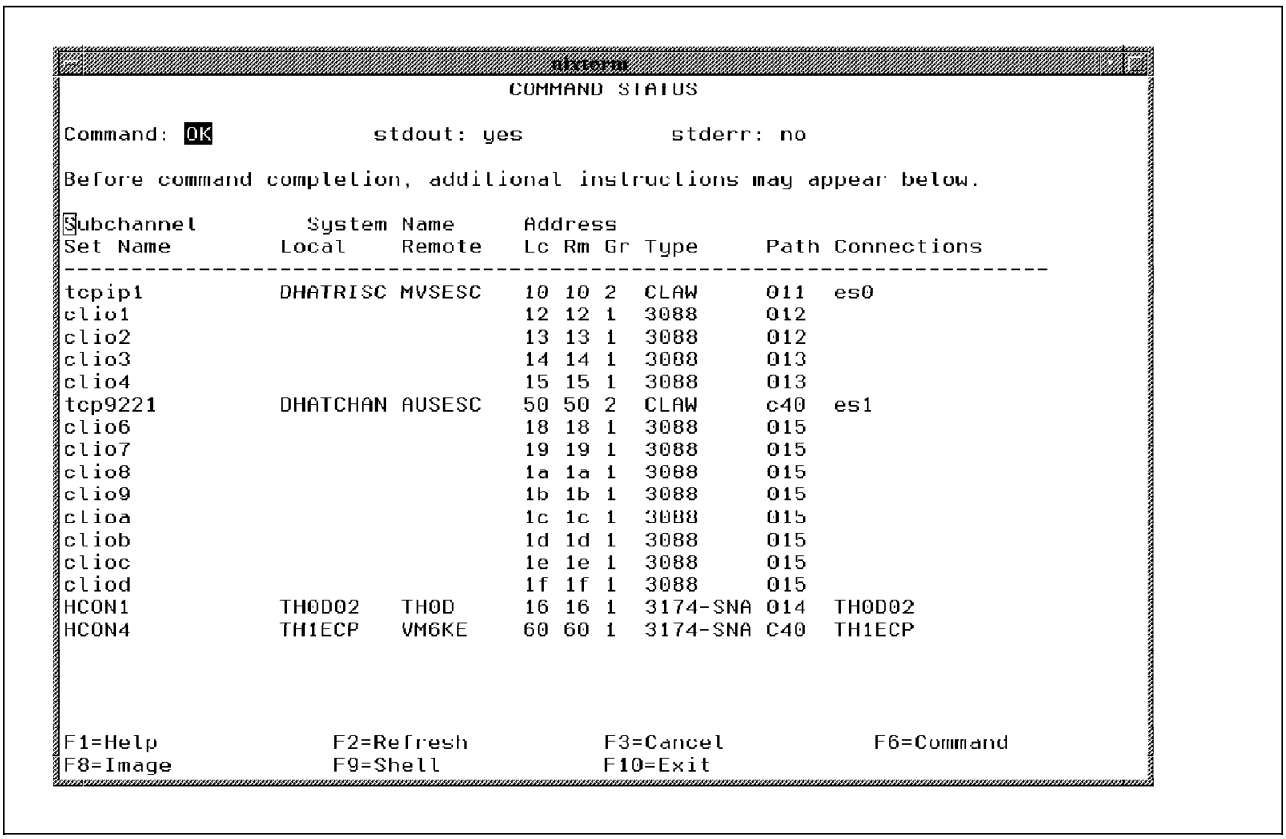


Figure 9. Subchannel Definitions

- **Type of Emulation**

  Select from 3088, 3174-SNA or CLAW for channel SNA and TCPIP, respectively. All three types are shown in the above diagram.

- **Predefined Connections**

  This value defines the connection name that is associated with the defined ESCON subchannel address. For TCP/IP, enter es0. For SNA, it should be the SNA Link Station Profile. For the normal channel definition for CLIO, take the default.

  Note: For AIX V4 this field is not used.

- **The Original Set Name**

  This shows the subchannel definition name as set above. The definition should be left to default to this.

- **Original Address**

  This shows the subchannel definition for the first local subchannel address. The definition should be left to default to this.

3.1.3.10 Some TCP/IP hints

- The subchannel MUST be configured as CLAW.
- The subchannel starting address must be even.
- The number of subchannels must be a multiple of two.
- The Local and Remote Names MUST be in uppercase and MUST match at both ends.

- A connection definition must exist which has Local and Remote Names and IP addresses.

3.1.3.11 Defining ESCON Connections

This definition will be created by default if the Predefined Connections field for the subchannel definition has been filled in. More detail of this screen will be discussed in the SNA setup section, but the correct definitions for all our connectivity SNA, IP and CLIO can be seen in Figure 10.

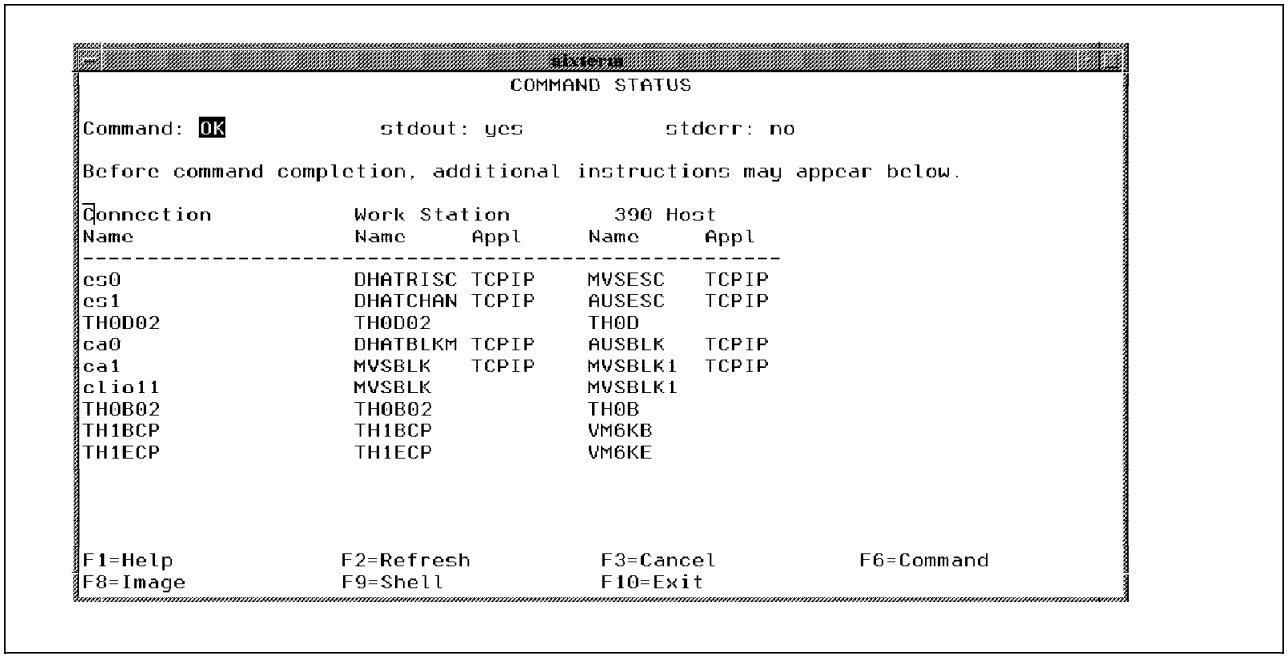


Figure 10. Defined ESCON Connections

3.1.3.12 Defining ESCON Fibres

The fibre definition identifies which ESCON card (via the slot number) is associated with which subchannel definition. The fibre has to have a label, which you should put in the slot number for reference only. The actual fibre name is required but not used. Figure 11 on page 29 shows the fibre definitions where slot5 is the connection to the MVS hosts, and slot16 is the connection to the VM host.

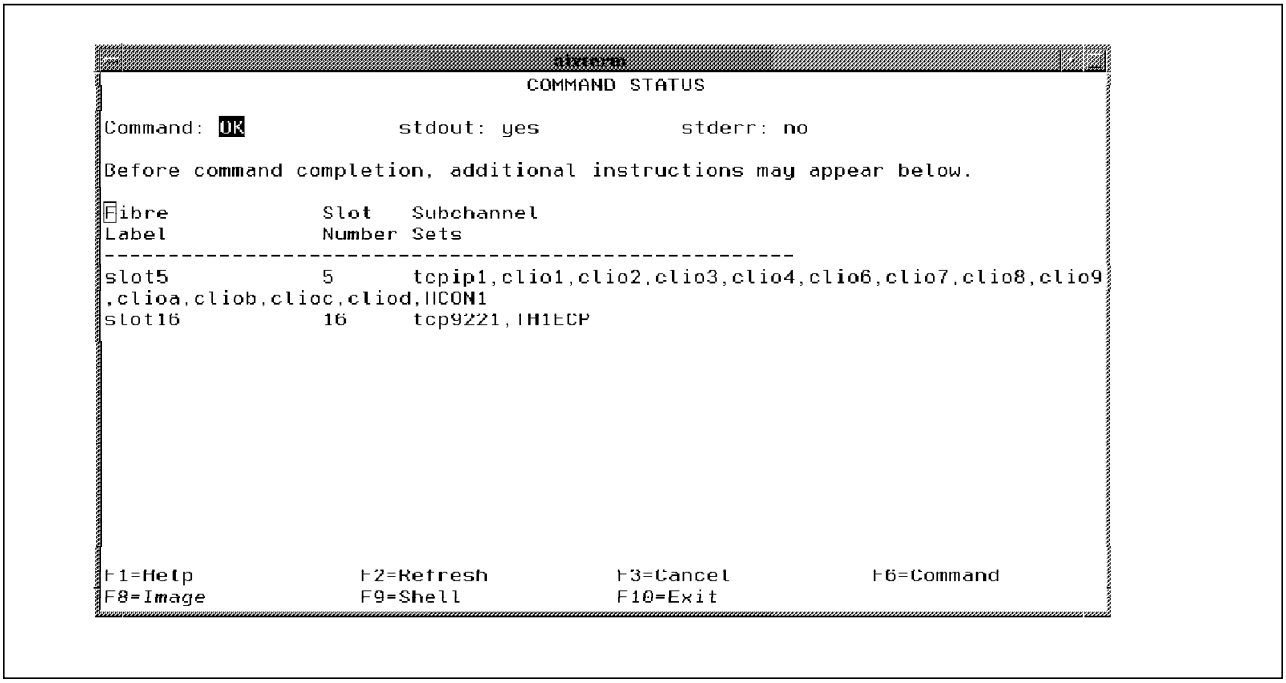


Figure 11. Fibre Definitions

3.1.3.13 Defining the ESCON Network Interface

This definition is for TCP/IP only, and the details must match the /etc/hosts file and the TCPIP.PROFILE definitions on the mainframe host. Please see A.2, “SYS1.TCPPARMS.AUSHATA” on page 138 for confirmation with Figure 12. The subchannel definitions **must** be in CLAW mode because this is for TCP/IP only.

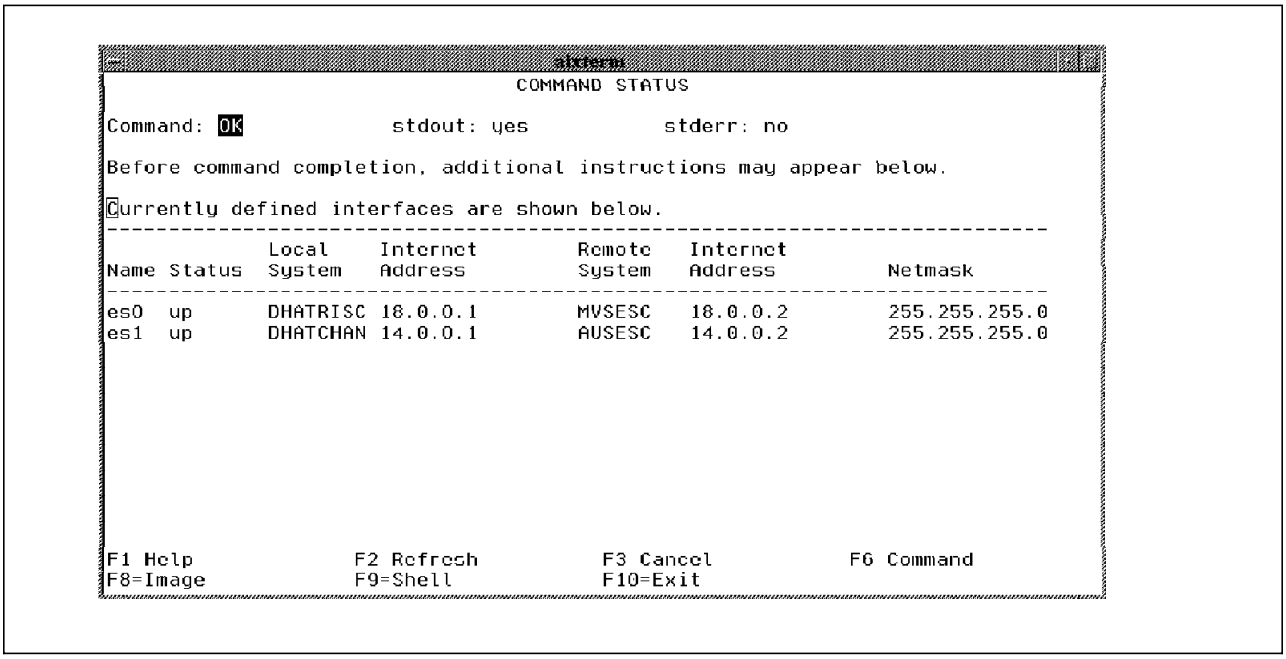


Figure 12. Network Interface

3.1.4 Testing the ESCON Channel

To verify the ESCON channel, issue the following command from the RS/6000 machine:

```
dhatchan:/> ifconfig es0     (or es1)
```

You should get a response equivalent to:

```
es0: flags=51<UP,POINTOPOINT,RUNNING>
   inet 18.0.0.1 --> 18.0.0.2 netmask 0xff000000
```

The 18.0.0.1 is the IP address of the ESCON adapter on the RS/6000, and 18.0.0.2 is the IP address of the MVS host.

To activate, issue the following commands from the RS/6000 machine:

```
dhatchan:/> rmdev -l escon0           (or escon1)
dhatchan:/> mkdev -l escon0           (or escon1)
dhatchan:/> ifconfig es0 18.0.0.1 18.0.0.2 mtu 4096 up
```

The 18.0.0.1 is the IP address of the ESCON adapter on the RS/6000, and 18.0.0.2 is the IP address of the MVS host. The mtu size of 4096 is the recommended size for this parameter.

Start TCP/IP on mvs with the S TCPIP command.

Verify that the subchannels are online at the MVS host console via:

```
d u,,,910,6
```

Subchannel 910 is the first looked at, and the 6 shows we are looking at the following 6 subchannels as defined in the HCD gen. The response should show that they are active and online.

Ping the MVS host from the RS/6000. Hopefully, you have established connectivity.

3.1.5 Setting Up the ESCD for VM TCP/IP over ESCON

Thus far, we have concentrated mainly on MVS; so, for a more complex path for using ESCON Directors, we have used VM connectivity. We set up a VM-based TCP/IP ESCON installation using the steps in the following sections.

3.1.5.1 Installing ESCON with One Director

We installed one director for the VM TCP/IP connection. The following information was needed from the IOCP definition.

- ESCON CHPID switch and port number
- The UNITADD, LINK (port number)

  We did not use a CUADD parameter, but this is needed if your environment is using one.

- Determine if this ESCD is static or dynamic. If two are used, the dynamic ESCD is the important one to note here.

Figure 13 shows the channel setup for our system. The lower path is the VM TCP/IP setup.

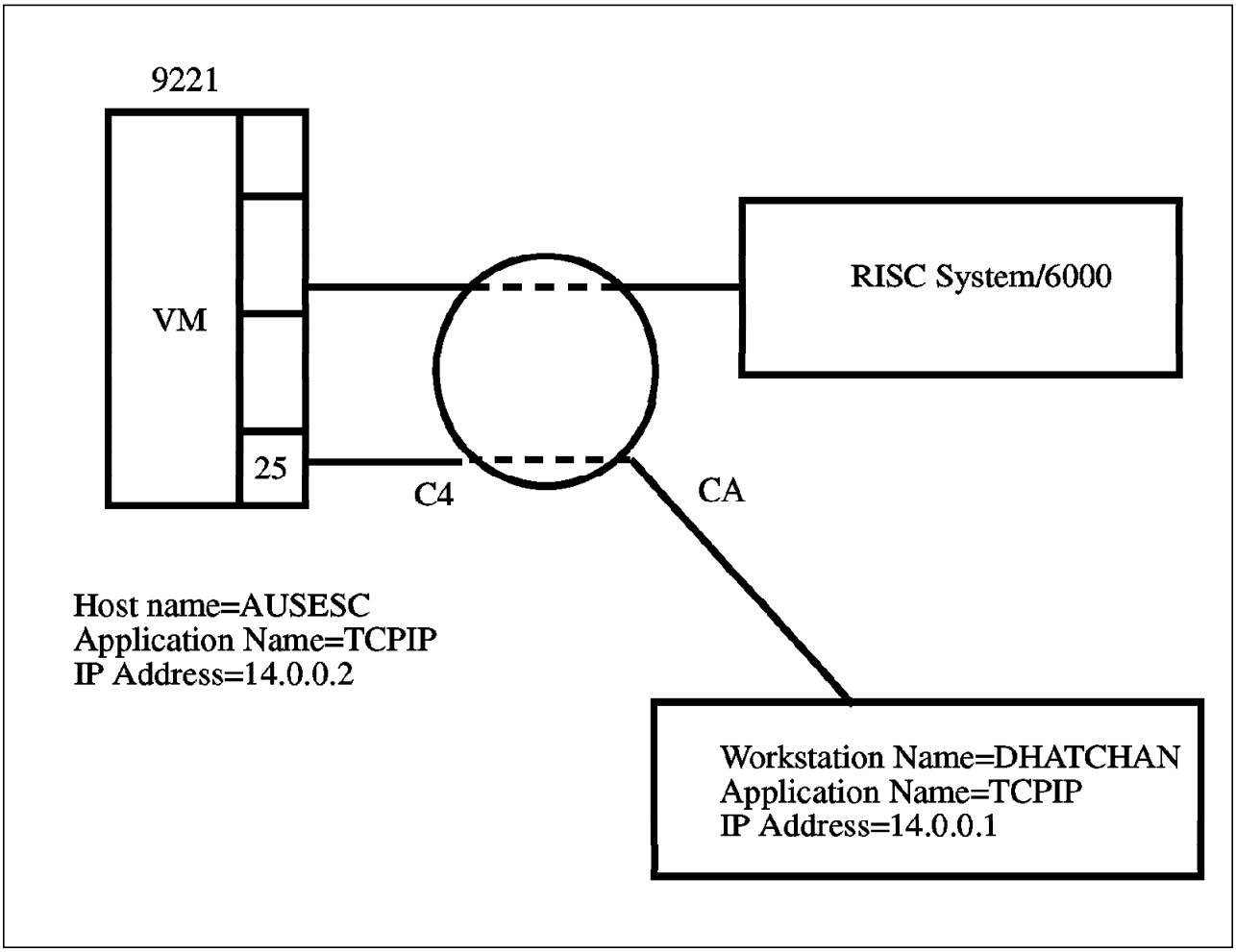


Figure 13. ESCD Setup for TCP/IP Attach to VM

From 3.1.5.2, “IOCP Definition,” channel 25 is defined as a normal (CNC) ESCON channel with the ESCON Director (9033). The RISC System/6000 is defined as an SCTC in the CNTLUNIT macro, and the LINK field defines the ESCON Director port CA. The UNITADD is from 50 upwards.

3.1.5.2 IOCP Definition

In order to make the channel connection between a mainframe and a workstation, you need the IOCP definition for the workstation on mainframe IOCDS (Input Output Configuration Data Set). Below is the IOCP definition for the channel and device (workstation) for the ESCON channel.

```
CHPID    PATH=((25)),TYPE=CNC,SWITCH=01
CNTLUNIT CUNUMBR=0250,PATH=(25),UNITADD=((50,032)),UNIT=SCTC,LINK=(CA)
IODEVICE ADDRESS=(A50,016),CUNUMBR=(0250),UNIT=SCTC
```

3.1.5.3 Defining the Subchannel Addresses

From the IOCP gen listing, we can define the subchannels. The local address and remote address for the first two subchannels are 50 and 51, respectively. These are the two CLAW (TCP/IP) addresses.

We have defined a connection path from channel 25 to DHATCHAN. We can therefore define the syntax of the channel path as C40, where the first two digits represent the link address for the 9221 CHPID on the ESCD, and the last digit, the 0, is the default because there is no CUADD parameter. If the CUADD parameter was 2, for instance, then the definition would be C42.

Chapter 4. RS/6000 S/370 Block Multiplexer Channel Connection

The BLKMUX channel adapter (or PCA) channel attaches to an ES/9000, 3090, 308X, or 4381 mainframe system. It effectively provides the functions required for the RS/6000 to be a control unit to the host.

4.1 Overview

The BLKMUX adapter from the mainframe view point should be thought of as an attached processor rather than as a control unit, and therefore, the RS/6000 should be running an application to be of any use. The CLIO product that is discussed in Chapter 6, " IBM Client Input Output/Sockets (CLIO)" on page 65is one such application.

The adapter can be defined in either CLAW or 3088 mode, and both modes provide an API for customer-written applications. Attachment to the mainframe is over the familiar BUS and TAG architecture associated with channel prior to ESCON. In setting up BLKMUX cabling, care must be taken to attach BUS to BUS and TAG to TAG between the mainframe and the RS/6000 or any intermediate device. The Block Multiplexer (BLKMUX) channel allows a single mainframe channel (copper) to attach up to eight control units.

There were two configurations for mainframe attachment in the Austin test setup. The one that we will discuss most in this chapter is the direct attach. The other is via 9034 attachment to the mainframe through fiber channel. Apart from the set up of the 9034, the two configurations look equivalent, as shown in Figure 14 on page 34.

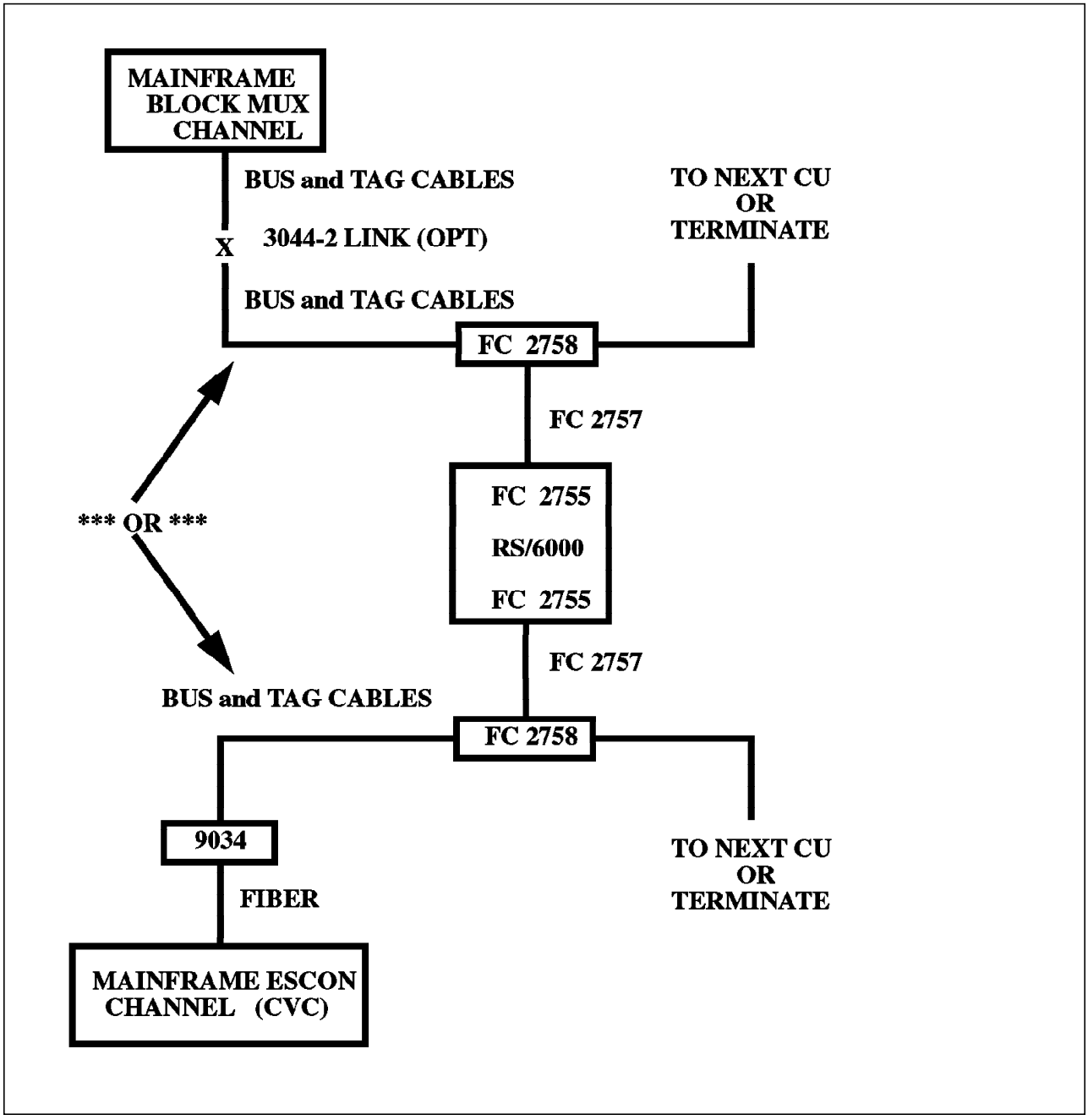


Figure 14. Block Multiplexer Channel Adapter Configuration

It is relatively easy to install the BLKMUX adapter through the SMIT interface of AIX. The major difficulties were in understanding the nomenclature between the two architectures and in ensuring that the channel definitions between the RS/6000 and the mainframe matched exactly. There are two BLKMUX device drivers: one for TCP/IP, feature code 5055, and one for TCP/IP and SNA, feature code 4760. You must order the correct one for your system if you are running at AIX 3.2.5.. The future development of the Block Multiplexer drivers will only be associated with the 4760 driver because this is the one that will be shipped with AIX 4.1.

Section 4.1.1, “Planning Information Required” on page 35 gives a list of details that must be thought about prior to installation. This list can be used as a checklist to determine the software and hardware requirements. After this, we show the exact details of our setup in 4.1.2, “BLKMUX Channel Implementation” on page 37.

4.1.1 Planning Information Required

Prior to install, it is necessary to collect the information below:

- Software requirements
  - AIX Version 3.2 or above
  - Feature 5055 or feature 4760 device driver
  - VM TCP/IP V2.2 or later (for TCP/IP with VM)
  - MVS TCP/IP V2.2 or later (for TCP/IP with MVS)
- Hardware Requirements
  - FC 2755 - Adapter
  - FC 2757 - Adapter cable
  - FC 2758 - Interface assembly
  - Channel bus and tag cables
- Supported Mainframes
  - ES9000-9021
  - ES9000-9121
  - ES9000-9221
  - 3090
  - 308X
  - 4381
- Supported RS/6000 Systems
  - 7012 - All models, 2 adapters
  - 7013 - All models, 2 adapters
  - 7015 - All models, 2 adapters (8 with an RPQ)
  - 7030 - All models, 2 adapters
- Mainframe Channel ID (CHPID)
- Unit address and range
- Device addresses for the mainframe subchannels (and type)
- Specific customer setup for TCP/IP files on MVS or VM
- If an application is used, the specific addresses associated with it
- RS/6000 and mainframe nodenames
- RS/6000 and mainframe IP addresses

4.1.1.1 Installation Sequence

The installation sequence is important in that the mainframe has to be gen'd before the connectivity can be tested. Here, we are discussing the FC 4760 driver only. The fast path for access to the RS/6000 menus is `smit blkmux`. This sequence should be followed:

- IOCDS gen (IOCP or HCD) (HCD in the MVS mainframe example here)
- Mainframe operating system I/O gen
- Mainframe application (TCP/IP) configuration
- RS/6000 Adapter S/W install
- RS/6000 Adapter card install and cabling
- RS/6000 Adapter Configuration via the SMIT interface
  - Select **Devices**
  - Select **communications**
  - **Select 370 Parallel Channel adapter**
  - **Configure and define the subchannels as follows:**

4.1.1.2 Define the Subchannels

- For addresses, mode (3088/CLAW/3174-SNA)
- Select **Subchannel Definitions**
- Select **Add a Subchannel**
- Fill in the Subchannel definition (Look below for careful hints.)
- Return to the BLKMUX Channel Adapter

4.1.1.3 Define Connections

- Select Connection definitions
- Select Add a connection
- Fill in as appropriate
- Return to the BLKMUX Channel Adapter

4.1.1.4 Define The Cables

The definition of the BLKMUX port ties in the subchannel addresses to the port with which they are associated.

- Select **BUS and TAG cables**
- Select **Add (or change if necessary) a Fibre**
- Fill in as appropriate
- Return to the BLKMUX Channel Adapter

4.1.1.5 Define BLKMUX Network Interface(s)

This is for TCP/IP only since there are associated IP addresses. This definition defines the interfaces that use the adapter.

- Select **BLKMUX Network Interface**
- Select **Add or Change an BLKMUX Network Interface**
- Fill in as appropriate

- Return to the BLKMUX Channel Adapter

4.1.2 BLKMUX Channel Implementation

The following sections outline in detail our specific setup for BLKMUX channels in the Austin test site.

4.1.2.1 BLKMUX Installation

The installation of the BLKMUX adapter is best accomplished in three phases.

1. Install the software making sure that the correct device driver is applied (see below).
2. Install the adapter card in RS/6000 after power-down.
3. Reboot the machine. The system should recognize the adapter, and issuing the command: `> lsdev -Cc adapter` should give you a defined adapter and interface.

IBM supplies two drivers for the BLKMUX adapter. One is a combined SNA and IP driver (feature code 4760), and the other is the TCP/IP-only driver (feature code 5055).

The installation media should explain which driver you have; however, if the driver is already installed and you are just reconfiguring, a simple way of determining which driver is installed is to type the following command at the RS/6000 command line:

```
dhatrisc:#>smitty blkmux
```

The FC 4760 driver has a SMIT fast path of `blkmux`; the FC 5505 driver does not. If you have AIX 4.1, FC 5055 is not supported.

Software Installation: The software installation for the BLKMUX driver was from tape using the familiar SMIT install panels. At level 3.2.5, it is important to distinguish between the device drivers, as was discussed above.

Verification of Correct Install: To verify that you have the correct hardware and software, issue the command (for the FC 5055 driver):

```
dhatrisc:#>lslpp -h blkmux.obj
```

4.1.2.2 BLKMUX Device Driver Configuration

The MVS BLKMUX link is direct attached for our test work and does not pass through any BLKMUX 9034 ESCON converters. We did, however, have another RS/6000 on the network with BLKMUX attached via 9034s, and the only difference in setup is with the subchannel definitions and the CHPID definition for CVC. The MVS system is HCD defined, and the listing associated with the BLKMUX channels is shown in Figure 15 on page 38. Becoming familiar with the contents of the HCD listing is a high priority as attention to detail is very important when setting up the subchannels.

4.1.2.3 Device Resolution for Channel Products 3088, RS6K and 3174

The channel path detailed report in the HCD listing is shown below. When MVS gens, it associates a CHPID (for instance for BLKMUX), such as 0C, with a control unit number or type, such as BLKMUX or BL, in the HCD gen.

```
                                CHANNEL PATH DETAIL REPORT                    TIME: 15:47  DATE: 95-08-04  PAGE E-   8
PROCESSOR ID FG9121   TYPE 9121     MODEL 320      CONFIGURATION MODE: LPAR  TOKEN: FG9121   95-08-04 14:58:44 IODF     IODF80
                      --- SWITCH ----                                    UNIT ADDR                    UNIT
                      ID  PR  CU  DYN  --- CONTROL UNIT ---  CU-           RANGE    -- DEVICE --  ADDR   DEVICE
CHPID  TYPE    SIDE  MODE      PN  PN  ID   NUMBER  TYPE-MODEL    ADD  PROTOCOL  FROM  TO   NUMBER,RANGE  START  TYPE-MODEL
-----  ------  ----  ----  ----------  ---  ------  ------------  ---  --------  ----  ---  ------------  -----  -------------
 0C    BL            DED                 00C1  3088                   S4        40   4F   0C40,16       40     3088
                                         00C2  RS6K                   S4        50   5F   0C50,16       50     RS6K
                     ------------------------------------------------------------------------------------------------------
                                         0C60  3174                   S         60   6F   0C60,16       60     3278-2
                     ------------------------------------------------------------------------------------------------------
                                         0C70  3174                   S         70   77   0C70,8        70     3174
                     ------------------------------------------------------------------------------------------------------
                                         0C71  3791L                  S         78   7F   0C78,8        78     3791L
-------------------------------------------------------------------------------------------------------------------------------
 2F    CNC           REC                 0184  SCTC            1               10   11   0910,2        10     SCTC
                     ------------------------------------------------------------------------------------------------------
                                         0185  SCTC            2               12   13   0912,2        12     SCTC
                     ------------------------------------------------------------------------------------------------------
                                         0186  SCTC            3               14   15   0914.2        14     SCTC
                     ------------------------------------------------------------------------------------------------------
                                         0187  3174            4   S           16   16   0916          16     3174
                     ------------------------------------------------------------------------------------------------------
                                         0188  SCTC            5               18   1F   0918,8        18     SCTC

ESOTERIC NAMES

-
                                        AFFINITY  ALLOCATION   ASSOCIATED
 NAME        NAME TYPE  VIO  TOKEN  PREF  INDEX    DEVICE TYPE  GENERICS              DEVICE NUMBER LIST
 ---------   ---------  ---  -----  ----  --------  -----------  ----------  ---------------------------------------------
 ------------------------------------------------------------------------------------------------------------------------
 RS6K17      ESOTERIC             8300   FFFF      00004101     SCTC        0912 0913 0914 0915
 ------------------------------------------------------------------------------------------------------------------------
 RS6K43      ESOTERIC             8389   FFFF      00004103     RS6K        0C50 0C51 0C52 0C53 0C54 0C55 0C56 0C57 0C58 0C59 0C5A
                                                                           0C5B 0C5C 0C5D 0C5E 0C5F
```

Figure 15. HCD gen listing

The explanation of the sense ID and data can be found in Chapter 3, “RISC System/6000 S/390 ESCON Channel Connection” on page 15.

From the HCD listing, we can verify that:

- The CHPID for MVS was 0C (BLKMUX)
- The PROTOCOL was S4 (4.5MB)

  Note: Other protocols are S for 3MB and D for DCI.
- Unit address and range 50 to 5F
- Device Address 0C50,16

  Note: Addresses 0C60,16 and 0C70,16 are reserved for SNA.

Enter the following command at the RS/6000 prompt:

```
smitty blkmux
```

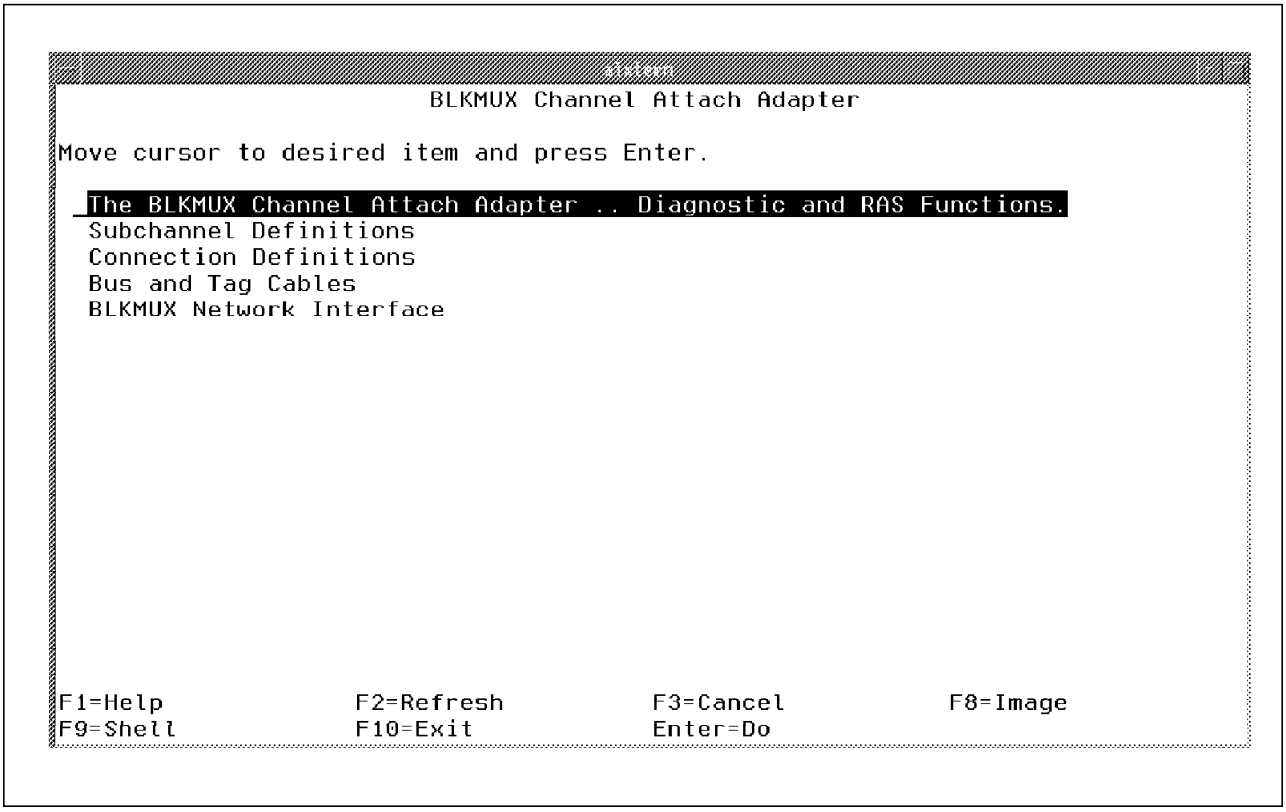


Figure 16. Defining an BLKMUX Channel Adapter

There are five menus to follow for the correct configuration, and we will follow each of them for our particular environment.

4.1.2.4 BLKMUX Adapter Definitions

Initially, we discuss the detailed install of the BLKMUX definitions Once we become familiar with these, we proceed to the SNA and CLIO definitions. If you are familiar with BLKMUX adapters, it may be wise to skip the next couple of pages and go directly to the SNA setup in order to save time.

The reboot of the system after hardware install automatically creates an adapter definition. Two cards were installed in dhatrisc, in slot 4 and slot 14, that were defined as cat0 and cat1, respectively.

4.1.2.5 Using SMIT to Define a BLKMUX Channel Adapter

Use the following SMIT fastpath to define the BLKMUX adapter:

```
smitty blkmux_chg
```

```
                                  smitty
        Change / Show characteristics of a BLKMUX Channel Attach Adapter
                These changes will apply to the data-base ONLY.
           They will affect active adapters only after the next ipl.


Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Name of this BLKMUX Channel Adapter                 cat0
  Slot occupied by this adapter                       00-04
* Channel SPEED for this adapter                      4                        +









F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 17. Define a BLKMUX Channel Adapter

The channel speed should be set to 4.5 MB/s on the RS/6000 for maximum speed. This matches the protocol parameter of the HCD gen of **S4**.

There were two adapters installed, cat0 and cat1, and they were configured for connection to the VM and MVS mainframes, respectively. Use of the command:

```
lsdev -Cc adapter
```

This will also show the slot location and status of the channel adapters. Exact details of the system can be found in A.9, “LSCFG of Test RS/6000.” on page 161.

4.1.2.6 Defining BLKMUX Subchannels

Subchannels were defined for TCP/IP (CLAW), for the CLIO application and for SNA in the following schematics. TCP/IP and SNA are also duplicated for connection to a VM host. At the time this redbook was written, there was no CLIO VM application available.

At the command line enter:

```
smitty blkmux_addSubd
```

As an example, we include one subchannel defined for our system, TCPIP2, for CLAW channel connection to the MVS hosts.

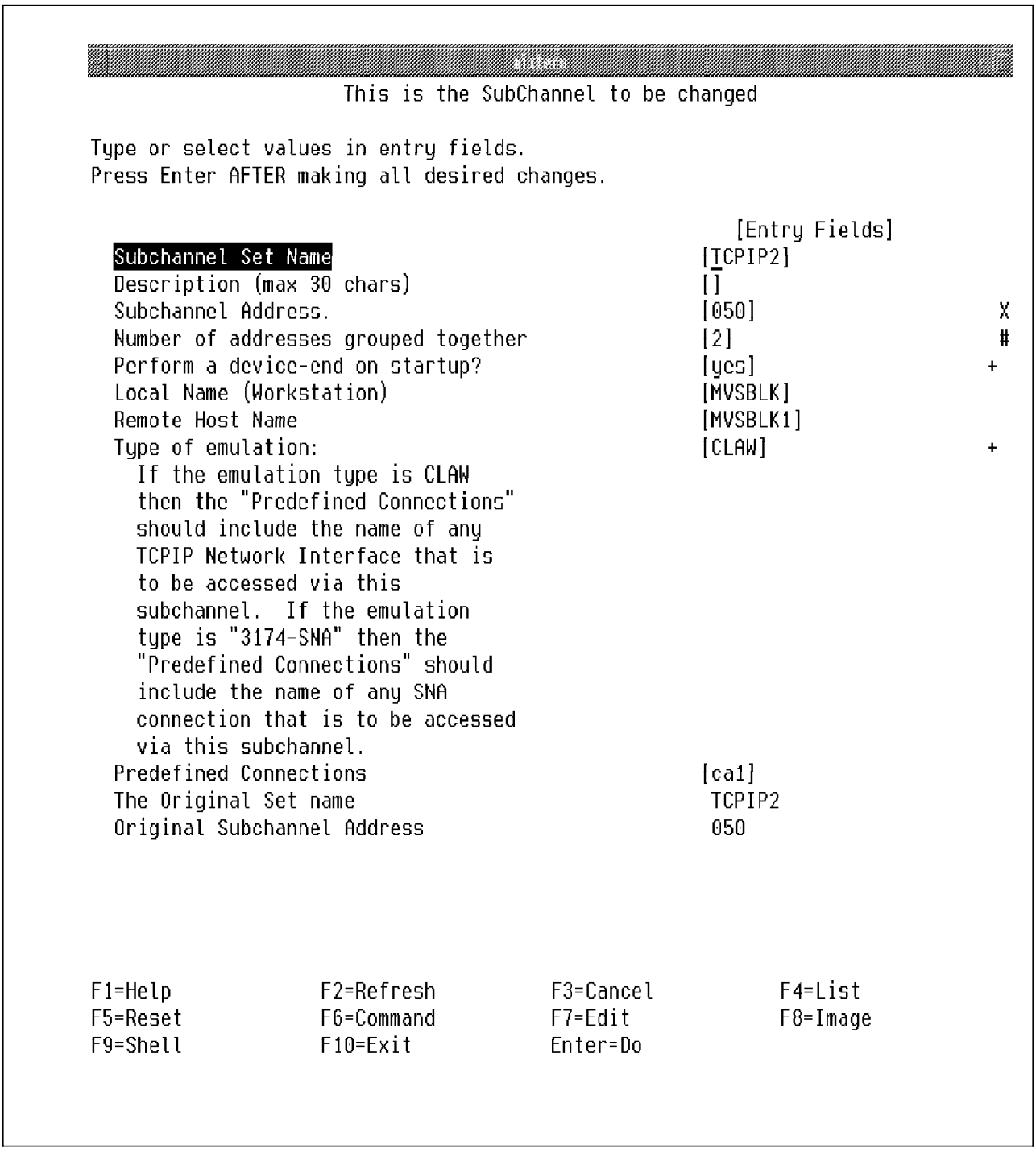


Figure 18. Add a BLKMUX Subchannel

The field definitions

- **Subchannel Set name**

  This is a default name that should be consistent with the customer site. Note that in the CLIO defitions, these were called CLIO′n′ for TCPIP TCPIP′n′, and for SNA, they were defined similarly.

  These names **must** match the Block Multiplexer cable definition of the adapter, as described later.

Note: For AIX V4 this name **must** be the TCP/IP interface name, ca(n).

- **Description**

  The description is not necessary and perhaps makes things more complex. We left it out as a rule.

- **Address (remote)**

  This is a single subchannel address. It is one of the addresses specified in the HCD gen, or IOCP in our case with VM, in the UNITADD field. For our example above, the TCP/IP CLAW channels were 050 and 051. For CLAW, two subchannels have to be defined, and they MUST be started on an even channel.

- **Number of addresses in the group**

  The system default is 2, but this value is only needed for TCP/IP and CLAW. The SNA and CLIO subchannels require only one.

- **Perform A device-end startup?**

  Yes for TCP/IP and CLIO; No for SNA.

- **Local Name (workstation)**

  This is required for TCP/IP. It MUST have the same value for the host value as the value shown for the workstation name in the TCPIP.PROFILE on the host.

- **Remote Host Name**

  This is requited for SNA and TCP/IP. For SNA, this must match the VTAM PU definition used in the host. For TCP/IP, it must match the TCP/IP.PROFILE file. It is not used for CTC applications, like CLIO.

- **Type of Emulation**

  Select from 3088, 3174-SNA or CLAW for channel SNA and TCP/IP, respectively. All three types are shown in Figure 18 on page 41.

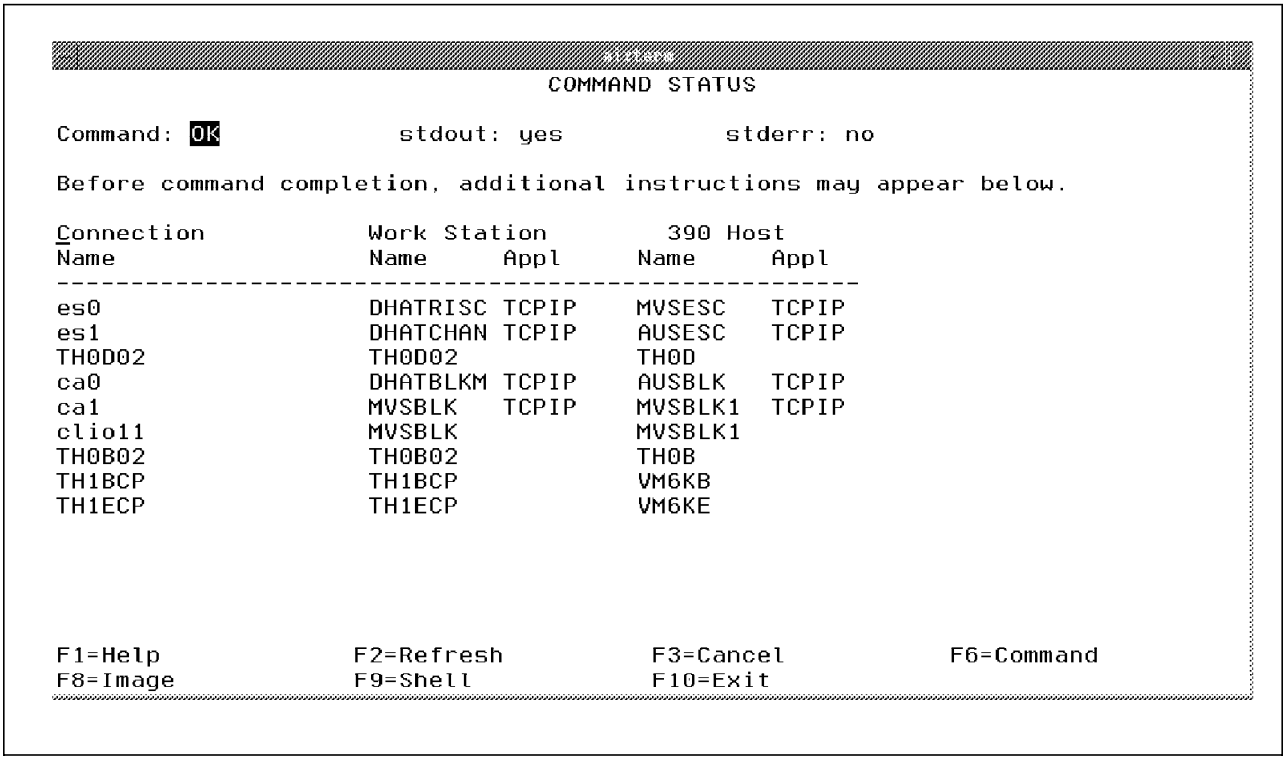


Figure 19. Subchannel Definitions

- **Predefined Connections**

  This value defines the connection name that is associated with the defined BLKMUX subchannel address. For TCPIP, enter ca1. For SNA, it should be the SNA Link Station Profile. See Chapter 5, “SNA Channel Connectivity Using AIX SNA Server/6000” on page 49. For the normal channel definition for CLIO, leave it blank or default.

  Note: This field is not used for AIX V4.

- **The Original Set Name**

  This shows the subchannel definition name as set above. The definition should be left to default. .

- **Original Address**

  This shows the subchannel definition for the first local subchannel address. The definition should be left to default. .

4.1.2.7 Some TCPIP Hints

- The subchannel MUST be configured as CLAW.
- The subchannel starting address must be even.
- The number of subchannels must be a multiple of two.
- The Local and Remote Names MUST be in uppercase and MUST match at both ends.
- A connection definition must exist which has Local and Remote Names and IP addresses.

4.1.2.8 Defining BLKMUX Connections

Type in the following command, or follow the menu path.

```
smitty blkmux_addCOND
```

This definition will be created by default if the Predefined Connections field for the subchannel definition has been filled in when the subchannels are first configured. More detail of this screen will be discussed in the SNA setup section, but as seen below, the correct definitions for all our connectivity SNA, IP and CLIO should be:

```
                                  Change a Connection

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Name of Connection to change                        cal                +
  Description (max 30 chars)                         []
  Name of Local System (Work-station)                [MVSBLK]
  Name of Local application.                         [TCPIP]
  Name of Remote Host System                         [MVSBLK1]
  Name of the Remote application.                    [TCPIP]

F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 20. Defined BLKMUX Connections

You must get the system names, shown previously, from the MVS host files and from the /etc/hosts file.

4.1.2.9 Defining BLKMUX Cables

The cable definition identifies which BLKMUX card (via the slot number) is associated with which subchannel definition. The subchannel definition must be referenced in the cable name prior to the device reconfiguration by using the mkdev command. The cable must have a label with respect to the slot number, but it is for reference only.

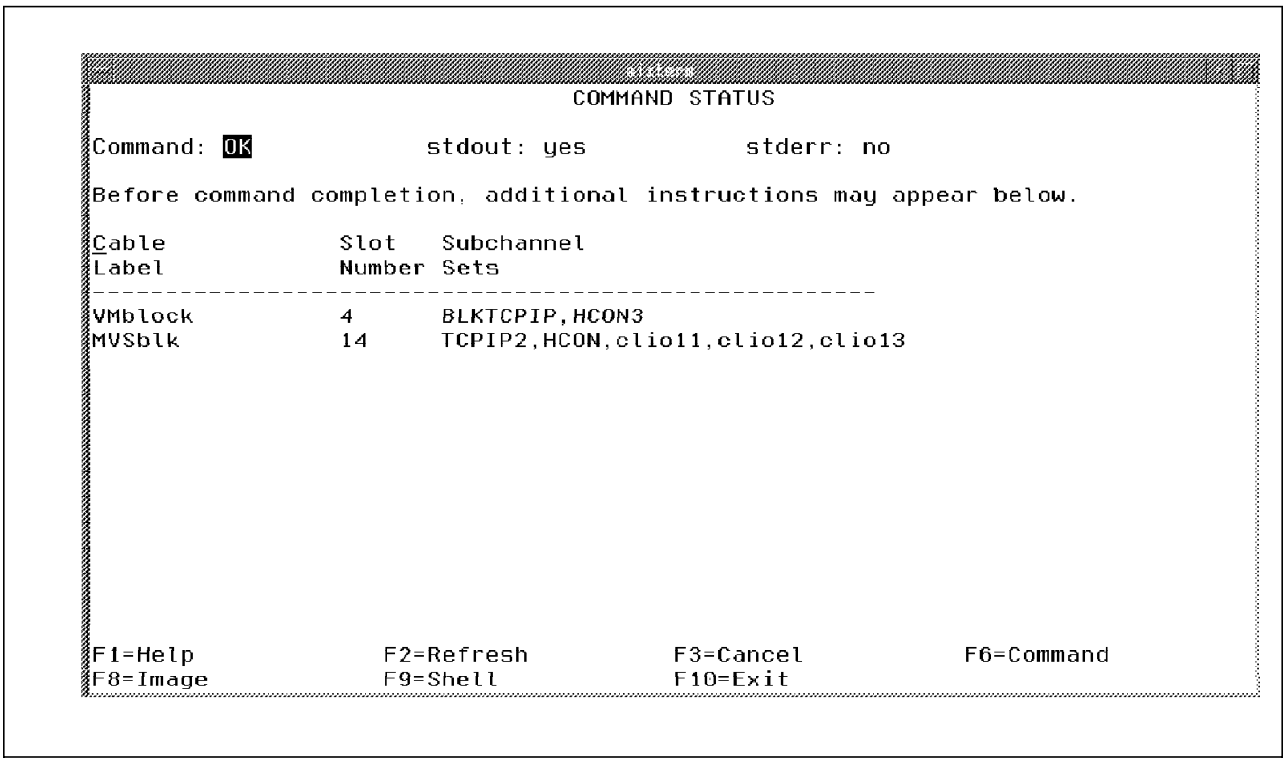


Figure 21. Block Multiplexer Cable Definitions

4.1.2.10 Defining the BLKMUX Network Interface

This definition is for TCP/IP only, and the details must match the /etc/hosts file, and the TCPIP.PROFILE definitions on the mainframe host. Please see Appendix C, “VM SNA Configuration Files” on page 177 for confirmation with the schematic below. The subchannel definitions must be in CLAW mode because this is for TCP/IP.

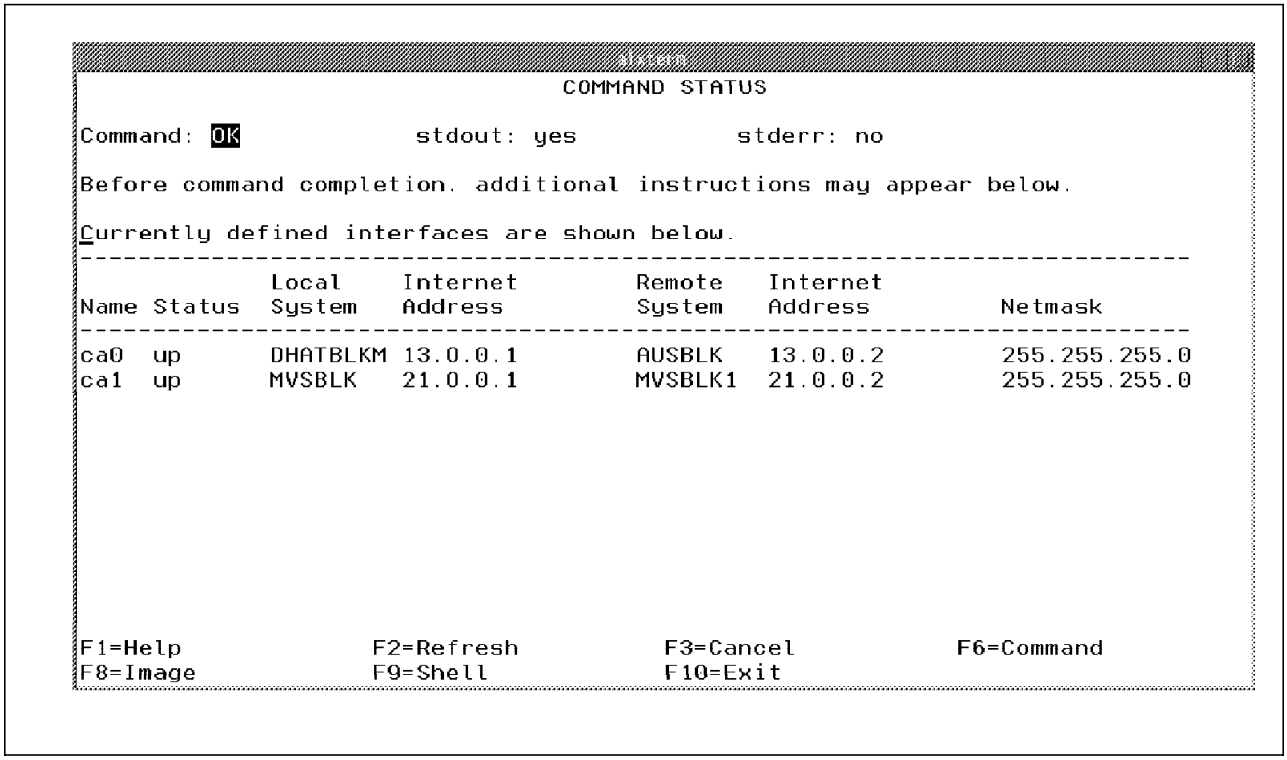


Figure 22. Network Interface

4.1.3 Testing the BLKMUX Channel

To verify the BLKMUX channel, issue the following command from the RS/6000 machine:

```
dhatchan:/> ifconfig cat0     ( or cat1 )
```

You should get a response equivalent to:

```
es0: flags=51<UP,POINTOPOINT,RUNNING>
   inet 21.0.0.1 --> 21.0.0.2 netmask 0xff000000
```

The 21.0.0.1 is the IP address of the BLKMUX adapter on the RS/6000, and 21.0.0.2 is the IP address of the MVS host.

To activate, issue the following commands from the RS/6000 machine:

```
dhatchan:/> rmdev -l cat0          (or cat1)
dhatchan:/> mkdev -l cat0          (or cat1)
dhatchan:/> ifconfig es0 21.0.0.1 21.0.0.2 mtu 4096 up
```

The 21.0.0.1 is the IP address of the BLKMUX adapter on the RS/6000, and 21.0.0.2 is the IP address of the MVS host. The mtu size of 4096 is the recommended size for this parameter.

Start TCP/IP on the MVS system with the S TCPIP command. Verify that the subchannels are online at the MVS host console via:

```
d u,,,c50,6
```

Subchannel c50 is the first looked at, and the 6 shows we are looking at the following 6 subchannels. The response shows they are online.

Ping the MVS host from the RS/6000. Hopefully, you have established connectivity.

Chapter 5. SNA Channel Connectivity Using AIX SNA Server/6000

This section is about setting up and testing IP and SNA connectivity over the **same** ESCON or BLKMUX channel on both VM and MVS.

For each channel adapter type on the RS/6000, you can define an SNA Data Link Control (DLC) and one or more SNA Link Stations to SNA Server/6000. Each link station is a PU (physical unit) and attaches to one host subchannel. The link station and each subchannel are associated by the *connection name* that is defined for the RS/6000 workstation and referred to in the SNA Channel AIX Link Station Profile.

There are some particular nuances involved with starting up SNA Server with respect to the VTAM definitions on the mainframe host and with varying on the particular channel. As a rule, the RS/6000 link station must be initialized **first**. Once the RS/6000 is configured properly it appears as a normal control unit, as designed.

This section contains the details associated with the MVS setup, and it references the VM setup. The two setups are almost identical in terms of the SNA configuration, and we include the subchannel VM addresses in most of the SMIT screens below. If a particular VM dataset is specifically used, it is referred to in detail and commented upon.

5.1 Install of SNA Server/6000 and Mainframe Definitions and Datasets

Installing SNA for ESCON and BLKMUX is standard in that the SMIT install of SNA Channel for AIX feature installs microcode for the adapter card, the device driver for the adapter, and the the channel DLC. There are two channel drivers with AIX 3.2.5. The TCP/IP-only driver has feature codes for BLKMUX of 5055 and for ESCON of 5056. The TCP/IP and SNA driver has feature codes for BLKMUX of 4760 and for ESCON of 4761. Historically, both drivers can work together, but it is more advisable to choose only one; and in the case of SNA, this must be the FC 4760 or FC 4761 driver. AIX 4.1 supports only the FC 4761 and FC 4760 drivers.

5.1.1 SNA Channel for AIX Packaging of the Files

There are five main files:

- **sna.dlcchannel.obj**

  Contains the SNA Channel for AIX DLC software.

- **sna.escon.usr**

  Contains the device driver software for the ESCON adapter.

  This replaces feature 5056 if this is already installed on your system. You can check feature codes by using the following command:

```
lslpp -ciq | tr ' ' ':' | cut -f 3,5,7 -d: |
    cut -c1-20,23-40 | sort | uniq
```

- **sna.escon.cuu**

  Contains the microcode for the ESCON adapter.

- **sna.blkmux.usr.**

Contains the device driver software for the BLKMUX adapter.

Note: If you are installing the feature code 5055 driver, install this first. In this respect, one should have a very good reason for running both drivers at the same time.

- **sna.blkmux.cuu**

  Contains the microcode for the BLKMUX adapter.

5.1.2 VTAM/Mainframe Datasets

An RS/6000 connected directly by channel to a mainframe host is considered to be a PU type 2.1 peripheral node. In order to ensure ease of use and connectivity for this work, the RS/6000 was set up to do 3270 emulation using HCON via LU 6.2 communication.

This redbook is not a detailed analysis of SNA connectivity and function; it is mainly a guide to what the RS/6000 specialist needs to quickly get up and running and the main issues associated with the editing of the mainframe data sets involved. Necessarily, the setup with customers will be somewhat different, but the main issues will be the same.

To prepare for the RS/6000 configuration, the following information is needed from the mainframe system programmers:

- The HCD or IOCP definition for defining a subchannel address as a 3174 link
- The NETID for the VTAM network which is usually found in the ATCSTR00 data set for MVS

  Section 7.5, “SNA Configuration Overview” on page 121 shows the VM datasets in more detail from the mainframe side, but in this chapter, we are more concerned with the RS/6000 setup.

  This very important dataset, with respect to our setup, has a full path name of SYS4.VTAMLST.ATCSTR00 and is shown below:

```
EDIT       SYS4.VTAMLST(ATCSTR00) - 01.36                Columns 00001 00072
000001 ************************************************************************
000002 **                                                                    **
000003 **       VTAM STARTUP OPTION - COMMON STARTUP PARAMETERS FOR:          **
000004 **                                                                    **
000005 **                              ALL DHAT MVS SYSTEMS                  **
000006 **                                                                    **
000007 **  NOTE:  ATCSTR00 IS THE DEFAULT VTAM START-UP OPTIONS MEMBER       **
000008 **         USED TO INITIALIZE A SPECIFIC VTAM DOMAIN AND IS ALWAYS    **
000009 **         PROCESSED.  IF A USER DEFINED "ATCSTRYY" IS SPECIFIED      **
000010 **         (WHERE "YY" IS ANY TWO ALPHANUMERIC DIGITS), THAT          **
000011 **         MEMBER IS PROCESSED BEFORE AND MERGED INTO THE ATCSTR00    **
000012 **         MEMBER.                                                    **
000013 **                                                                    **
000014 **                                                                    **
000015 *
000016 *   BUFFER DEFINITION POSITIONAL PARAMETERS
000017 *   (BASENO,BUFSIZE,SLOWPT,F,XPANNO,XPANPT,XPANLIM)
000018 *
000019  CRPLBUF=(150,,10,,30,30),   RPL-COPY - PAGEABLE                      X
000020  IOBUF=(200,3968,3,,90,36),   I/O BUFFERS - FIXED                     X
000021  LFBUF=(80,,2,,5,5),         LARGE GENERAL PURPOSE - FIXED            X
000022  LPBUF=(60,,1,,6,2),         LARGE GENERAL PURPOSE - PAGEABLE         X
000023  SFBUF=(30,,1,,5,5),         SMALL GENERAL PURPOSE - FIXED            X
000024  SPBUF=(30,,2,,8,4),         SMALL GENERAL PURPOSE - PAGEABLE         X
000025  CDRSCTI=3600,                    DYNAMIC CDRSC RETENTION  0-32767 480 0
000026  IOINT=28800,                 SUPPRESS PENDING MESSAGES 0-99999999 180 0
000027  MAXSUBA=31,                                                       15 0
000028  MSGMOD=NO,                  INCLUDE VTAM MODULE NAME IN MESSAGES   NO 0
000029  NETID=USIBM00,                 ** SNI - UNIQUE NETWORK IDENTIFIER  ** X
000030  PROMPT,                                                    PROMPT 0
000031  NOTRACE,TYPE=VTAM,                                            ** X
000032  PPOLOG=YES,                      ** VTAM COMMANDS TO NETWORK LOG   NO X
000033  HOSTSA=01,                                                       X
000034  SSCPID=0000,                                                     X
000035  SSCPNAME=TH0VTAM,                                                X
000036  HOSTPU=S02PUS,
000037  CONFIG=00,                                                       X
000038  NOPROMPT
****** **************************** Bottom of Data ****************************
```

Note the important NETID parameter for use in the configuration below.

- The datasets for the actual LU and PUs for the defined RS/6000.

  These were datasets H0LRS6KB and H0LRS6KD for BLKMUX and ESCON connectivity, respectively.

H0LRS6KB

```
****** **************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 ************************************************************************
000002 *                                                                      *
000003 *    LOCAL RS/6000
000004 *    ATTACHED TO 9121 - UCB ADDR C70
000005 *    USED BY AUSESC                                                    *
000006 *                                                                      *
000007 *  CHANGE LOG                                                          *
000008 *                                                                      *
000009 * 12/01/94  JAMES BOYKIN                                               *
000010 *           - CREATED FOR USE BY AUSESC ON LPAR 9121G                  *
000011 *                                                                      *
000012 ************************************************************************
000013 H0LRS6KB VBUILD TYPE=LOCAL
000014 *
000015 THOB     PU    CUADDR=C70,                                            X
000016                XID=YES,                                               X
000017                CPNAME=THOBCP,
000018                MAXBFRU=1,                                             X
000019                VPACING=7,                                             X
000020                PACING=7,                                              X
000021                DELAY=0.00,                                            X
000022                USSTAB=USSHATS,                                        X
000023                MODETAB=MTAWD,                                         X
000024                DLOGMOD=D3270MOQ
000025 THOBCP   LU    LOCADDR=00,DLOGMOD=LU62P
000026 THOB02   LU    LOCADDR=02
000027 THOB03   LU    LOCADDR=03
****** **************************** Bottom of Data ***************************
```

H0LRS6KD

```
EDIT       SYS4.VTAMLST(H0LRS6KD) - 01.00                Columns 00001 00072
Command ===>                                               Scroll ===> PAGE
****** **************************** Top of Data ******************************
==MSG> -Warning- The UNDO command is not available until you change
==MSG>           your edit profile using the command RECOVERY ON.
000001 ************************************************************************
000002 *                                                                      *
000003 *    LOCAL RS/6000
000004 *    ATTACHED TO 9121 - UCB ADDR 916
000005 *    USED BY AUSESC                                                    *
000006 *                                                                      *
000007 *  CHANGE LOG                                                          *
000008 *                                                                      *
000009 * 08/11/95  JAMES WENZLAFF                                             *
000010 *           - CREATED FOR USE BY AUSESC ON LPAR 9121G                  *
000011 *                                                                      *
000012 ************************************************************************
000013 H0LRS6KB VBUILD TYPE=LOCAL
000014 *
000015 THOD     PU    CUADDR=916,                                            X
000016                XID=YES,                                               X
000017                CPNAME=THODCP,                                         X
000018                MAXBFRU=1,                                             X
000019                VPACING=7,                                             X
000020                PACING=7,                                              X
000021                DELAY=0.00,                                            X
000022                USSTAB=USSHATS,                                        X
000023                MODETAB=MTAWD,                                         X
000024                DLOGMOD=D3270MOQ
000025 THODCP   LU    LOCADDR=00,DLOGMOD=LU62P
000026 THOD02   LU    LOCADDR=02
000027 THOD03   LU    LOCADDR=03
****** **************************** Bottom of Data ***************************
```

- The method for varying on and off the connectivity from the mainframe side. If this is dynamic, as in our case, the commands from the MVS console were:

```
To vary on

> D NET,ID=HOLRS6KB,ACT,ALL


To vary off:

> D NET,ID=HOLRS6KB,ACT,ALL


To check the connections:

> D NET,ID=HOLRS6KB,E
```

We had the luxury of the RS/6000 console being next to the MVS console; so testing was relatively easy. If this can be achieved at your site, it is advisable.

5.1.3 HCD definition for SNA

The details from The HCD gen listing that are appropriate are shown below:

```
                                  CHANNEL PATH DETAIL REPORT                    TIME: 15:47  DATE: 95-08-04  PAGE E-   8
PROCESSOR ID FG9121   TYPE 9121     MODEL 320      CONFIGURATION MODE: LPAR  TOKEN: FG9121   95-08-04 14:58:44 IODF     IODF80
                           --- SWITCH ----                                    UNIT ADDR                 UNIT
                           ID  PR  CU  DYN   --- CONTROL UNIT ---  CU-          RANGE    -- DEVICE --  ADDR   DEVICE
CHPID  TYPE    SIDE  MODE      PN  PN  ID   NUMBER  TYPE-MODEL    ADD  PROTOCOL  FROM  TO  NUMBER,RANGE  START  TYPE-MODEL
-----  ------  ----  ----  ----------  ---  ------  ------------  ---  --------  ----  --  ------------  -----  -------------
 0C    BL            DED                 00C1  3088                   S4        40   4F   0C40,16       40    3088
                                         00C2  RS6K                   S4        50   5F   0C50,16       50    RS6K
                                   ----------------------------------------------------------------------------------------
                                         0C60  3174                   S         60   6F   0C60,16       60    3278-2
                                   ----------------------------------------------------------------------------------------
                                         0C70  3174                   S         70   77   0C70,8        70    3174
                                   ----------------------------------------------------------------------------------------
                                         0C71  3791L                  S         78   7F   0C78,8        78    3791L
-------------------------------------------------------------------------------------------------------------------------------
  2F    CNC           REC                 0184  SCTC            1              10   11   0910,2        10    SCTC
                                   ----------------------------------------------------------------------------------------
                                         0185  SCTC            2              12   13   0912,2        12    SCTC
                                   ----------------------------------------------------------------------------------------
                                         0186  SCTC            3              14   15   0914.2        14    SCTC
                                   ----------------------------------------------------------------------------------------
                                         0187  3174            4   S          16   16   0916          16    3174
                                   ----------------------------------------------------------------------------------------
                                         0188  SCTC            5              18   1F   0918,8        18    SCTC

ESOTERIC NAMES

-                                          AFFINITY  ALLOCATION   ASSOCIATED
  NAME        NAME TYPE  VIO  TOKEN  PREF   INDEX    DEVICE TYPE  GENERICS                DEVICE NUMBER LIST
  ---------   ---------  ---  -----  ----  --------  -----------  ----------  ----------------------------------------------
  ----------------------------------------------------------------------------------------------------------------------------
  RS6K17      ESOTERIC             8300    FFFF      00004101     SCTC        0912 0913 0914 0915
  ----------------------------------------------------------------------------------------------------------------------------
  RS6K43      ESOTERIC             8389    FFFF      00004103     RS6K        0C50 0C51 0C52 0C53 0C54 0C55 0C56 0C57 0C58 0C59 0C5A
                                                                              0C5B 0C5C 0C5D 0C5E 0C5F
```

Figure 23. HCD Gen Listing for SNA Configurations

This is a section of the channel path detailed report. The column headings specify BL for Block Multiplexer and CNC for ESCON. Notice the parameters for the Device Number and Range, the device type of 3174, the UNIT ADDR RANGE, the CUADD, and whether it is a CVC- or CNC-type connection. CVC is through an ESCON converter; CNC is direct attached. These are explained in detail later on and used as parameters in the RS/6000 setup.

5.1.3.1 Datasets H0LRS6KB and H0LRS6KD for BLKMUX and ESCON

The two data sets for either channel are equivalent, apart from the PU and LU names. The PU and LU names are needed for parameters in the subchannel definitions that are described below. In configuring SNA, the NETID and the subchannel definitions are required.

5.2 ESCON and Block Multiplexer Subchannel Configuration on RS/6000

The detailed subchannel configuration is in Chapter 2, “MVS TCP/IP V3R1M0 Overview” on page 5 and in Chapter 7, “VM Host Connection” on page 93. We only touch on specific differences in this section for the SNA subchannel. There is no difference in complexity between ESCON and BLKMUX configuration; so we will show the ESCON configuration as an example for our discussion.

Before configuring, stop SNA with the following commands:

```
sna -stop sna
```

Bring down the TCP/IP channel interfaces. In our case, we had four separate adapters.

```
ifconfig es0 down detach
ifconfig es1 down detach
ifconfig ca0 down detach
ifconfig ca1 down detach
```

Note: These are the TCP/IP interfaces, and they may be up and in use.

Run the rmdev command on the existing channel devices.

```
rmdev -l escon0
rmdev -l escon1
rmdev -l cat0
rmdev -l cat1
```

Not doing this results in ODM corruption which involves a reboot, at best, and a reinstall of the product, at worst.

5.2.1 ESCON Add a Subchannel Definition for SNA

The subchannels for SNA must be defined to AIX.

To do this, issue the command:

```
smitty escaaddSubd
```

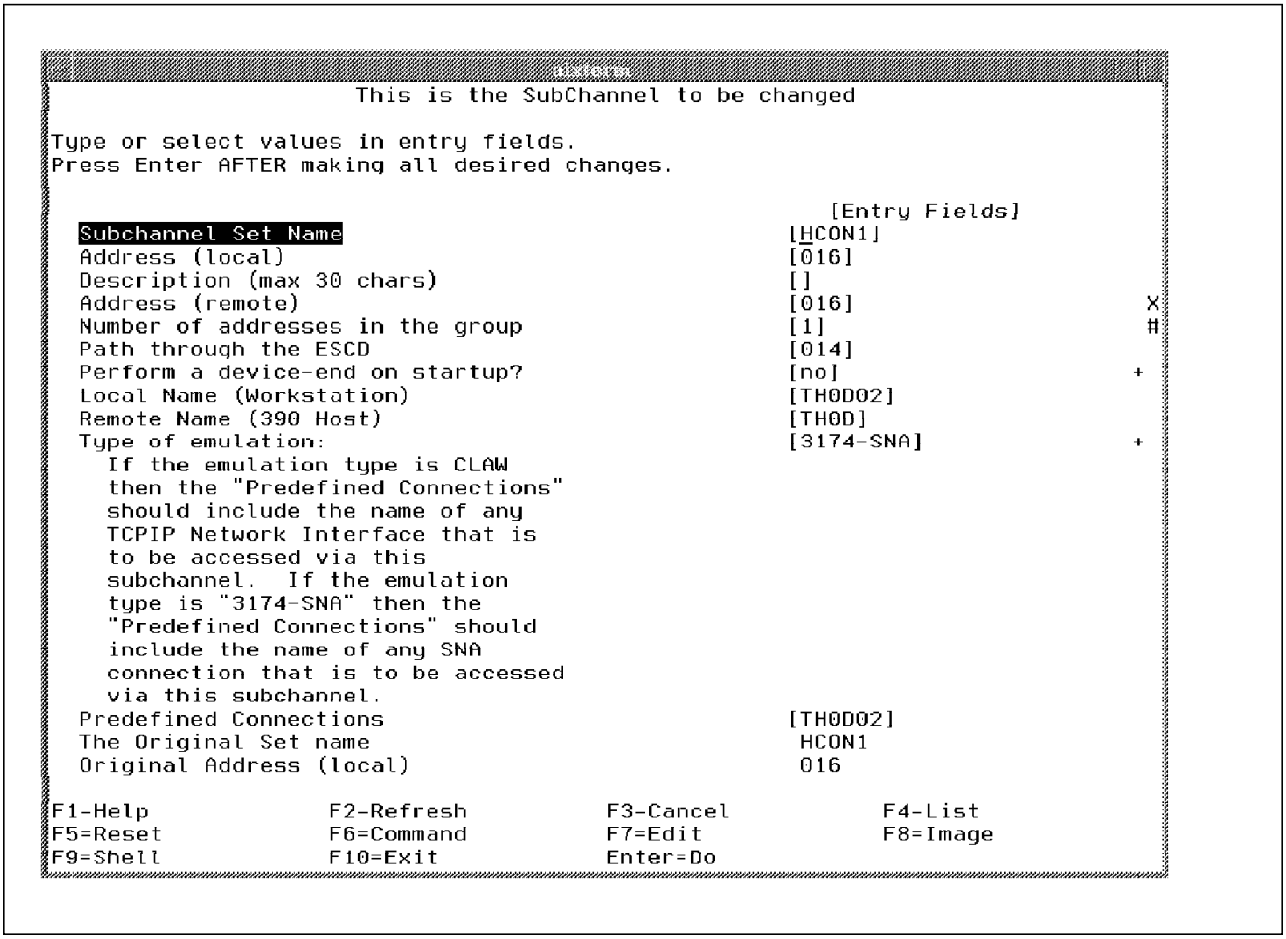


Figure 24. ESCON Add a Subchannel

- **Subchannel Set Name**

  The Subchannel Set Name of HCON1 is only for reference. When set up the HCON definition, we called this subchannel. Please see the next section for SNA and HCON quick set up.

- **Address (local and remote)**

  The local address is simply an internal reference to the remote address, which is the important definition as it refers to the subchannel definition from the HCD gen shown above. This is the UNIT ADDR value associated with the 3174 definition. Note the device number and range in the HCD gen are usually the same, but they **do not** have to be. The imagined case of a customer with vast banks of DASD who potentially runs out of numbers for definitions may result in different device numbers and ranges.

- **The path through the Extended System Contents Directory (ESCD)**

  This parameter is the most complex; it usually comes in two parts. If there is no ESCD between the the RS/6000 and no CUADD parameter is specified, then the default value in the field should be taken. If there is an ESCD (as in our case) or a CUADD parameter is specified, then the value entered consists of two parts:

  For the MVS system, the ESCON is direct attached, and the system default is that the first two characters of this definition are 01. However, a CUADD

parameter of 4 was specified, and this is added as the last digit in the path definition, giving a total path definition of 014.

For the VM system, as described in 3.1.5, “Setting Up the ESCD for VM TCP/IP over ESCON” on page 30, the first two hexadecimal digits are the port number of the ESCD to which the host channel is connected, for example C4. Since no CUADD parameter is specified, the Path through the ESCD is C40. If two directors are used, the host port address of the dynamic ESCD should be used for the first two digits.

- **Perform a device-end on startup?**

  For SNA Channel for AIX, select **no** so that no device end should be issued by the RS/6000 to MVS. It is worth a discussion as to why SNA and TCP/IP need to be configured differently here:

  The “on startup” is the key. “On startup” means on startup of the device driver, not on startup of the SNA Link Station. Startup of the device driver is during the boot of AIX, many seconds or even minutes before SNA (on the RS/6000) and the SNA Link Station get started.

  When VTAM gets an unsolicited device-end, it restarts the PU by sending the first XID down. If that XID arrives before the SNA Link Station is ready (“Starting” state) to receive it, it gets lost. So, for SNA, we want to control the unsolicited device-end at the link start time, not at the device driver start-up time.

  TCP/IP is different in that it does reads and writes on different subchannels (even-odd pair); so an unsolicited device-end means to put a read on the read subchannel and wait for the validation from AIX TCP/IP.

  VTAM interprets unsolicited device-end as the notification to do a write (of the XID), and because TCP/IP interprets unsolicited device-end as the notification to do a read, they have to be configured differently. For TCP/IP (CLAW) subchannels, this needs to be yes; for SNA (3174) subchannel this needs to be no.

- **Local name**

  This is the local name associated with the subchannel; it was set up as the LU from the H0LRS6KD dataset with a value of TH0D02, as shown in Figure 24 on page 55.

- **Remote Name (390 Host)**

  This **must** match the VTAM PU definition as in the H0LRS6KD dataset.

- **Type of Emulation:**

  This **must** be the 3174 definition as shown in the HCD gen listing; ensure that the channel definition numbers above match up.

- **Predefined Connections**

  This must be equivalent to the SNA Link Station Profile we set up in the next section. SNA uses this value to reference the subchannel and after `rmdev` and `mkdev`, the value is picked up.

- **The Original Set Name and Original Address (local)**

  These are display only, but for reference, they should be set out as shown.

5.2.2 Defining the ESCON Fibre with SNA Subchannel

The ESCON fibre for use with SNA must be defined to AIX.

To do this, issue the command:

```
smit escaddFibre
```

Add the definition into the correct fibre as shown in Figure 11 on page 29. Note the definition is called HCON1, which gives a strong hint that we are going through SNA for our HCON definitions.

5.2.3 Defining the ESCON Network Interface with SNA Subchannel

The value specified for Predefined Connections for defining the subchannel above should automatically create a connection definition for the connection name.

5.3 SNA Quick Configuration for Channel Link Stations and HCON

In setting up SNA connectivity over channel using SNA Server/6000 to the mainframe, the methodology in the start-up sequences and editing of SNA need to be followed in menu fashion. We endeavor to take the UNIX specialist with limited SNA knowledge through the methodology to get to a 3270 screen via an RS/6000 link station, not simply by Telneting into the MVS TCP/IP.

Our end point was HCON SNA sessions working over the same channel as the TCP/IP applications, thus configuring the dual device driver as well.

5.3.1 Setting Up SNA Using the SMIT Panels

There are four main areas to define:

- Initial node set up
- The SNA DLC Profile
- Defining the SNA DLC Profiles
- Defining the LU2 session

5.3.1.1 Initial Node Set Up

This setup is for the MVS network. Note TH0D02 is the same as in Figure 24 on page 55. A different NETID is specified for the VM network.

The initial node setup **for the Austin site** required the command sequence:

```
smitty sna
>  Configure SNA Profiles
>  Initial Node Setup
>  PF4 to expand the choices and choose the channel option
```

Figure 25 on page 58 shows the Initial Node Setup screen.

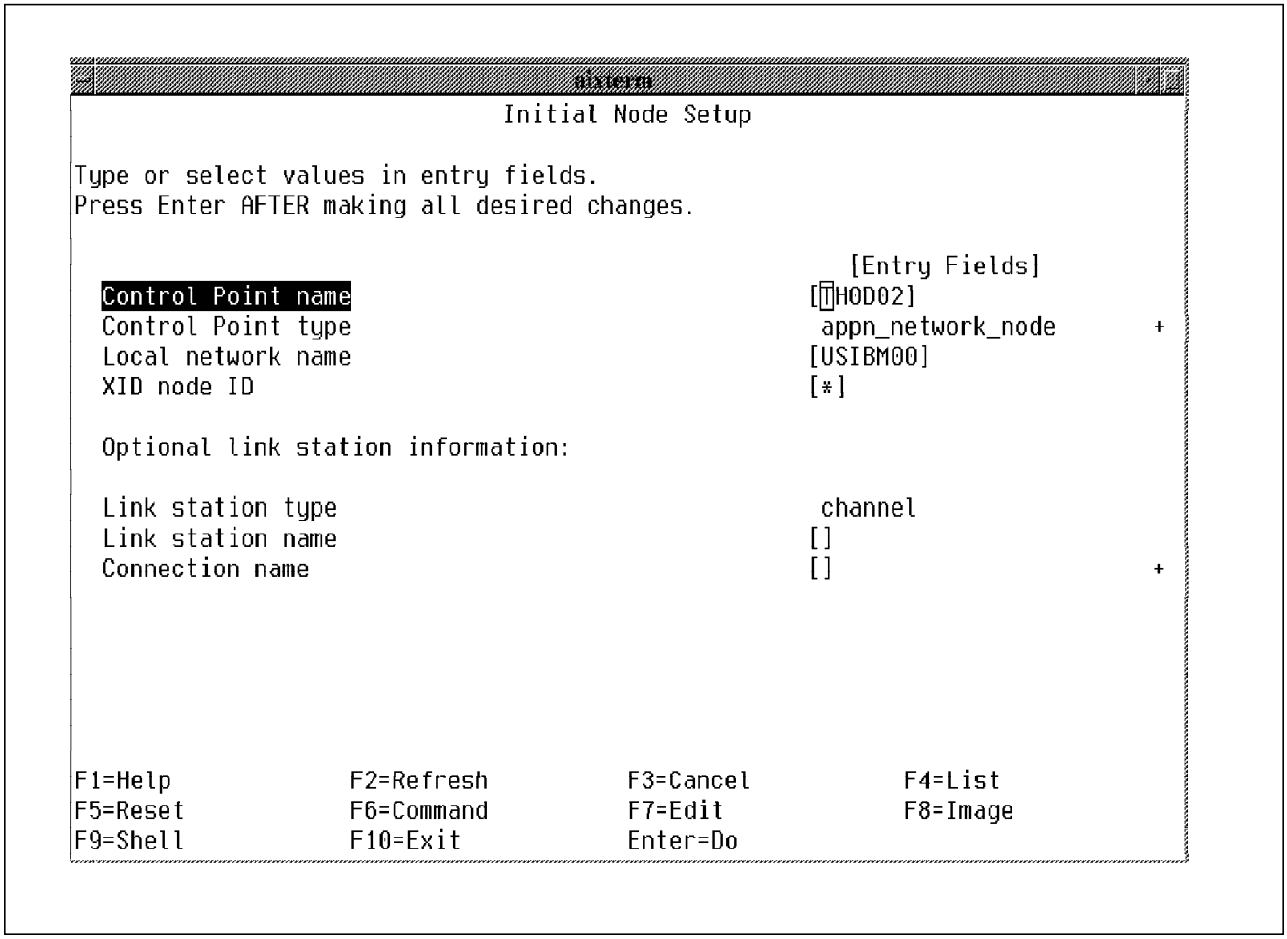


Figure 25. Initial Node Setup

- **Control Point name**

  The predefined definition from the subchannel network is also the Control Point Name, TH0D02, as defined above.

- **Control Point type**

  This is a network node.

- **Local network name**

  The Local network name is taken from the MVS dataset SYS4.VTAMLST(ATCSTR00) for a NETID value of USIBM00.

- **The Link station type**

  This is channel.

This defines the Initial node setup.

5.3.1.2 The SNA DLC Profile

The SNA DLC Profile must be defined to AIX.

To do this, the easiest way is to use the main SMITTY fast path with the command:

```
smitty _snachannellinkmk
```

Figure 26 on page 59 shows the screen brought up by the SMITTY fast path that was just mentioned.

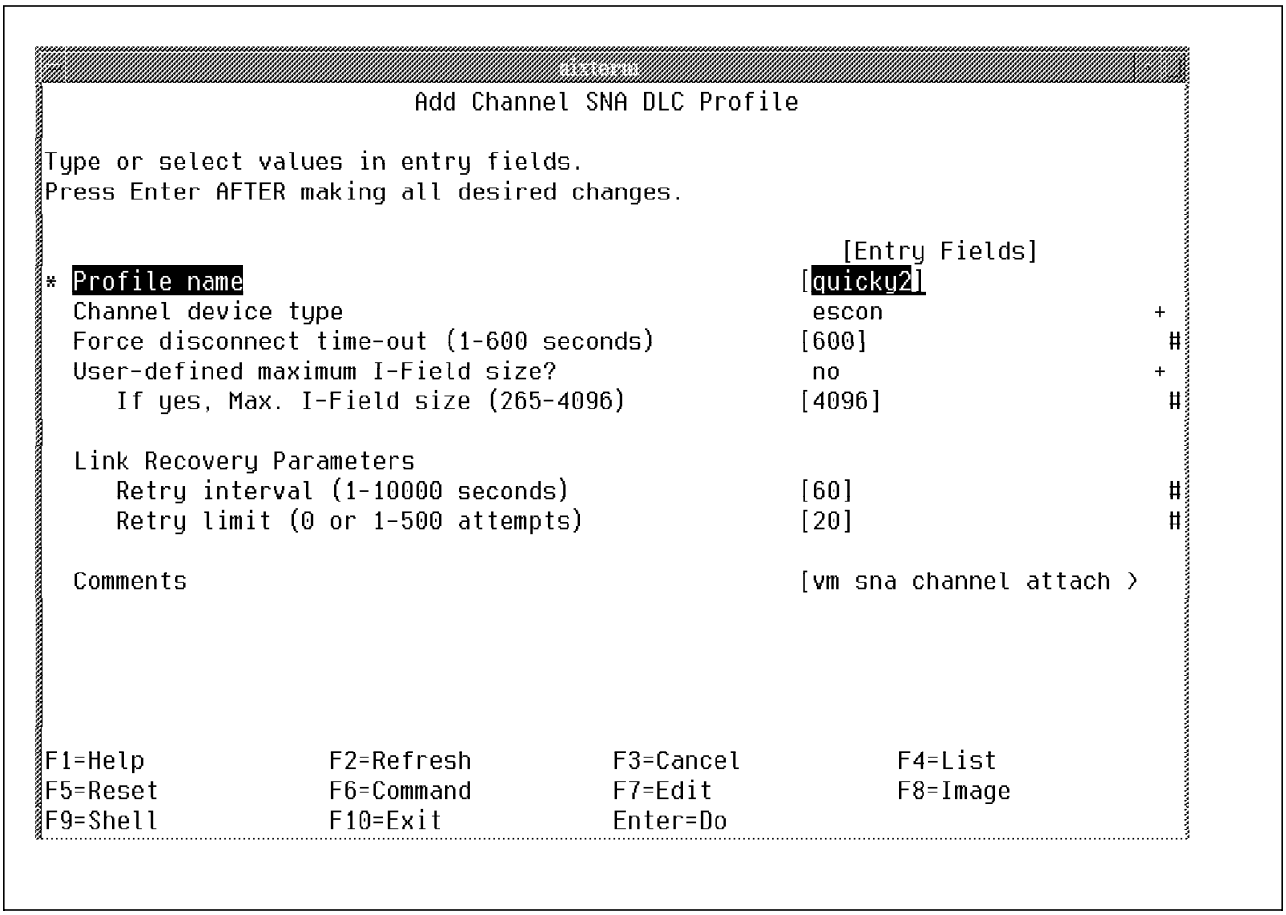


Figure 26. Add SNA DLC Profile

This has taken you through the following SMIT screens:

```
1. SMIT SNA
2. Configure SNA Profiles
3. Advanced configuration
4. Links
5. Channel
6. Channel SNA DLC
7. Add a profile
```

The SNA DLC Profile name is referred to by the Link Station Profile, which we are about to configure. It points the Link Station towards the correct channel device type. Leave the other parameters as we display them above.

5.3.1.3 Defining the SNA DLC Profiles

The SNA DLC Profiles must be defined to AIX.

The easiest way to do this is to use the main SMITTY fast path with the command:

```
smitty _snachannelattcmk
```

```
                                aixterm
                    Change/Show Channel Link Station Profile

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
  Current profile name                                THOD02
  New profile name                                    []
  Use Control Point's XID node ID?                    yes                +
     If no, XID node ID                               [*]
* SNA DLC Profile name                                [quicky2]          +
  Connection name                                     [THOD02]           +
  Stop link station on inactivity?                    no                 +
     If yes, Inactivity time-out (0-10 minutes)       [10]                #
  LU address registration?                            no                 +
     If yes,
        LU Address Registration Profile name          []                 +
  Trace link?                                         no                 +
     If yes, Trace size                               long               +

  Adjacent Node Identification Parameters
     Verify adjacent node?                            no                 +
     Network ID of adjacent node                      []
     CP name of adjacent node                         []
     XID node ID of adjacent node (LEN node only)     [*]
     Node type of adjacent node                       learn              +

  Link Activation Parameters
     Solicit SSCP sessions?                           yes                +
     Activate link station at SNA start up?           yes                +
     Activate on demand?                              no                 +
     CP-CP sessions supported?                        yes                +
     If yes,
        Adjacent network node preferred server?       no                 +
     Partner required to support CP-CP sessions?      no                 +
     Initial TG number (0-20)                         [0]                 #

  Restart Parameters
     Restart on normal deactivation?                  yes                +
     Restart on abnormal deactivation?                yes                +

  Transmission Group COS Characteristics
     Effective capacity                               [393216000]         #
     Cost per connect time                            [128]               #
     Cost per byte                                    [128]               #
     Security                                         nonsecure          +
     Propagation delay                                minimum            +
     User-defined 1                                   [128]               #
     User-defined 2                                   [128]               #
     User-defined 3                                   [128]               #

  Comments                                            []


F1=Help              F2=Refresh           F3=Cancel            F4=List
F5=Reset             F6=Command           F7=Edit              F8=Image
F9=Shell             F10=Exit             Enter=Do
```

Figure 27. Add Channel Link Station

This has taken you through the following SMIT screens:

```
1. SMTI SNA
2. Configure SNA Profiles
3. Advanced configuration
4. Links
5. Channel
6. Channel Link Station
7. Add a profile
```

Most of the parameters above should be taken as default, but a couple are worth commenting on:

- **Connection Name**

  This name must be the same as the Predefined Connections field, TH0D02, set up in the ESCON subchannel definition.

- **SNA DLC Profile name**

  This should refer to the SNA DLC Profile created above.

The other parameters should be set to the ones above. For more information, please refer to the AIX SNA Server/6000 Channel Connectivity Feature User's Guide for Version 2.1.

5.3.1.4 Configuring the LU2 Session

To configure the LU2 session, enter:

```
smit sna
>  configure SNA Profiles
>  advanced configuration
>  Sessions
>  LU 2
>  Add a Profile
```

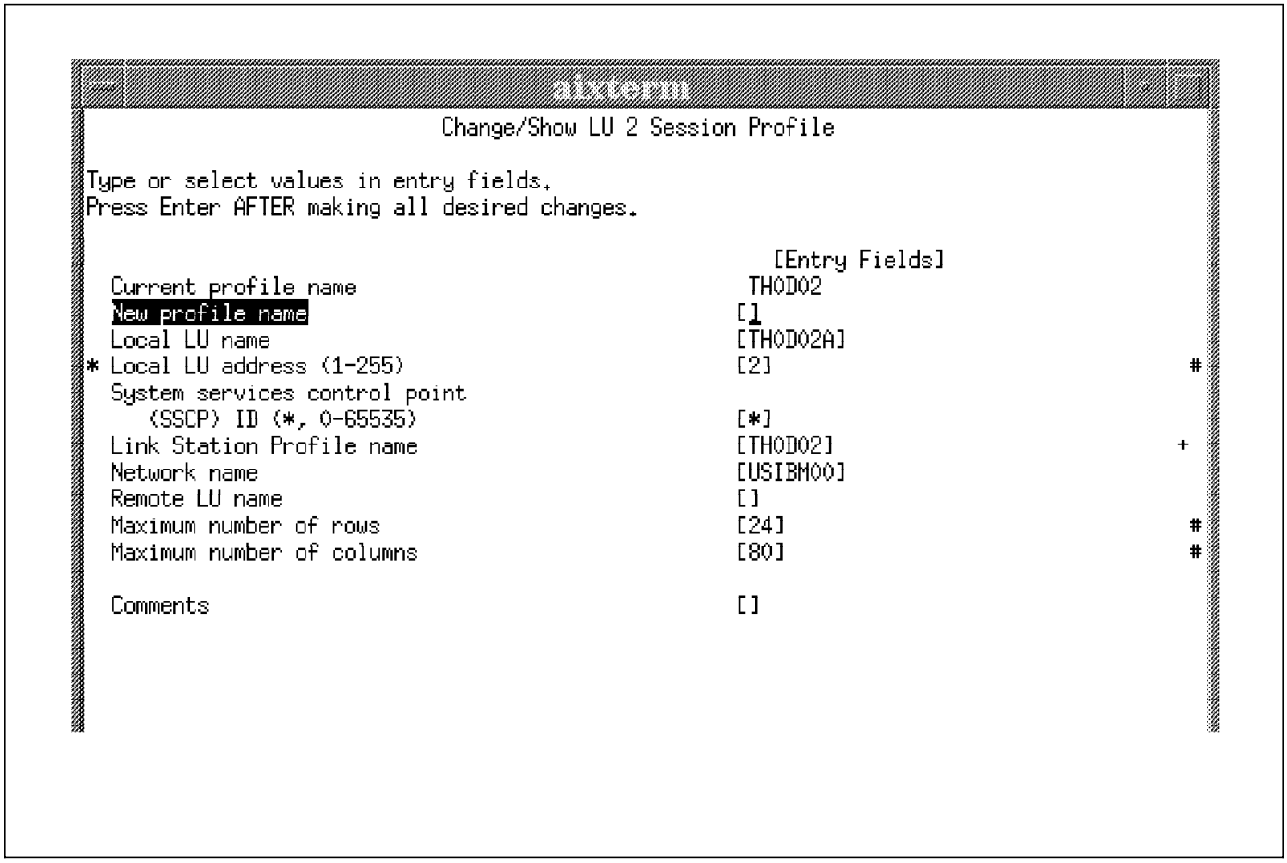


Figure 28. Configuring the LU2 session

The Local LU name is for reference purposes only, but the Link Station Profile name and the Network name **must** be those from the SNA definitions (and the predefined definition in the subchannel definition) and the host dataset, respectively.

5.3.2 Starting Up the Link

Once the SNA definitions and the subchannel definitions are configured, proceed to redefine the devices to ensure the SNA subsystem obtains the parameters about the SNA Link Profile and DLC Profile via the commands:

```
rmdev -l escon0
mkdev -l escon0
```

To activate the RS/6000 SNA Link Station, start SNA, and by default, it will start the link. If SNA is already running, issue the command:

```
sna -start link_station -profile linkstationProfileName
```

Only **after** the link is in a starting state can you vary on the VTAM PU, as described above in the host parameters section.

Stop the process starting with the VTAM PU first, and then stop the AIX process. If this method was not followed exactly, we experienced complete channel hang, and it was difficult to retrieve the connection without rebooting.

5.3.2.1 Verifying the Link is Active

Check the link via the command line by entering the following command:

```
sna -d 1
```

This will display the *active* and *starting* links.

5.3.2.2 Display the Sessions

You can display the SNA sessions with the following command:

```
sna -d s123
```

5.3.3 Setting Up HCON

The only difference for the HCON session, apart from normal use, is to ensure during the set up that the SNA logical connection prefix or profile is set up. In our case, TH0D02 was defined as the LU and the Link Station Profile name. Issue the command:

```
> smitty hcon
```

and add the respective SNA profile.

Start the HCON session with the command:

```
> e789
```

Chapter 6. IBM Client Input Output/Sockets (CLIO)

With the growth in RISC/6000 performance and SP2 products, applications have migrated to, and have been developed for, AIX architectures that need to move large amounts of data between the mainframe environment and the AIX platform. The CLIO application satisfies this need for communication between mainframe MVS and RS/6000s.

6.1 CLIO Overview

CLIO is a set of commands and APIs for high-speed communication and for accessing tape devices between a host MVS mainframe and RS/6000s. CLIO solves many of the problems associated with moving data between the architectures. CLIO is relatively easy to install. The mainframe is loaded, and then the AIX program product is downloaded over TCP/IP. The configuration has been carefully thought through to facilitate ease of understanding in terms of the mainframe system programmer's view point and the RS/6000 specialist's approach to MVS datasets.

CLIO provides the following functions:

- Fast transfer of data using:
  - A file transfer command called `clftp`
  - A pipe link command called `clplink`
  - A sockets interface
  - A communications manager programming interface
  - A distributed processing interface
- A tape server that can be accessed by the CLIO tape interface or via a pseudo device driver

We have used CLIO in an established test network to `clftp` large files in BIN, ASCII and database (DB2/6000) formats in both directions. In this chapter, we explain to the reader the methods, performance implications and use of the multiple tools provided with the software that show how the systems are performing. This is both to and from MVS disk and tape.

6.1.1 CLIO User's Guide

The CLIO User's Guide is provided on the CLIO tape. It can be printed and includes an excellent step-by-step methodology for understanding, configuring and using CLIOs in a standard environment. When necessary, we summarize some of the pages to specifically explain the Austin test setup and provide tailored explanations for use in a customer environment.

6.1.2 CLIO Throughput

We have not used the facilities of BatchPipes/MVS and specific tuning for the parallel environment mainly because of a lack of hardware and time. We do, however, explain briefly where these would be used, the performance gains that could be achieved and provide some customer examples.

There are constantly questions from customers and IBMers alike about the achievable bandwidth when using CLIO and channel products. It has become apparent that there is no standard answer, and the reasons are relatively straight forward. Published CLIO figures have always been for memory-to-memory transfers. It is likely that one of the external peripherals will, in practice, usually be a gating factor on the observed transfer rates. In this respect, it can be shown that the only way to get full benefit (or maximum throughput) of a channel is to do multiple transfers simultaneously.

Continuing with this conversation, we can carefully mention performance implications surrounding the complete data pipe from MVS disk (or tape) to AIX disk (or tape). However, one fact should be emphasized up front. The CLIO software does allow a memory-to-memory exchange between the mainframe and the RS/6000 of very close to the channel bandwidth. However, it is likely that the rate-determine step in the system concerns factors outside the scope of CLIO tuning. These are the relevant disk architectures and the CPU implications on both sides. It is possible, although quite expensive, to obtain a data pipe bandwidth approaching ESCON speeds by using a lot of CPU power and configuring the DASD architecture via striping and RAID. We quote such a setup in the throughput section, 6.4.3, “CLFTP Results and Throughput Considerations” on page 81. When configuring the customer setup, the sustained bandwidth possible is, of course, determined by the slowest step in the chain. In the Austin setup, this was quite obviously the RS/6000 disk as we used only the 2 GB SCSI 2 internal disk and drove it as fast as it could read or write in a sustained environment. Again, we emphasize the word sustained because this is not the burst speed of the disk or the speed obtained on a reorganized fresh environment but what should be seen in an average non-tuned customer environment. Use of SSA disks or a more appropriate architecture for downloading a much larger volume of data would have given a much larger data rate.

CLIO Fast Transfer

The `clftp` command is similar to FTP. It facilitates the use of BLKMUX and ESCON channels without programming and has similar interface to FTP in terms of help and directory access. We show examples of its use and the underlying testing features in the results section below.

The Tape Server

After set up, the CLIO tape pseudo device driver allowed us to use the AIX `tar` command to back up DB2/6000 database files on the MVS tape system. We did not use the programming interface for our work, but we comment below on specific customer examples for writing to the API. There are specific MVS issues surrounding the designation of tape drives, and we also comment on these in the configuration section.

6.1.3 A Brief View of How CLIO Works

Figure 29 on page 67 shows how CLIO communicates:

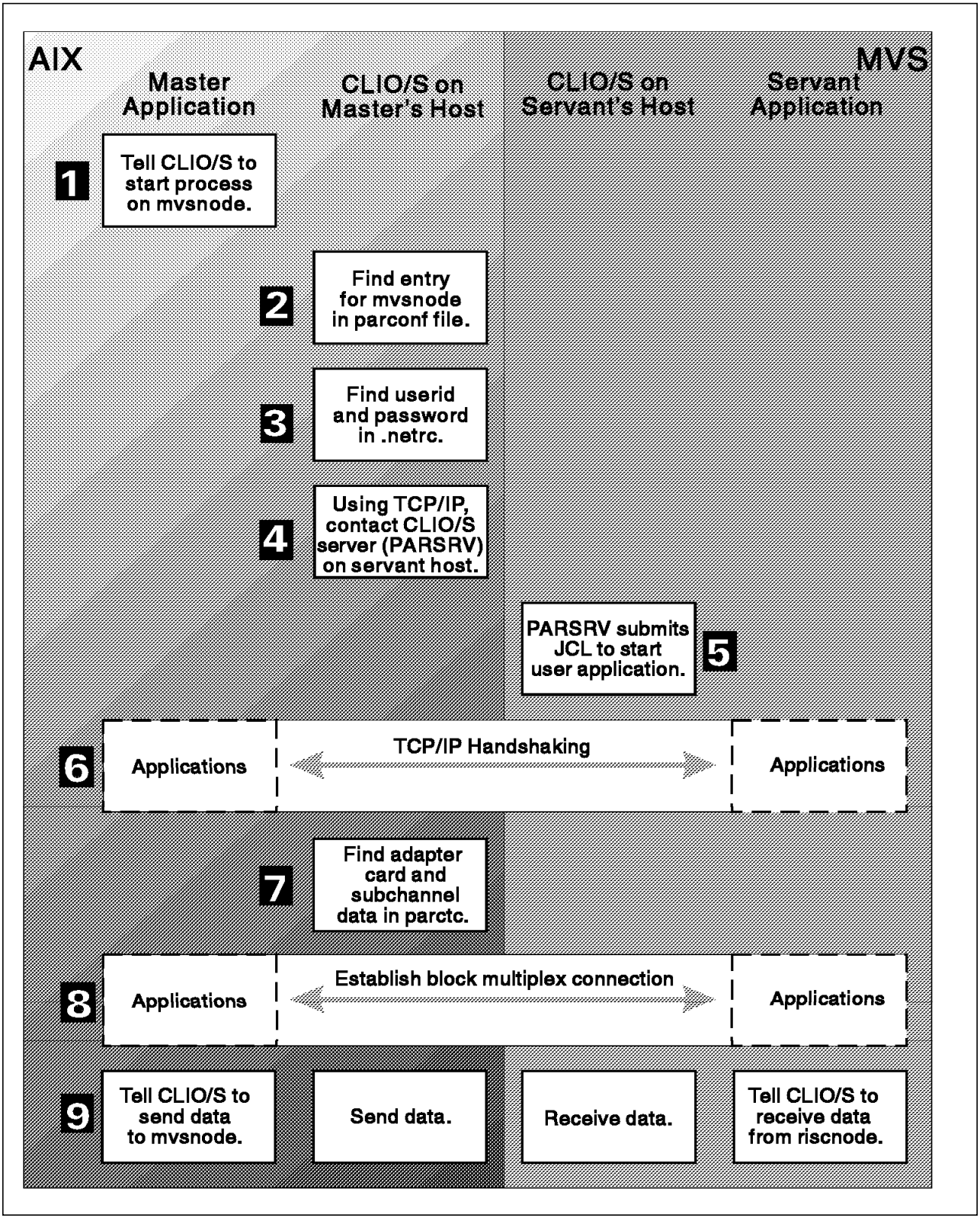


Figure 29. How CLIO Communicates

A deamon called *parsrv* handles all the interfaces (apart from the sockets interface which is more complex) and is started on both the mainframe and the RS/6000 systems. The system consists of a CLIO client and a CLIO master; it can either be started from the RS/6000 or from the MVS host. When a CLIO client

issues a command, for instance via `clftp`, to a CLIO master it does so via parsrv over TCP/IP. The initial negotiation or “hand shaking” between the two systems occurs through TCP/IP. The parsrv deamon then creates a high-speed link between the RS/6000 and the MVS host and starts the servant program. Depending on which CLIO function is envoked, subroutines cause parsrv to start the appropriate servant program. For example, for `clftp`, it instructs parsrv on the remote host to start CLIO′s internal file transfer server, called FCFSERV. FCFSERV takes over and transfers the file(s) as requested by the `clftp` command using a `get` or `put` similar to AIX FTP at the `clftp` prompt. When you quit from `clftp`, parsrv closes the channel and returns you to the command prompt.

The following steps correspond to the steps in Figure 29 on page 67.

1. The application on the RS/6000, the master, calls CLIO to start a process on the MVS node
2. CLIO looks in the PARCONF configuration file to find out where the MVS node is, what type of communication links to connect and what operating system is running.
3. CLIO looks in the NETRC file to determine which user ID starts the process.
4. Via TCP/IP, the parsrv on the RS/6000 communicates with the parsrv running on the MVS node asking it to start the servant application.
5. The parsrv deamon creates the JCL to start the requested application on MVS (an example in the next section).
6. The application on the MVS servant and the application at the AIX master establish TCP/IP communication.
7. CLIO looks up the adapter and subchannel information specified in the parctc file.
8. The applications establish a fast transfer connection for exchanging data.
9. The two processes can exchange data via the communication commands, for instance with CLFTP `put` and `get`.

The three files: parconf, .netrc and parctc are very important configuration files and **must** be configured exactly for the complete process to work. We comment on these below.

6.2 CLIO Installation on RS/6000 and MVS Systems

CLIO is supplied in MVS tape format, and part of the overall installation is downloading the AIX part by using FTP over a TCP/IP link between the two systems. Part of the process of installation is channel configuration, and you will therefore need to plan carefully with the mainframe systems programmers.

6.2.1 Planning Information Required

Prior to install, it is necessary to obtain the following configuration information:

- Mainframe Channel ID (CHPID)
- Unit address and range
- Device addresses for the mainframe subchannels (and type)
- Link address for the ESCON Director (if ESCON)
- ESCON Director ports to the channel adapter (if one is to be used)

- Specific customer setup for TCP/IP files on MVS
- Specific addresses associated for CLIO (see parctc file later)
- RS/6000 and mainframe nodenames
- RS/6000 and mainframe IP addresses

6.2.2 Installation Requirements

In addition to this section, please refer to the Program Directory for IBM CLIO that is supplied with the program product.

6.2.2.1 Mainframe Considerations

The system requirements for running CLIO are:

- MVS/ESA Release 4.2.0 and above (program number 5695-047 for JES2 or 5695-048 for JES3)
- System/390 processor
- Storage requirements shown in the Program Directory, page 9 and 10
- IBM C/370 Run-Time Library Version 2.1 or higher (program number 5688-188)
- One of the following
  - IBM PL/1 Library Version 2.3 or higher (program number 5669-911)
  - IBM SAA AD/CYCLE language environment/370 (program number 5688-198)
- IBM TCP/IP Version 2.2.1 or higher (program number 5735-HAL)

6.2.2.2 RS/6000 Considerations

- Machine type 7009 Model C10, 7012, 7013, or 7015 (as in this work) with one microchannel slot for Block Multiplexer Channel Adapter (BMCA) or two for ESCON.
- IBM 9076 Scalable POWERparallel Systems SP1 or SP2

BMCA requirements

- BMCA card (feature 2755)
- BMCA cable (feature 2757)
- BMCA Interface assembly (feature 2758)

SP2 or SP1 requirements

- BMCA card (feature 2755)
- BMCA cable (feature 2753)
- Block Multiplexer Node-Node Channel Adapter cable (feature code 2752) if more than one card is being daisy chained in a 9076 frame

ESCON requirements for RS/6000 or IBM 9076

- ESCON Channel Adapter card (feature code 2756)
- ESCON Channel Adapter cable
- Software required
  - AIX 3.2.5

–

The following maintenance is required on the RS/6000 workstation where an ESCON adapter is used:

- Fix for Authorized Program Analysis Report (APAR) IX39362 which is an update on the ESCON device driver
- EC C74216D. Update on the ESCON microcode and fixes data loss and corruption problems

To verify that you have the correct hardware and software, issue the command to verify the ESCON level: `lslpp -h esc*.*` The resulting output should be equivalent to:

```
Name
-----------------
Fix Id  Release         Status     Action     Date       Time   User Name

------ --------------- ---------- ---------- ---------- -------- --------

Path  /usr/lib/objrepos
escon.cuu
       03.02.0000.0000 COMPLETE   COMMIT     12/13/94   16:14:11 root

escon.usr
         03.02.0000.0000 COMPLETE   COMMIT     05/03/95   15:18:17 root
U 422416 03.02.0000.0000 COMPLETE   COMMIT     05/03/95   15:18:18 root
U 491144 03.02.0000.0000 COMPLETE   COMMIT     05/03/95   15:18:19 root


Path  /etc/objrepos
escondiag.obj
       03.02.0000.0000 COMPLETE   APPLY      03/22/95   10:58:09  root
```

If the PCA (BLKMUX) card is required, enter the command:

```
ls -l /etc/microcode/fe92*
```

The response includes the microcode level, and it should be fe92.00.03 or higher. Our system response was:

```
12834 Aug 21 10:14 /etc/microcode/fe92d.00.00
```

6.2.3 Customization of the Main MVS Datasets Used by CLIO Under MVS

We followed the install procedure to the letter. The time taken is a couple of hours to edit some of the JCL. We installed the base product into a directory with a high-level qualifier of CLIO, for ease of use. The default is SYS1. When comparing our information below with the install guide, replace “CLIO” with “SYS1.”

After the installation of the base product, there are a number of files that need to be customized. They are:

- FCFDEFS
- FCFREXX
- FCFPARSR

Table 2 on page 71 is a summary of the functions of the MVS datasets used under CLIO:

Table 2. Main MVS Datasets Used by CLIO on Austin System

Default File Name	File name for Austin MVS System	Purpose
FCFPARSR	PARSV2	Starts parsrv from SYS1.PROCLIB The DD statement specifies file CLIO.SFCFSAMP(MARTDEFS)
CLIO.SFCFSAMP(FCFDEFS)	MARTDEFS	Controls JCL generation for PARSV2
CLIO.SFCFSAMP(FCFREXX)	CLIO.SFCFSAMP(FCFREXX)	Controls JCL generation for PARSV2

The User's Guide refers to two steps:

1. We edited member FCFPARSR of CLIO.PROCLIB and copied it to PARSV2 making it the executable for starting the parsrv deamon on the mainframe. We updated the datasets referenced in the JCL to correspond to those in our installation and uncommented STEPLIB DD cards as appropriate for our case, as shown in Figure 30

```
//FCFPARSR EXEC PGM=FCFPARSR
//STEPLIB  DD   DISP=SHR,DSN=CLIO.SFCFLOAD DATA S                        ET
//         DD   DISP=SHR,DSN=SYS1.SEDCLINK    C/370 OR LE/370 RUNTIME
//*        DD   DISP=SHR,DSN=SYS1.SEQALINK    INSPECT DATASET, OPTIONAL
//*        DD   DISP=SHR,DSN=SYS1.SAMPRUN2 VSPASCAL RUNTIME
//SYSPRINT DD   SYSOUT=4,DCB=BLKSIZE=3429
//SYSTSPRT DD   SYSOUT=4,DCB=BLKSIZE=3429
//INSPLOG  DD   SYSOUT=4,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//DDTERM   DD   DISP=SHR,DSN=CLIO.SFCFSAMP(FCFDDT)
//*DEFAULTS DD   DISP=SHR,DSN=CLIO.SFCFSAMP(FCFDEFS)
//REXXJCL  DD   DISP=SHR,DSN=CLIO.SFCFSAMP             (FCFREXX)
//*                                        REXX JCL GENERATION METHOD
```

Figure 30. SYS1.PROCLIB(PARSV2)

2. Again, the User's Guide suggested that we perform **EITHER** step a or b. We actually did both in separate configurations, but step b is recommended.

   a. FCFDEFS of dataset CLIO.SFCFSAMP. We copied it to MARTDEFS. MARTDEFS controlled the JCL generation for the PARSV2 program of CLIO which controls the servant process under MVS. The details are shown in Appendix A, "TCPIP and CLIO Configuration Files" on page 137. The RCFREXX DD statement should be commented out. If this card is in the FCFPARSR PROC, then CLIO uses it instead of the FCFDEFS file.

   b. Edit member RCFREXX. There are useful comment lines advising one of the correct statements to change. The relevant changed part of this file is shown in Appendix A, "TCPIP and CLIO Configuration Files" on page 137.

Once these files have been edited, we submitted the Installation Verification Procedure (IVP), ensuring that the return codes were good and that we had no JCL errors.

6.2.4 Downloading and Building the RS/6000 Code

Two files need to be downloaded to the RS/6000, one for CLIO fast transfer and one for the CLIO tape server feature. If you have CLIO installed already, it is recommended that you copy and save three files.

```
/etc/parconf
/etc/netrc
/etc/parctc
```

The two files for downloading are in compressed *installp* format and are named FCFFIO and FCFTAPE in the CLIO.SFCSSAMP dataset. The default location is SYS1.SFCSSAMP, but we loaded in a CLIO high-level qualifier to makes things a little clearer. We downloaded them using FTP on the RS/6000.

```
#ftp MVSESC     (where MVSESC is the hostname of the MVS machine )

ftp> bin
ftp> get 'clio.sfcsamp(fcffio)' /usr/sys/inst.images/fcffio.Z
```

and then

```
ftp> get 'clio.sfcsamp(fcffio)' /usr/sys/inst.images/fcftape.Z
ftp> quit
#
```

Uncompress the files

```
#uncompress /usr/sys/inst.images/fcftape.Z

#uncompress /usr/sys/inst.images/fcffio.Z
```

Then follow normal install procedure:

```
#cd /usr/sys/inst.images

#inutoc
```

Then do a normal install of the products via:

```
smitty installp
```

Choose, of course, the fcffio and fcftape program products. The install takes a relatively short time.

6.3 CLIO Configuration on RS/6000 and MVS Systems

There are a number of tasks that we had to do:

- Configuring the CTCs
- Creating and customizing the CLIO configuration files
- Method of starting parsrv
- CLIO environmental variables
- Using CLIO striping
- Using CLIO and SNA on the same adapter

There are a number of other functions mentioned in the User's Guide, but we did not test or configure them because they were not needed in our installation. They were:

- Running multiple parsrv
- Ensuring password security
- Enabling TSO access to `clftp` and `clplink` commands

They may well be valuable functions at your site, and we refer you to the User's Guide for more details.

6.3.1 Configuring the CTCs

CLIO's main reason for existence is to take advantage of channel architecture; so we configured CLIO over ESCON and BLKMUX. Multiple subchannel definitions were configured over one channel link. At the time of this redbook was written, we were also running CLIO over multiple ESCON channels with AIX 4.1 to test bandwidth. We comment on this in the Results section.

As the initial CLIO handshake is over TCP/IP at least two CLAW TCP/IP subchannels must be available to the application. Figure 31 shows the relevant sections of our HCD listing for CLIO over the ESCON connection. We refer to these in the the definition of the parctc file later.

```
                                CHANNEL PATH DETAIL REPORT                    TIME: 15:47  DATE: 95-08-04  PAGE E-   8
PROCESSOR ID FG9121   TYPE 9121     MODEL 320      CONFIGURATION MODE: LPAR  TOKEN: FG9121   95-08-04 14:58:44 IODF     IODF80
                           --- SWITCH ----                                    UNIT ADDR                UNIT
                           ID  PR  CU  DYN  --- CONTROL UNIT ---  CU-            RANGE    -- DEVICE --  ADDR   DEVICE
CHPID  TYPE    SIDE  MODE     PN  PN  ID   NUMBER  TYPE-MODEL    ADD  PROTOCOL  FROM  TO   NUMBER,RANGE  START  TYPE-MODEL
-----  ------  ----  ----  ----------  ---  ------  ------------  ---  --------  ----  ---  ------------  -----  -------------
 0C    BL            DED                 00C1  3088                    S4        40   4F    0C40,16       40    3088
                                         00C2  RS6K                    S4        50   5F    0C50,16       50    RS6K
                                   ---------------------------------------------------------------------------------------
                                         0C60  3174                    S         60   6F    0C60,16       60    3278-2
                                   ---------------------------------------------------------------------------------------
                                         0C70  3174                    S         70   77    0C70,8        70    3174
                                   ---------------------------------------------------------------------------------------
                                         0C71  3791L                   S         78   7F    0C78,8        78    3791L
-------------------------------------------------------------------------------------------------------------------------------
  2F    CNC           REC                 0184  SCTC          1               10   11    0910,2        10    SCTC
                                   ---------------------------------------------------------------------------------------
                                         0185  SCTC          2               12   13    0912,2        12    SCTC
                                   ---------------------------------------------------------------------------------------
                                         0186  SCTC          3               14   15    0914,2        14    SCTC
                                   ---------------------------------------------------------------------------------------
                                         0187  3174          4   S           16   16    0916          16    3174
                                   ---------------------------------------------------------------------------------------
                                         0188  SCTC          5               18   1F    0918,8        18    SCTC

ESOTERIC NAMES

-
                                        AFFINITY  ALLOCATION   ASSOCIATED
  NAME        NAME TYPE  VIO  TOKEN  PREF  INDEX   DEVICE TYPE  GENERICS                DEVICE NUMBER LIST
  ---------  ---------  ---  -----  ----  --------  -----------  ----------  ------------------------------------------------
  ------------------------------------------------------------------------------------------------------------------------
  RS6K17      ESOTERIC              8300   FFFF     00004101     SCTC        0912 0913 0914 0915
  ------------------------------------------------------------------------------------------------------------------------
  RS6K43      ESOTERIC              8389   FFFF     00004103     RS6K        0C50 0C51 0C52 0C53 0C54 0C55 0C56 0C57 0C58 0C59 0C5A
                                                                            0C5B 0C5C 0C5D 0C5E 0C5F
```

Figure 31. HCD gen listing

Note in both cases the first two subchannel addresses were taken for the CLAW, C40 and C41 in the BLKMUX definition and 910 and 911 in the ESCON definition. Thereafter, a number of CLIO subchannels are defined for each. For the ESCON, those used were in addresses 12 to 15. The other definitions were not varied on at the MVS host end. For BLKMUX, they were 53, 54 and 55. Also included are the `ESOTERIC` names for the device number list. `ESOTERIC` names are used in the MVS system configuration to give a name to a list of devices, thus making

identification simpler. CLIO utilizes the ESOTERIC name of the list and is referenced in the *$HOME/parctc*.

6.3.1.1 Configuring ESCON and BLKMUX Channels for CLIO

Follow the steps outlined in Chapter 3, “RISC System/6000 S/390 ESCON Channel Connection” on page 15 and Chapter 4, “ RS/6000 S/370 Block Multiplexer Channel Connection” on page 33 for configuring the channels and subchannels. Figure 32 shows the subchannel definitions for the various subchannel addresses for CLIO with ESCON and for the two TCP/IP CLAW channels. The subchannel set name was chosen to show it is a CLIO subchannel, but the naming convention is entirely up to you. The ESCON configuration was direct attach, and the default path through the ESCD was taken.

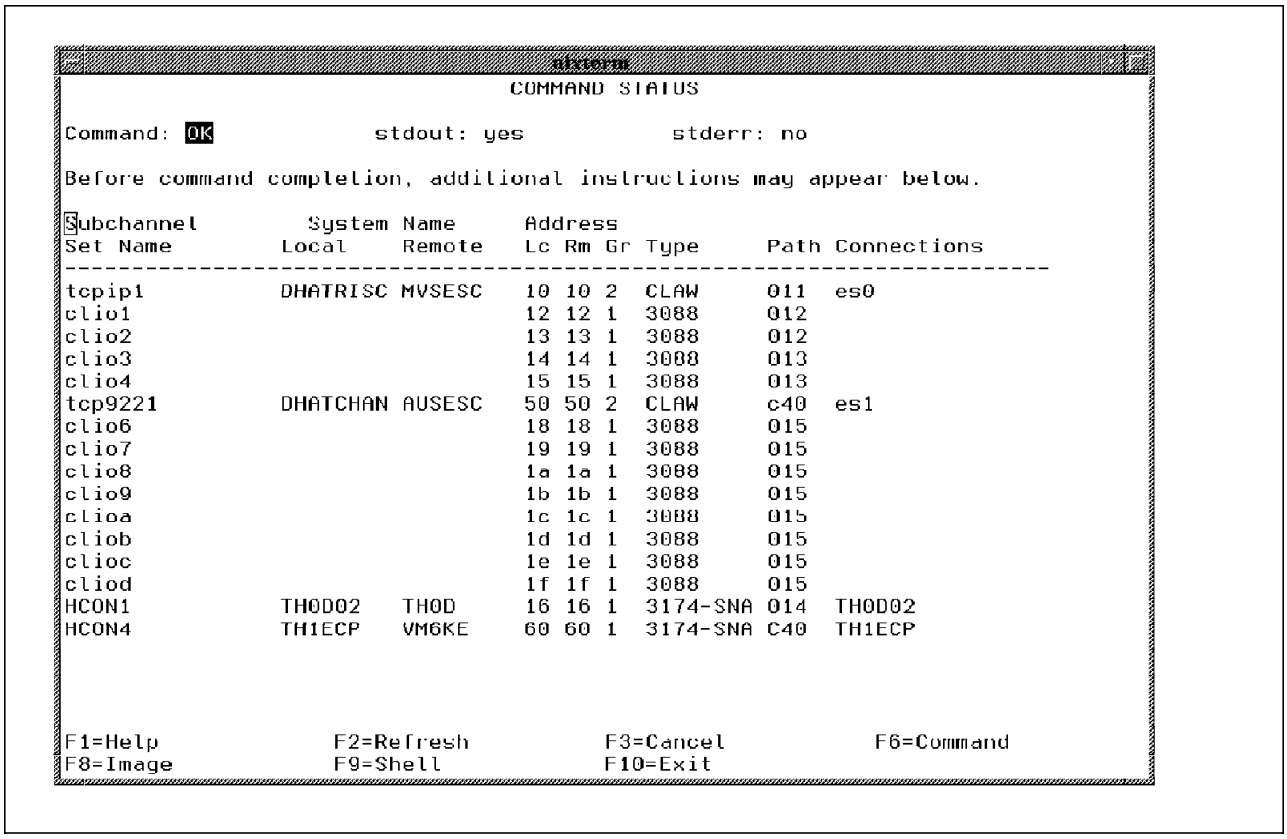


Figure 32. Subchannel Definitions for ESCON

Figure 33 on page 75 shows the subchannel definitions for the various subchannel addresses for CLIO with BLKMUX and for the two TCP/IP CLAW channels

The BLKMUX configuration was direct attach. Figure 33 on page 75 for the various subchannel addresses for CLIO.

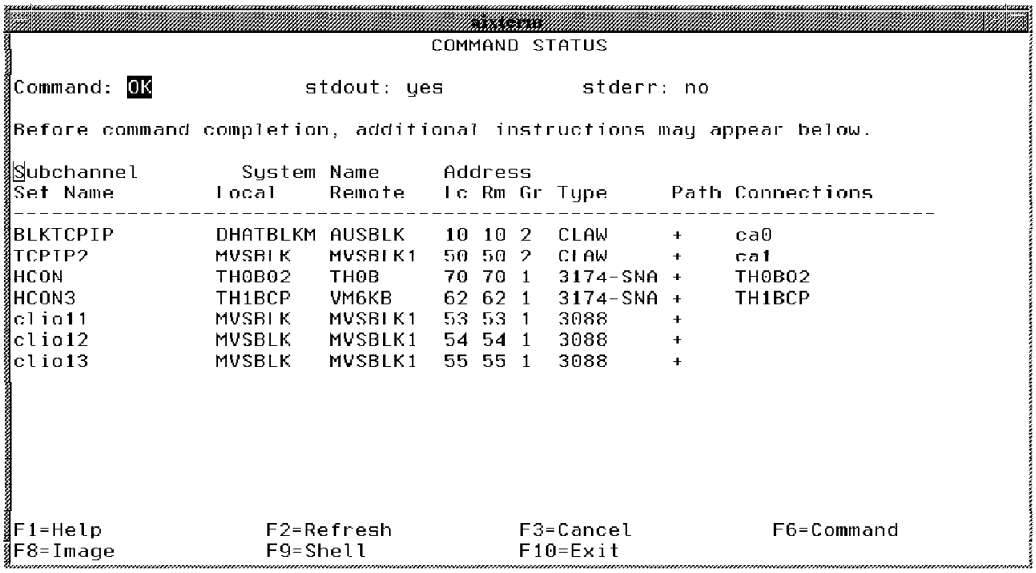


Figure 33. Subchannel Definitions for BLKMUX

6.3.2 Creating and Customizing the CLIO Configuration Files

There are three CLIO configuration files that need to be created and updated. They are used so that the applications that use CLIO do not have to know about the hardware, network definitions and the software protocols being used. They are shown in Table 3.

Table 3. Configuration Files Used by CLIO

File Name	Created by	Resides on	Purpose	Notes
PARCONF	Administrator, User	MVS and AIX	Used to establish the connection	Each user needs access
PARCTC	Administrator, User	Each Workstation using ESCON or BLKMUX	Used when creating a CTC connection	Not needed on MVS
NETRC (.netrc on AIX)	User	MVS and AIX	Holds hostnames and passwords	Standard AIX file

The main RS/6000 CLIO user was user “martin,” and the three main files are in the user martin home directory.

6.3.2.1 The PARCONF File

The parconf file was set up on our RS/6000 as /u/martin/parconf.

```
MVSBLK   2  3   MVSBLK1
DHATRISC 2  4   MVSESC
```

The parconf file on the MVS system was set up as DEMO2.PARCONF.

```
riscdum   1  4   DHATRISC
risc1     1  3   MVSBLK
```

Parconf contains information about how to locate nodes and how to connect to them. The one-line entries above have a syntax:

```
nickname type protocol nodeaddress extra
```

Which is explained by:

- nickname

  Arbitrary name that is unique within the parconf file.

- type

  This is the processor type. Number 1 is the RISC System /6000; number 2 is for MVS.

- protocol

  Communications method used. Number 3 is block multiplexer; number 4 is ESCON.

- nodeaddress

  Internet address or nodename as specified in /etc/hosts.

- extra

  We did not use this, but it can overide defaults. See the 6.1.1, “CLIO User’s Guide” on page 65.

We have two connections to define ESCON and BLKMUX. The nodeaddress can be referenced to the nodename for the adapter card via the /etc/hosts file in AIX.

```
18.0.0.2        MVSESC
18.0.0.1        DHATRISC dhatrisk
21.0.0.1        MVSBLK dhatrisk
21.0.0.2        MVSBLK1
```

Similarly, the SYS1.TCPPARMS.AUSHATA file (the default for TCP/IP is SYS1.TCPPARMS.PROFILE) references to the IP address, the network and the channels. Please see A.2, “SYS1.TCPPARMS.AUSHATA” on page 138 for more details.

6.3.2.2 The PARCTC File

The parctc file contains information about the subchannels, the ESOTERIC name and the hostnames. The parctc file in our system was in /u/martin/parctc.

```
MVSESC dhatrisk /dev/esca0 17 912-915
MVSBLK1 dhatrisk /dev/cat1 43 0C53-0C55
```

It contains one-line entries for each BLKMUX or ESCON connection. The syntax is:

```
hostnode workstationnode adaptername adapterid unitrange
```

Where:

- hostnode

  The Internet address or nodename associated via /etc/hosts.

- workstationnode

  The Internet address or nodename associated via /etc/hosts.

- adaptername

  The ESCON or BLKMUX adapter name with format **/dev/cat**n or **/dev/esca**n where **n** is the adapter number returned by the `lsdev -Cc adapter` command.

- adapterid

  This corresponds to the ESOTERIC NAME of the device address. The ESOTERIC for the ESCON subchannels was RS6K17 and RS6K43 for the BLKMUX channels. The last two digits are set to this value. Please see Figure 31 on page 73 for further details.

- unitrange

  The range associated with the CLIO subchannels, NOT the CLAW.

6.3.2.3 The .netrc

The .netrc is used for CLIO for user ID and password information. For our set up in /u/martin/.netrc and DEMO2.NETRC, and for simplicity, we have identical files:

```
machine MVSESC login demo2 password cou1rt
machine DHATRISC login martin password cou1rt
machine AUSBLK login VMFTP1 password HK123
machine MVSBLK1 login demo2 password cou1rt
machine MVSBLK login martin password cou1rt
```

6.3.3 Method of starting parsrv

The parsrv is started on the MVS host via the SYS1.PROCLIB member, PARSV2 (in our case) or FCFPARSR as a default.

It is started on the RS/6000 by issuing the command `parsrv`.

It was simpler in our setup to put this inside the .profile of the user martin. A better option in a customer site would be to put it inside /etc/services.

6.3.4 CLIO Environmental Variables

There are a number of environment variables associated with CLIO, and as a whole, we left them to default. Setting the CLIO debugging variable

```
CLIODEB=-1
```

was useful when we were testing the system using the FCFIDUM module that we report on in the results section.

The variable CLIOTMO was set to 600 and exported:

```
dhatrisk:/u/martin> export CLIOTMO=600
```

CLIOTMO sets the length of time the master waits for the servant. We allowed a much greater time for connection since this was needed. It took between 15 and 20 seconds for the parsrv deamon to come back with the clftp prompt and be ready for put and get.

6.3.5 Using CLIO Striping

CLIO/S striping function allows a single CLIO/S data transfer to be distributed across two or more channel adapters. We used striping from a Model C20 RS/6000 with two ESCON channels running AIX 4.1 when this redbook was written. We tested using memory-to-memory transfer and obtained significantly improved performance (8.2 MB/S to 12.4 MB/S for one or two adapters). We also striped on our own system across multiple subchannels on a single adapter. The performance benefit of this was much less.

Striping across adapters follows these rules:

- All adapters must be the same type: all ESCON or all BLKMUX
- All adapters must be attached to the same host

Configuring striping is relatively trivial and involves some minor modifications in the parctc file.

The parctc file configured for striping across one channel looks like:

```
MVSESC dhatrisk /dev/esca0 17 912
MVSESC dhatrisk /dev/esca0 17 913
MVSESC dhatrisk /dev/esca0 17 914
MVSESC dhatrisk /dev/esca0 17 915
MVSBLK dhatrisk /dev/cat1 43 0C53-0C55
```

If we had multiple adapters, for instance es0 and es1, add multiple entries for each device.

```
MVSESC dhatrisk /dev/esca0 17 912-915
MVSESC dhatrisk /dev/esca1 18 660-66F
```

To activate striping, one has to update the *CLIOSTRIPES* variable by using the command dhatrisk:/#> CLIOSTRIPES=2 or by the number of stripes that you need.

6.3.6 Using CLIO and SNA on the Same Adapter

Our environment was set up to run SNA and CLIO over the same channel adapter. A quick look at Figure 31 on page 73 shows that there are reserved subchannels for 3174 control units and we have configured SNA to run the RS/6000 as a link station (LU2) over the same CHPIDs. Our test was to log into the mainframe as a TSO user and do active work while we were doing a sustained download of a large file using CLIO. Please see Chapter 5, “SNA Channel Connectivity Using AIX SNA Server/6000” on page 49 as a reference. Similarly, we ran TCP/IP tests while CLIO was utilizing the channel. CLIO can run alongside CLAW applications on the same channel.

The reason for specifying a range of subchannels is that if the first subchannel is busy with another application, then CLIO will use the next subchannel available. Applications cannot use the same subchannel as CLIO.

6.4 Configuring and Managing the CLIO/S Tape Pseudo Device Driver

Part of this work was to back up a DB2/6000 database, or part of one, to MVS tape and then to reinstall it. The test MVS system has a bank of 3380 DASD and 3480 tape drives. One of the 3380s was available to our system. The database is international; it is data in Spanish. We section parts of it, pass it through the back-up architecture to MVS tape and reinstall it.

There is a standard AIX interface to the CLIO tape server, TAPESRV, giving users access to the pseudo tape device by using standard commands such as `tar`, `cpio` or `backup`. For our backups, we used `tar`.

There is a tape mount deamon running on the AIX system that looks after all requests sent to the pseudo device driver.

6.4.1 Configuring and Starting the CLIO Tape Mount Deamon

The first device created is */dev/pst0*, the pseudo device driver. For more tape devices, more pseudo device drivers have to be created.

After successfully installing the product, we used the following steps to configure our pseudo device driver. From the command line, you can issue the command:

```
mkdev -c tape -s remote -t cliotape
```

This will take you to step 4 immediately. As user root:

1. Start parsrv.
2. Invoke the correct SMIT screen with the command

   ```
   smitty tape
   ```

3. On the tape drive panel, select **Add a tape drive**.
4. In the selection list of devices, select **cliotape**.

   Note: If cliotape does not exist, the device driver was not installed properly.

   Figure 34 on page 80 is the configuration of the tape server.

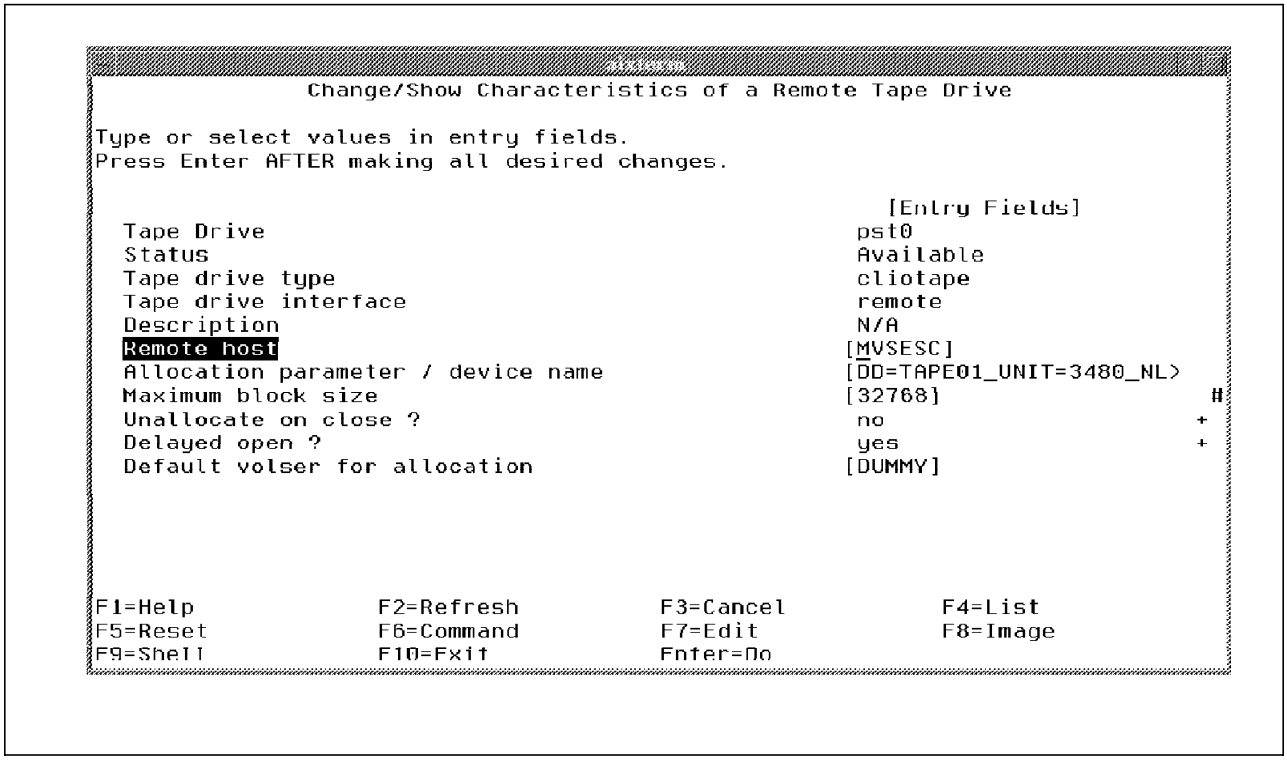


Figure 34. Adding a Remote Tape for CLIO

5. We updated the following fields:

   - **Remote Host**

     MVSESC is the hostname of the MVS host as defined in the parconf file (on the RS/6000 and the MVS user ID of DEMO2.PARCONF in this case), in the /etc/hosts file on the RS/6000 and in the SYS1.TCPPARMS.AUSHATA (or SYS1.TCPPARMS.PROFILE for default) on the MVS host.

   - **Allocation parameter/device name**

     For RS/6000 tape drives, use /dev/rmt0 and the remote host name of another RS/6000 as above.

     For MVS tape drives, we used the typical MVS definitions.

     DD=TAPE01_UNIT=3480_NL>_DSN=DEMO2.TEST

     Where the DEMO2.TEST is our chosen default dataset name. Add this complete definition in the Allocation parameter field.

   - We updated the unallocate field to specify *no*.

     Yes unallocates the device after a command, such as tar, issues a close. We wanted to make the tape stay mounted; so we used *no*.

   - We updated the delayed open to *yes*.

     Yes means the device waits for a read or write or IOCTL before the device is opened. *No* opens the device when allocated.

     When doing backups, the *yes* insert should be used.

- We used the default VOLSER of *DUMMY*. The LED on the 3480 asks for this volume to be mounted in the instruction for the tape operators. (This is a useful check that you have everything correct!!)

6. Press Enter, and select **do**.

6.4.2 Using the Tape Mount Daemon

We used a couple of script files to allocate the device write to a tape and then read back. The `cltalloc` command allocates the tape device and has some reasonably complex syntax:

```
Usage: cltalloc [-v [vol=]volser] [-d [dsn=]dsname] [-p new | old] [-c]
[-l nl | al | blp | sl] [-e [ex=]yyddd] | -r [ret=]nnnn] -f device
```

The `cltunall` command unallocates the device. The two scripts were relatively simple:

- **write script**

  ```
  cltalloc -p old -v DUMMY -d DEMO2.TEST -f /dev/pst0
  tar -cvf /dev/pst0 /tmp/DB2
  cltunall -f /dev/pst0
  ```

  Where the VOLSER ID is `DUMMY` and `/tmp/DB2` is the DB2 database files.

- **read script**

  ```
  cltalloc -p old -v DUMMY -d DEMO2.TEST -f /dev/pst
  tar -cvf /dev/pst0
  cltunall -f /dev/pst0
  ```

The files were then re-entered into the DB2 structure, and they tested for authenticity successfully.

6.4.3 CLFTP Results and Throughput Considerations

This section concerns the:

- Throughput considerations
- Testing of what is happening beneath the surface of the product using some CLIO functions
- The actual throughput achieved using CLFTP in the Austin configuration See Chapter 8, “Throughput Figures and Tuning” on page 131 for more information.
- Throughput achievable for other configurations, including maximum performance.

6.4.3.1 Throughput Considerations

CLIO throughput figures are usually quoted as memory-to-memory and there is a very good reason for doing so. This is to simplify the performance monitoring of only the CLIO product itself and not any other rate-determining factor in the data pipe from the mainframe host to the RS/6000.

Note: The term *data pipe* will be used frequently and is the path of data from RS/6000 disk or tape to MVS disk or tape.

However, memory-to-memory is meaningless in a real environment, and without determining how the other factors effect the throughput, an estimate cannot realistically be made.

When configuring a data-transfer process, all of the following factors must be considered.

- MVS tape subsystem
- RS/6000 tape subsystem
- MVS DASD subsystem
- RS/6000 disk
- RS/6000 CPU size
- Mainframe CPU cycles associated with TC/PIP and CLIO

  The LPAR size, TCP/IP dispatch priority, batch window
- Method of connectivity for all of the above (ESCON, BLKMUX, TR,) and so on)

Getting maximum performance from CLIO/S requires some effort to analyze and configure the systems involved and to tune the processes that use the CLIO/S interfaces. As a rule, mainframe sites will implement CLIO with as little disruption of their existing configurations and practices as possible. This makes getting into production much easier, but limits the results that CLIO/S can achieve. Spending more time up front will prove to be beneficial in the long run.

6.4.3.2 The Performance Task You May Approach at Your Site

Creating an effective data-transfer process requires some effort. Here is one view from CLIO development experts of the steps involved. IT could be used a project definition workshop structure for your customers. This will be an iterative process since the outcome of later steps may cause you to revisit earlier ones.

1. Determine your requirements
   - Source and target
   - Transfer size
   - Transfer window
2. Design a transfer process
   - Data extraction process
   - Data transfer method
   - Intermediate storage media
   - Final storage media
3. Match the process to your system
   - Analyze components
     - Capacity
     - Workload
     - Configuration
   - Step-by-step testing
     - Identify potential bottlenecks
4. Implement the process
   - Install any new components

- Write code, JCL, shell scripts, etc.

5. System testing
   - Throughput
   - Resource utilization
   - Ease of use
   - Total effectiveness

6.4.3.3 Testing of the CLIO function

CLIO has ample functions for testing the throughput of memory memory-to-memory and the detail of what is happening throughout the processes.

The test site data pipe is shown in Figure 36 on page 89

The FCFIDUM program verifies that CLIO is set up correctly. Issue the command for the RS/6000 or the MVS host. We mainly submitted it from the RS/6000.

The syntax is relatively simple, and an example of our commands is:

```
>  fcfidum MVSESC 2000 2
```

Where:

- MVSESC is the node name on which to start the servant, which, in this case, was the MVS host.
- The 2000 is the size of the array to be submitted; the default is 1000.
- The 2 is the *count* for the number of times you wish to submit the array/

If the CLIO debug flag is set to *on* via setting and exporting:

```
> export CLIODEB=-1
```

The FCFIDUM output looks like this:

```
dhatrisk:clio_ini_0: CLIO/S Version 2, Release 1
dhatrisk:readconf_0: Using parconf=/u/martin/parconf
dhatrisk:readconf_0: Machine 1 on mvst=mvsesc. Type=2 proto=4
dhatrisk:readconf_0: Machine 2 on risdum=dhatrisc. Type=1 proto=4
dhatrisk:soc_ok_0: SOC is not available
dhatrisk:ctc_ok_0: ESCON may be available
dhatrisk:ctc_ok_0: PCA may be available
dhatrisk:add_mach_0: Machine 3 on dhatrisk. Type=0 proto=0
dhatrisk:OPNFAM_0: Created new family 0
dhatrisk:OPNPAR_0: CLIODEB = 0xffffffff
dhatrisk:OPNPAR_0: CLIOBIN = $HOME/bin
dhatrisk:OPNPAR_0: CLIOSYS = /usr/local/CLIO:/usr/bin
dhatrisk:OPNPAR_0: CLIOBUF = 32768
dhatrisk:OPNPAR_0: CLIOBLK = 0
dhatrisk:OPNPAR_0: CLIOTBS = 0
fcfidum: OPNPAR succeded. myid=0 pid=44928
dhatrisk:add_mach_0: Machine 4 on 18.0.0.2. Type=0 proto=0
dhatrisk:OPNSRV_0: Starting slave on 18.0.0.2. Port=1037
dhatrisk:OPNSRV_0: Contacting slave on 18.0.0.2
dhatrisk:do_contact_0: Contacting slave 1 in family 0
dhatrisk:tcp_accept__0: OPNSRV  Starting port=1037
dhatrisk:tcp_accept__0: OPNSRV  OK
dhatrisk:clio_connect_0: Starting hid=0x12000002  Proto=0
dhatrisk:CTC_OPN_0: ctc_opn - Protocol=ESCON. CLIOstripes requested: 1
dhatrisk:CTC_OPN_0: ctc_opn - Hostnode is mvsesc.
dhatrisk:CTC_OPN_0: readconf- parctc file used : /u/martin/parctc
dhatrisk:CTC_OPN_0: readconf- MVSESC  device=/dev/esca0  id=17  unit=(912,915)
dhatrisk:CTC_OPN_0: ctc_opn - Stripe 1: Esoteric name is RS6K17. Unit=(912,915)
dhatrisk:CTC_OPN_0: ctc_opn - Stripe 1: Trying unit 0x912
dhatrisk:CTC_OPN_0: ctc_opn - Stripe 1: Opening /dev/esca0
dhatrisk:OESC   -0: oesc    - schl=0x12:  status=C0 cutype=3 opmode=80
dhatrisk:OESC   -0: oesc    - poll 0002 - ready for writing
dhatrisk:OESC   -0: oesc    - Subchannel 0x12: Open_Cnt=1 fd=5 Exit OK
dhatrisk:CTC_OPN_0: ctc_opn - Stripe 1: Subchannel is 0x12
dhatrisk:CTC_OPN_0: ctc_opn - Stripe 1 of 1: RS6K17. Subchannel=0x12 Ackint=214..
dhatrisk:clio_connect_0: OK       hid=0x12000002  Feat=0x00000000  ESCON
dhatrisk:do_contact_0: p->pid = 207685
dhatrisk:do_contact_0: Contacted process 1 on 18.0.0.2. Protocol=ESCON
dhatrisk:OPNSRV_0: Started slave 1 on 18.0.0.2. Pid=207685  ESCON
fcfidum_0: OPNSRV succeded. isrv=1
dhatrisk:do_writ_0: INTE dst=1 nitem=1 bpi=4 stride=4
dhatrisk:clio_writ_0: --> 0x12000002(207685)  INTE ESCON   nbyte=4
dhatrisk:CTC_OPN_0: ctc_wr  - Protocol=ESCON  nbyte=4 async=0
dhatrisk:OESC   -0: wesc  WR- nbyte=4
dhatrisk:OESC   -0: wesc  WR- Stripe 1: Bytes xferred : 4 Total=4 status=  0x0
dhatrisk:OESC   -0: wesc  WR- Total bytes xferred: 4
dhatrisk:do_writ_0: INTE dst=1 nitem=1 bpi=4 stride=4
dhatrisk:clio_writ_0: --> 0x12000002(207685)  INTE ESCON   nbyte=4
dhatrisk:CTC_OPN_0: ctc_wr  - Protocol=ESCON  nbyte=4 async=0
dhatrisk:OESC   -0: wesc  WR- nbyte=4
dhatrisk:OESC   -0: wesc  WR- Stripe 1: Bytes xferred : 4 Total=4 status=  0x0
dhatrisk:OESC   -0: wesc  WR- Total bytes xferred: 4
fcfidum_0: nbyte=2400800  niter=1
dhatrisk:do_writ_0: INTE dst=1 nitem=600200 bpi=4 stride=4
dhatrisk:clio_writ_0: --> 0x12000002(207685)  INTE ESCON   nbyte=2400800
dhatrisk:CTC_OPN_0: ctc_wr  - Protocol=ESCON  nbyte=2400800 async=0
dhatrisk:OESC   -0: wesc  WR- nbyte=2400800
dhatrisk:OESC   -0: wesc  WR- Stripe 1: Bytes xferred : 61440 Total=61440 sta..

PLUS LOTS OF LINES SHOWING THE DATA TRANSFER....and then
```

```
dhatrisk:clio_read_0: <-- 0x12000002(207685)  DOUB ESCON   nbyte=8
dhatrisk:CTC_OPN_0: ctc_rd  - Protocol=ESCON  nbyte=8 async=0
dhatrisk:OESC   -0: resc  RD- nbyte=8
dhatrisk:OESC   -0: resc  RD- Stripe 1: Bytes xferred : 8 Total=8 status=  0x0
dhatrisk:OESC   -0: resc  RD- Total bytes xferred: 8
Speed_0: 5645.840634 kb/s 415.266991 ms
Speed_1: 5620.618127 kb/s 417.130500 ms
Mean   : 5633.201147
CHECK enabled
dhatrisk:CLSPAR_0: Calling clsexit(0)
dhatrisk:CLSFAM_0: Closing family 0
dhatrisk:CLSSRV_0: Closing process 1 in family 0
dhatrisk:CTC_OPN_0: ctc_cls - Stripe 1: Allowing Host to Close 0x912
dhatrisk:CTC_OPN_0: ctc_cls - Stripe 1: Waiting for Host to Close 0x912
dhatrisk:CTC_OPN_0: ctc_cls - Stripe 1: Received code 17 across TCP/IP
dhatrisk:CTC_OPN_0: ctc_cls - Stripe 1: Received code 18 across TCP/IP
dhatrisk:OESC   -0: cesc    - Stripe 1: schl=0x12 fd=5 Open_Cnt=1
dhatrisk:OESC   -0: cesc    - Stripe 1: Close subchannel 0x12 succeeded.   RC=0
dhatrisk:CTC_OPN_0: ctc_cls - Stripe 1: Close Subchannel Finished Normally
dhatrisk:CTC_OPN_0: ctc_cls - Exit
dhatrisk:CLSSRV_0: Received HARAKIRI from slave 1
dhatrisk:clio_end_0: Entered. Pid=44928
dcfidum finished successfully!!!
```

It is instructive to follow the process all the way through to the point where three files .netrc, parctc and the parconf file are used.

The .netrc file gets us to line 1.

This is the handshake via TCP/IP. From the top on line 2, the parconf file is used and identifies that we have asked for the ESCON protocol by Type proto=4. At line 17, it is starting the slave on the MVS host on port 1037. The system uses /u/martin/parctc to look at which esoteric device we have determined and which individual subchannel device range is to be used. As the CLIOstripes variable is set to 1, it only proceeds to use channel 912 if it is available.

It finds it available and then proceeds to send the data in the allotted schedule.

After all the data is sent, it tallies up an average, which in this case was 5.6 MB/S, and proceeds to close the subchannel and give you a pleasant message.

6.4.3.4 Using FCFRWTST

Similar to FCFIDUM, the FCFRWTST program can be used to determine the bandwidth we are obtaining across the channel.

Here is an example of commands we issued:

```
>  fcfrwtst 18.0.0.2 6000000 9 1 0
```

The output is shown in Figure 35 on page 86, where:

- 18.0.0.2 was the IP address of the MVS host for ESCON
- 6000000 is the nbytes
- nbuf=2
- nloop=9
- M->S Which means we are downloading from the MVS the RS/6000

```
For the MVS machine, our LPAR was only 40 percent of the 9121. Two
LPARs were present, both were capped; so no interference between the two
occurred.                                                              .

TEST1
fcfrwtst MVSESC 6000000 2 9 1 > cap.test1

RESULTS 1

rwtst: nbyte=6000000 nbuf=2 nloop=9 M->S
0 - speed: 4963.552947 KB/s  2360.960007 ms
1 - speed: 5777.735617 KB/s  2028.259993 ms
2 - speed: 5777.032090 KB/s  2028.506994 ms
3 - speed: 5781.740700 KB/s  2026.854992 ms
4 - speed: 5776.829755 KB/s  2028.578043 ms
5 - speed: 5777.852436 KB/s  2028.218985 ms
6 - speed: 5775.440618 KB/s  2029.065967 ms
7 - speed: 5781.720637 KB/s  2026.862025 ms
8 - speed: 5781.084813 KB/s  2027.084947 ms


TEST2
fcfrwtst MVSESC 6000000 2 9 0 > cap.test2

RESULTS 2

rwtst: nbyte=6000000 nbuf=2 nloop=2 S->M
0 - speed: 6761.625929 KB/s  1733.126044 ms
1 - speed: 8497.444906 KB/s  1379.091024 ms
2 - speed: 8497.402304 KB/s  1379.097939 ms
3 - speed: 8497.266420 KB/s  1379.119992 ms
4 - speed: 8497.001277 KB/s  1379.163027 ms
5 - speed: 8492.174300 KB/s  1379.946947 ms
6 - speed: 8496.292595 KB/s  1379.278064 ms
7 - speed: 8458.515852 KB/s  1385.438085 ms
8 - speed: 8493.251380 KB/s  1379.771948 ms
```

Figure 35. Output of FCFRWTST Command

6.4.3.5 Transferring Files Using CLIO

We transferred files using `clftp` followed by `put` or `get`. A brief discussion is warranted about how CLIO FTP works, the command syntax and how to automate a more complex data transfer in a customer environment inside a batch job. Chapter 6 in the CLIO User's Guide goes into `clftp` in great detail, but we specifically want to explain how to quickly get up and running with file transfer and how one would go about transferring data from a working database that has changed during the period of one day. The scenario of sending data to an RS/6000 overnight from MVS, working with it during the day and then saving the changes back to MVS tape/DASD is a quite common request.

The syntax of CLFTP is similar to FTP. It uses the .netrc file, or the NETRC file on the MVS machine, to interpret user ID and passwords.

The syntax is:

```
clftp remote_host [:remote_host]
```

Where the only complexity is:

`remote_host:remote_host ...`

If you specify a colon-separated list of remote_hosts, you are requesting file transfer through intermediate host.

After issuing the `clftp` command, you are returned to the `clftp` prompt. For example, for our BLKMUX connectivity:

```
dhatrisk:/u/martin> clftp MVSBLK1
Connecting to : MVSBLK1
Remote system MVSESC job DEMO2D connected with protocol BMCA.
clftp >
clftp > get 'DEMO2.FTP.LARGE' /tmp/db2/datafile

226 Transfer complete(53681163 bytes).
Elapsed time: 40.95 sec.     1.25 Meg/s
```

You can check the status of the connection and your default parameters via:

```
clftp > status

Connected to: 21.0.0.2 (MVS)
ascii: On               Status Freq.: 4
Last Site String:
Transport Protocol: BMCA
Translation Site: Local
Maxbuflen = 500000      Firstbuflen = 100000    Secondbuflen = 100000

 Remote file allocation values.

File Organization  : Physically Sequential.
Record Format      : Variable Blocked.
Record Length      : 260
Block Size         : 27648
Storage Unit Type  : Cylinders.
Primary Units      : 1
Secondary Units    : 1
Directory Blocks   : 27
Volume             :
Dcbdsn             :
Unit               :
Alloc              :

200 Command completed successfully.
```

The following commands are the commands that are similar to FTP that can be used:

```
clftp > help
The Commands available are:

APpend     ENDONERRor LMKdir     PUt
AScii      Get        LOCSIte    PWd
Binary     HAsh       LOCSTat    QUIt
CD         HElp       LPwd       SIte
DELete     LCd        LS         TRANSlate
ECHO       LLS        MKdir      ?

clftp >
```

Most are self-explanatory, but LPwd tells you the present directory, for example.

6.4.3.6 Starting CLFTP as a Batch Job

We submitted clftp as a batch job via the JCL below. Clftp reads input using the SYSIN DD statement and writes output using the SYSPRINT DD statement.

```
       //FCFTP1    JOB 1,PP.NUMBER.5648.129,MSGLEVEL=1,REGION=4096K
       //FCFTP     EXEC PGM=CLFTP,PARM='DHATRISC'
       //SYSUDUMP  DD SYSOUT=4
       //STEPLIB   DD DSN=SYS1.SFCFLOAD,DISP=SHR
       //SYSPRINT  DD SYSOUT=4
       //SYSIN     DD *
       userid
       password
       echo
       binary
       locsite pri=10 sec=10
       get /tmp/DB2/datafile 'demo2.db2file.large'
       quit
       //
```

For equivalent jobs that are run every night, this is obviously the preferred method.

6.4.3.7 Results from Using CLFTP for the Austin Setup

Figure 36 on page 89 shows the hardware and software setup for the Austin system.

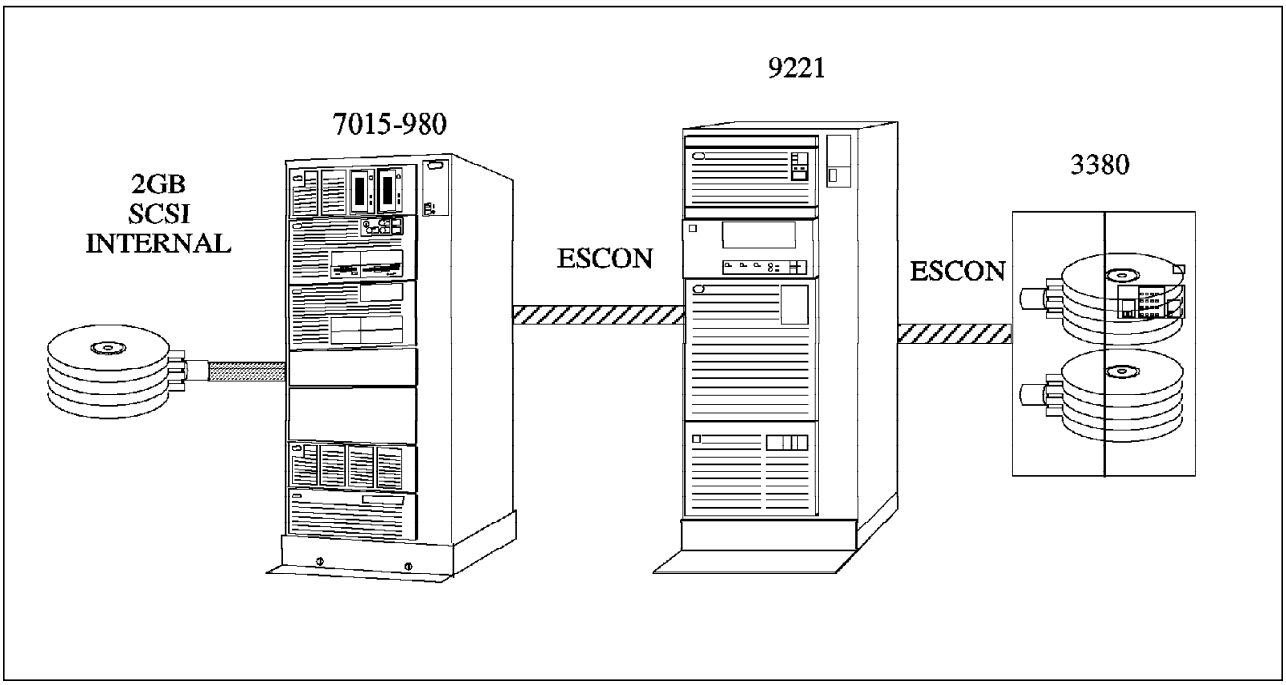


Figure 36. Diagram for the Austin System

The CLIO performance obtained for the Austin system is shown in Table 4.

Table 4. CLIO/S Performance for the Austin Test Site

Channel Type	Workstation Model	Direction	Megabytes Per Sec	CLIO Interface	MVS CPU Utilization	RISC CPU Utilization
Protocol: CLIO/S CTC connection over ESCON						
ESCON	990	from MVS	2.6	DPI	14%	24%
ESCON	990	to MVS	2.1	DPI	13%	26%
Protocol: CLIO/S CTC connection over BLKMUX						
BLKMUX	990	from MVS	1.25	DPI	15%	26%
BLKMUX	990	to MVS	1.1	DPI	14%	28%
MVS: LPAR was maximum 40 percent of the CPU. TCP/IP and user jobs had normal dispatch priority.						

The above is for the complete disk-to-disk transfer. In order to check how fast the disk and internal BUS would drive the process, we wrote from one RS/6000 internal disk to another RS/6000 internal disk using the `tar` command. We obtained the following figures from the `iostat` command:

```
Disks:        % tm_act     Kbps      tps    Kb_read   Kb_wrtn
hdisk0          100.0   2740.0    172.0       2740         0
hdisk1          100.0   2688.0     68.0          0      2688
hdisk4            0.0      0.0      0.0          0         0
hdisk3            0.0      0.0      0.0          0         0
cd0               0.0      0.0      0.0          0         0
```

One can see that the ESCON channel is driving the disk as fast as it can go and that the performance achievable is much higher, but we are disk limited. See

6.4.3.8, “Maximum Throughput” on page 90 for maximum CLIO ESCON throughput.

The BLKMUX channel appears to be running as fast as it can since we have not reached the limitations of the disk speed.

Mainframe people like to talk in GB/hour. For the system above under ESCON, we could achieve about 7.2 GB/hour. As seen below, this a factor of ten or so slower than a disk tuned system, but shows what can be achieved with slower internal 2 GB SCSI 2 disks. The disk was HDISK0 on the SCSI I/O controller:

```
scsi1            00-08            SCSI I/O Controller

      Device Driver Level.........00
      Diagnostic Level............03
      Displayable Message.........SCSI-2
      EC Level....................D25855
      FRU Number.................. 52G5484
      Manufacturer................IBM97N
      Part Number................. 52G5483
      Serial Number...............00013855
      Loadable Microcode Level....00A0
      ROS Level and ID............0005
      Read/Write Register Ptr.....0100

hdisk0           00-08-00-00      2.0 GB SCSI Disk Drive

      Manufacturer................IBM
      Machine Type and Model......0664M1H
      Part Number.................86F0101
      ROS Level and ID............35203530
      Serial Number...............00006214
      EC Level....................895118
      FRU Number..................86F0118
      Device Specific.(Z0)........000002029F00001E
      Device Specific.(Z1)........86F0620
      Device Specific.(Z2)........0980
      Device Specific.(Z3)........92323
      Device Specific.(Z4)........0002
      Device Specific.(Z5)........22
      Device Specific.(Z6)........895108
```

6.4.3.8 Maximum Throughput

It is very important to show what CLIO is capable of in an ideal environment as opposed to that achieved in our disk-limited situation. We include below the results from IBM benchmarks for CLIO; this should be born in mind when the overall data pipe is thought of.

This section discusses the maximum performance that was achieved by CLIO/S. In actual practice, lower levels of performance will be achieved due to factors other than CLIO or ESCON.

Memory-to-memory

- Performance testing consisted of sending blocks of data many times and capturing average performance. Block sizes range form 60 K to 1200K. Iterations vary between 90 and 1000. On systems with no other workload, there was very little variation between runs.

- CPU utilization was gathered by various methods, including the AIX `iostat` command, TSO RMFMON and the C language `clock()` function. Due to differences in these tools and the way they are used, the utilization percentages should only be considered to be rough estimates.
- Data in Table 5 is for memory-to-memory transfer. Using the CLIO/S FTP or tape server interfaces will involve additional device overhead and will result in lower transfer rates.
- MVS system is a 600-J.
- AIX system is SP2 wide node.
- ESCON channels are 10-megabyte.
- One megabyte = 1,048,576 bytes.

Table 5. CLIO/S Performance

Channel Type	Workstation Model	Direction	Megabytes Per Sec	CLIO Interface	MVS CPU Utilization	RISC CPU Utilization
Protocol: CLIO/S CTC connection over ESCON						
ESCON	SP2	from MVS	9.1	DPI	3%	9%
ESCON	SP2	to MVS	8.3	DPI	3%	9%
Protocol: CLIO/S TCP/IP connection over ESCON						
ESCON	SP2	from MVS	3.0	DPI	15%	1%
ESCON	SP2	to MVS	1.8	DPI	27%	15%

Discussion: As advertised, CLIO/S can deliver “near-channel” speeds, surpassing 80-90 percent of a 10-megabyte ESCON channel. By adjusting the transmitted block sizes, it is possible to surpass 98 percent of channel speed from MVS to AIX.

Note that even with the increased data transmission rates, MVS CPU utilization is much less for CTC connections than for TCP/IP. Given that TCP/IP takes 3-4 times as long as CTC to move data, and uses 5-9 times as much CPU while doing it, it is fair to say that TCP/IP uses 15-36 times as many MVS cycles as CLIO/S CTC connections.

6.4.3.9 Instruction Counts

Using the Generalized Trace Facility (GTF) on MVS, it has been observed that CLIO/S uses less than 2000 instructions to accomplish a 4 K memory-to-memory transfer to AIX. This is in rough agreement with the utilization and transfer rate data measured in the above testing.

6.4.3.10 Media-to-Media

The following configuration was used for a customer benchmark. Here, the requirement was to move 52 gigabytes of data per hour from AIX DASD to MVS DASD. In fact, 55 gigabytes per hour was achieved.

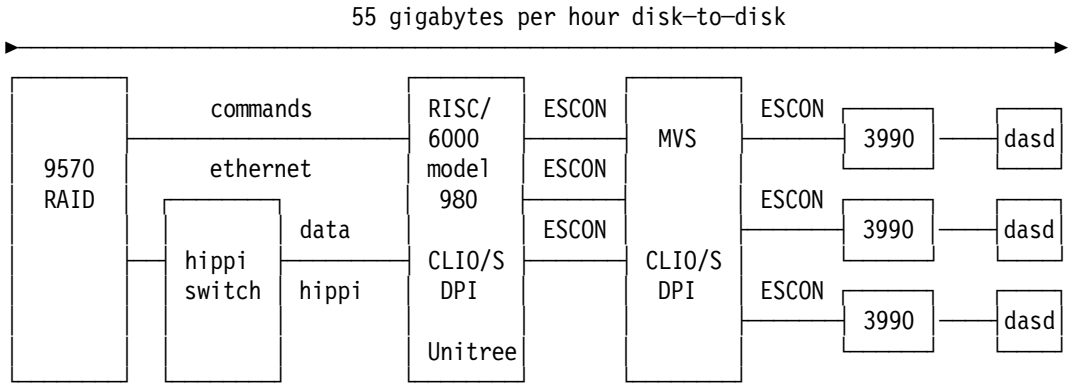


Figure 37. High-speed Disk-to-Disk Transfer

Discussion: Although all of the components are generally available, this is not a typical production environment:

- A High Performance Parallel Interface (HIPPI) switch and Unitree were used to pull data from the 9570 RAID array.
- Three ESCON channels were used to connect the workstation and mainframe, and three more were used to connect to the 3990 control units.
- Dasd striping was used on MVS for speed.

There is always a bottleneck. In this case, it was the RISC/6000 model 980 CPU, which ran at 98 percent utilization.

Chapter 7. VM Host Connection

This chapter explains about the RS/6000 direct channel attachment to VM hosts. With the advent of VM/ESA V1R2, VM offers the most flexible way to connect to the other machines. The dynamic device sensing allows users to avoid coding in DMKRIO or HCPRIO to define the rdevice, generate the CP nucleus, and shut down and reIPL of the system when the remote device is changed. This provides users with maximum system availability. We are also deal with TCP/IP and SNA connection between RS/6000 and VM hosts by both ESCON and Block Multiplexer channels, as discussed in the previous MVS section.

7.1 VM TCP/IP Connection

The RS/6000 workstation channel attachment should be thought of as an “attached processor” rather than a control unit. An operating system, a configuration definition, and an application are required. The most common application used with the RISC System/6000 channel attachment is TCP/IP. This section will describe the TCP/IP configuration for both VM host RS/6000 workstations and some FTP tests and measurements between them.

Following is the requirement of both software and hardware for direct channel attachment by VM TCP/IP;

- VM/SP Release 5 (5664-173) or VM/XA Release 2 (5664-308)
- VM/ESA V1.1.1 or higher (our environment is V.1.2.2)
- TCP/IP for VM V2.2 or higher (our environment is V2.3)
- S/370, S/390 or ES/9000 (in our case ES/9221-150 4 MIPS)
- 7 MB of user storage
- 40 MB direct access storage
- One tape drive
- One Block Multiplexer channel or 1 ESCON channel

For the requirements of both software and hardware for direct channel attachment implementation by AIX TCP/IP, please see Chapter 4, “ RS/6000 S/370 Block Multiplexer Channel Connection” on page 33.

The following figure is the Austin test environment. We have 14.0.0.X for ESCON connection and 13.0.0.X for BLKMUX connection; each have hostnames for the IP addresses for differentiation.

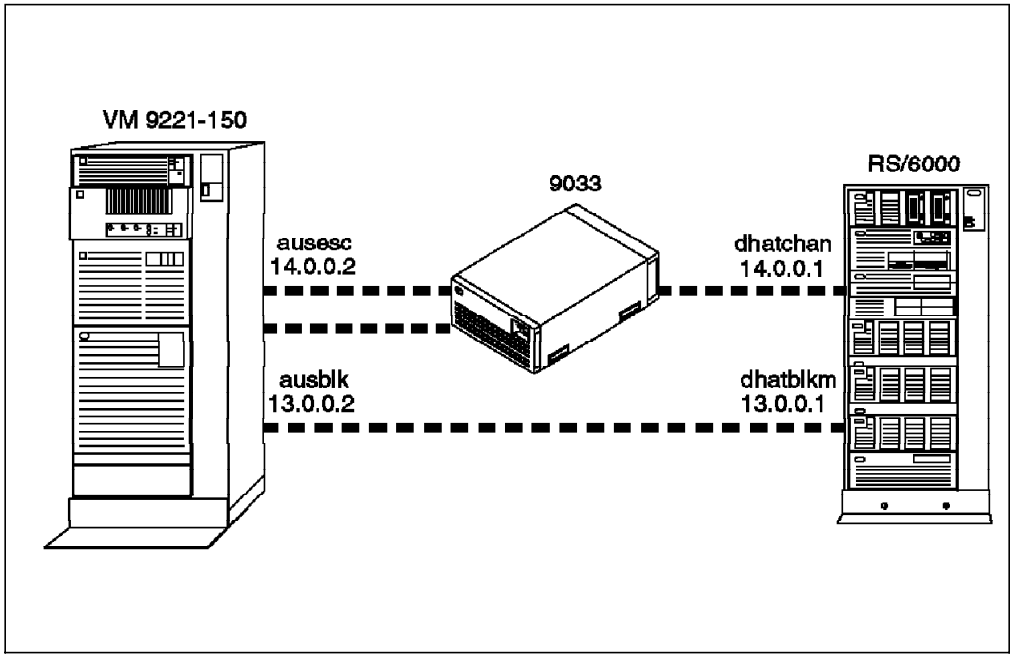


Figure 38. TCP/IP Channel Attach Test Configuration

7.1.1 TCP/IP Configuration Overview

Let′s take a brief overview for the TCP/IP configuration files, TCP/IP server and user setup, and so on, for VM system before you continue. You will refer to these sample definitions all through this chapter.

7.1.2 Server Configuration

We will now look at the setup on the VM host.

7.1.2.1 Typical Sequence of a Server Startup

Figure 39 on page 95 shows the typical sequence of a server setup.

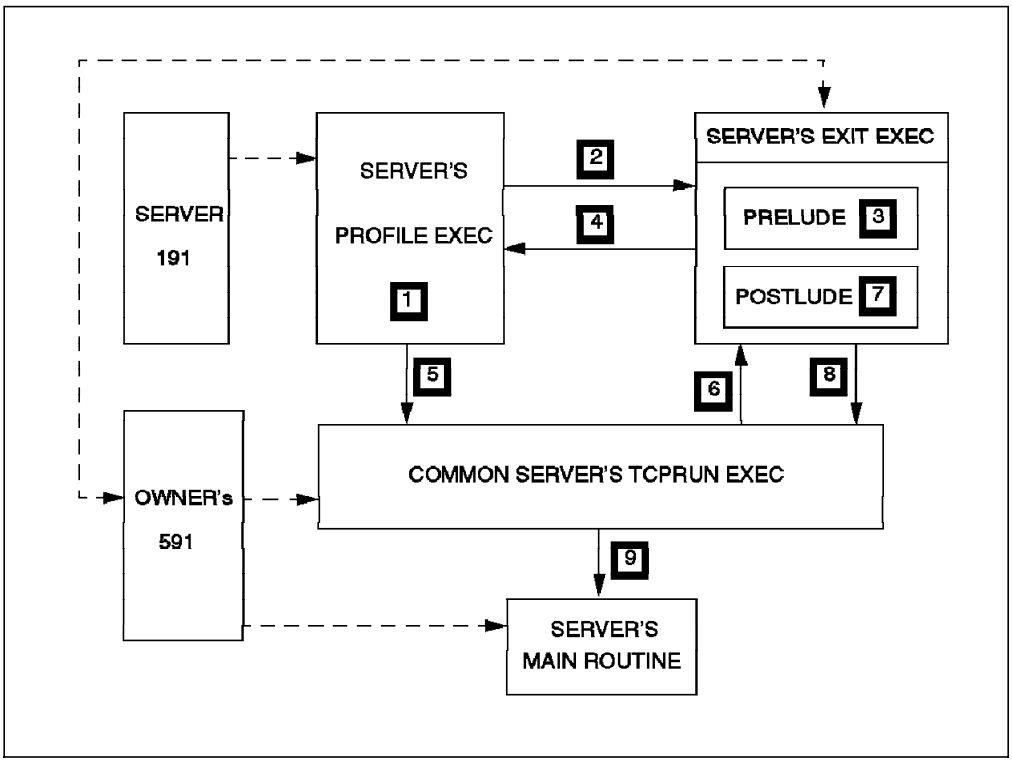


Figure 39. The Sequence of a Server Startup.

1 - When the server is logged on, the PROFILE EXEC read from the server′s 191 minidisk is automatically started.

2 - The PROFILE EXEC calls a new procedure called the server′s *exit EXEC* (read from the common server disk), targeting its PRELUDE subroutine.

3 - The PRELUDE subroutine is responsible for the *LINK* and *ACCESS* to the necessary disks (C/370, SQL/DS, and so on). You should set your PF keys and attach the necessary communications hardware here as well.

4 - After executing the PRELUDE subroutine, the *exit EXEC* returns control to the server′s PROFILE EXEC.

5 - The PROFILE EXEC calls the TCPRUN EXEC procedure which is the common standard procedure used to start up all the servers and which is read from the common server disk.

6 - The TCPRUN EXEC procedure calls the server′s *exit EXEC* once again, this time targeting its POSTLUDE subroutine.

7 - The POSTLUDE subroutine is responsible for setting the options of the server start up (that is, the parameters of the command used to load the server′s main routine).

8 - The POSTLUDE subroutine returns control to the TCPRUN EXEC by supplying the selected options to invoke the server′s module.

9 - Now the TCPRUN EXEC has the necessary environment set up for the server, and the server′s main routine is started.

7.1.2.2 TCP/IP Installation and Maintenance User Directory

The TCPMAINT user definition looks like this:

```
USER  TCPMAINT T3MPPW 16M 32M BG                                    07171642
ACCOUNT 60L                                                         07171642
MACH ESA                                                            07171642
IPL CMS                                                             07171642
CONSOLE 009 3215                                                    07171642
SPOOL 00C 2540 READER *                                             07171642
SPOOL 00D 2540 PUNCH  A                                             07171642
SPOOL 00E 1403 A                                                    07171642
LINK MAINT 190 190 RR                                               07171642
LINK MAINT 19E 19E RR                                               07171642
LINK MAINT 19D 19D RR                                               07171642
MDISK 0191 3380 1456 20 VMESC2 MR                                   07171642
MDISK 0198 3380 1476 10 VMESC2 MR                                   07171642
MDISK 0591 3380 1486 30 VMESC2 MR                                   07171642
MDISK 0592 3380 1516 65 VMESC2 MR                                   07171642
```

Figure 40. TCPMAINT User Directory

7.1.2.3 TCP/IP Server Virtual Machine Directory

```
USER  TCPIP T3MPPW 32M 64M ABG                                      07191516
ACCOUNT 60L                                                         07191516
MACH ESA                                                            07191516
OPTION MAXCONN 1024 DIAG98                                          07191516
IUCV ALLOW                                                          07191516
IUCV ANY PRIORITY                                                   07191516
IUCV *CCS PRIORITY MSGLIMIT 255                                     07191516
IPL CMS                                                             07191516
CONSOLE 009 3215                                                    07191516
SPOOL 00C 2540 READER *                                             07191516
SPOOL 00D 2540 PUNCH  A                                             07191516
SPOOL 00E 1403 A                                                    07191516
DEDICATE 340 340                                                    07191516
DEDICATE 341 341                                                    07191516
LINK MAINT 190 190 RR                                               07191516
LINK MAINT 19E 19E RR                                               07191516
LINK MAINT 19D 19D RR                                               07191516
LINK TCPMAINT 591 591 RR                                            07191516
LINK TCPMAINT 592 592 RR                                            07191516
LINK TCPMAINT 198 198 RR                                            07191516
MDISK 0191 3380 2061 10 VMESC1 MR                                   07191516
```

Figure 41. TCP/IP Server directory

The TCPIP Virtual Machine handles all the hardware and physical interfaces used for TCP/IP communication (except SNA and X.25); it also handles all TCP/IP requests from servers or clients attached to TCP/IP for VM. The Telnet server is implemented within the TCPIP Virtual Machine itself. It uses Console Communication Services (*CCS) to create line mode VM connections for remote users and the Logical Device Support Facility (LDSF) to deal with Conversational Monitor System (CMS) users. This machine uses the CP Virtual Machine Communication Facility (VMCF) function or Communication Control Program (CP)

Inter User Communication Vehicle (IUCV) facility to manage its communication with all servers and users on its VM system.

7.1.2.4 FTPSERVE Virtual Machine

The FTPSERVE Virtual Machine serves client requests coming in via VMCF or IUCV from the TCPIP Virtual Machine. The TCPIP Virtual Machine gets its requests from anywhere in the network and delivers them to the appropriate server. The file server is just an example of many other servers, such as NFS, REXEC and so on. This machine is started by the TCPRUN Exec procedure. TCPRUN Exec is invoked by the Profile Exec residing on the FTPSERVE 191 minidisk.

```
USER  FTPSERVE T3MPPW 6M 8M BDG                                    07180856
ACCOUNT 6OL                                                        07180856
OPTION ACCT                                                        07180856
MACH XC                                                            07180856
IUCV ANY PRIORITY MSGLIMIT 1OO                                     07180856
IPL CMS                                                            07180856
CONSOLE 0O9 3215                                                   07180856
SPOOL 0OC 254O READER *                                            07180856
SPOOL 0OD 254O PUNCH  A                                            07180856
SPOOL 0OE 1403 A                                                   07180856
LINK MAINT 19O 19O RR                                              07180856
LINK MAINT 19E 19E RR                                              07180856
LINK MAINT 19D 19D RR                                              07180856
LINK TCPMAINT 591 591 RR                                           07180856
LINK TCPMAINT 592 592 RR                                           07180856
LINK TCPMAINT 198 198 RR                                           07180856
MDISK 0191 338O 2O76 1O VMESC1 MR                                  07180856
```

Figure 42. FTPSERVE User Directory

7.1.3 Configuration Files

The TCP/IP software is very flexible. At startup time, the configuration parameters for TCP/IP and for all its services (server processes) are read from configuration files. All of them are supplied by IBM as samples, but some of them must be modified by you in order to reflect your environment.

The configuration of IBM TCP/IP Version 2 Release 3 for VM is done by modifying these configuration files. There are a few required files and many optional files, depending on your configuration. The following tables gives you an overview of the configuration files which need your attention.

Table 6 (Page 1 of 2). Configuration Files Used by IBM TCP/IP Version 2 Release 3 for VM.

File Name	Origin	Used By	Purpose
PROFILE TCPIP	TCPMAINT 198	TCPIP	Main configuration file
TCPRUN EXEC	TCPMAINT 591	Servers	Controls the start up of the servers
Exit EXEC	TCPMAINT 591	TCPIP and all servers	Software and hardware setups

Table 6 (Page 2 of 2). Configuration Files Used by IBM TCP/IP Version 2 Release 3 for VM.

File Name	Origin	Used By	Purpose
PROFILE EXEC	TCPIP or server 191	TCPIP and all servers	Do not modify this file
TCPIP DATA	TCPMAINT 592	Clients and servers	Clients and servers configuration file
HOST LOCAL	TCPMAINT 592	All clients and servers	Flat table for name to IP address translation

Please Read This!

Depending on your environment, you will have to create or modify some of these configuration files. Please read the *IBM TCP/IP R3 for VM: Planning and Customization manual* to build these important files.

TCPIP and TCPMAINT are the system default user IDs. You may change the names, but if you do, you must customize related files, such as the Profile Exec on the TCPIP 191 disk and Profile TCPIP on TCPMAINT′s 592 disk.

7.1.4 Customized Files For the Austin Test

The important files for communicating with other systems are TCPIP DATA, PROFILE TCPIP, PROFILE EXEC of TCPIP server, and the HOSTS LOCAL file. These files are for your customization. We will illustrate only the key part of them. For compelete files, please see the Appendix B, “VM TCP/IP Configuration Files” on page 167.

7.1.4.1 TCPIP DATA File on the TCPMAINT 592 Disk

This file, TCPIP DATA, is used to specify configuration information required by TCP/IP client programs. Following are the definitions and some descriptions for the definitions.

```
; TCPIPUSERID specifies the userid of the TCP/IP Virtual Machine.
; TCPIP is the default userid.
TCPIPUSERID TCPIP
;
; HOSTNAME specifies the TCP host name of this VM host.  If not
; specified, the default HOSTNAME will be the node name returned
; by the CMS IDENTIFY command.
;
; For example, if this TCPIP DATA file is shared between two systems,
; OURVM and YOURVM, then the following two lines will define the
; HOSTNAME correctly on each system.
;
HOSTNAME  AUSESC
HOSTNAME  AUSBLK
;AUSESC is the hostname for ESCON interface(14.0.0.2)
;AUSBLK is the hostname for BLKMUX interface(13.0.0.2)
```

Figure 43. TCPIP DATA File

7.1.4.2 Profile Exec on the TCPIP Server's 191 Disk

The following figure is the PROFILE EXEC for TCPIP server with some descriptions.

```
/*********************************************************************/
/* Profile exec for the TCPIP Virtual Machine.                       */
/*   Access the TCP/IP server and client disks.                      */
/*   Invoke the user exit with the argument 'PRELUDE'                */
/*     to allow the network programmer to change his disk            */
/*     configuration or other Virtual Machine options.               */
/*   Invoke the TCPRUN EXEC with arguments specific to this server.  */
/*********************************************************************/
/* Initialize common variables.                                      */
/*********************************************************************/
userexit = 'TCPIPXIT'                  /* Name of user exit EXEC.      */
serverid = 'TCPIP'                     /* TCPRUN routine for this srvr.*/
tcpexit = '*IGNORE*'                   /* Assume no user exit exists.  */
exittype = 'PRELUDE'                   /* User exit argument from here.*/
tcprun = 'TCPRUN'                      /* Name of exec to invoke server*/
owner = 'TCPMAINT'                     /* Assign default ownerid.      */
command = ''                           /* Let TCPRUN or user exit      */
parms = ''                             /* handle these assignments.    */
invoke_tcprun = 1                      /* Hope for the best.           */
stay_online = 0                        /* Assume the worst.            */
globalv = 'globalv SELECT TCPRUN'      /* Nickname for readability.    */

/*********************************************************************/
/* Set up specific to this server before calling TCPRUN.             */
/*********************************************************************/
'ATTACH A50 *'
'ATTACH A51 *'
'ATTACH 410 *'
'ATTACH 411 *'
/* ----------------------------------------------------------------- */
/*    Next line has been modified for APAR PN15969                   */
/* ----------------------------------------------------------------- */
Call Set_Up_Cons owner                 /* Initiate console spooling    */
'Access 198 D'                         /* Access configuration files.  */
'Access 591 E'                         /* Access the server modules.   */
'Access 592 F'                         /* Access the TCP/IP TXTLIBs.   */
```

Figure 44. Profile Exec on TCPIP 191 disk

A50 and A51 are the ESCON channel address. 410 and 411 are BLKMUX channel address. Remember that TCP/IP requires 2 channel addresses; one for writing, the other for reading.

7.1.4.3 PROFILE TCPIP File

During initialization of the TCPIP Virtual Machine, system operation and configuration parameters are read from this file. TCPIP tries to read a file, node_id TCPIP. If not there, a file called PROFILE TCPIP is searched. We customized this file as follows:

```
; Use statements below to alter sizes of free pools.
; See section in this manual on TCP/IP Configuration
; Commands for more information.

ACBPOOLSIZE                 1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                 150
DATABUFFERPOOLSIZE          300 32768
ENVELOPEPOOLSIZE            750
IPROUTEPOOLSIZE             300
LARGEENVELOPEPOOLSIZE       300 8192
RCBPOOLSIZE                 50
SCBPOOLSIZE                 256
SKCBPOOLSIZE                256
SMALLDATABUFFERPOOLSIZE     0
TCBPOOLSIZE                 256
UCBPOOLSIZE                 100

; Flush the arp tables every 5 minutes
ARPAGE 5
;
; Inform the following users of serious errors
INFORM
    OPERATOR TCPMAINT
ENDINFORM

; Obey the following users for restricted commands
OBEY
    OPERATOR TCPMAINT VMFTP1
ENDOBEY

; Autolog the following server machines
AUTOLOG
    FTPSERVE FIREWRK     ; FTP SERVER
;   DSSERV   FIREWRK     ; WDSF Server
    LPSRV23  FIREWRK     ; LP SERVER
;   NAMESRV  FIREWRK     ; DOMAIN NAME SERVER
    PORTMP23 FIREWRK     ; PORTMAP SERVER
    SMTP23   FIREWRK     ; SMTP SERVER
    VMNFS23  FIREWRK     ; NFS SERVER
ENDAUTOLOG

; Reserve the following ports for specific servers
; values from RFC 1060, "Assigned numbers"
PORT
    20 TCP FTPSERVE  NOAUTOLOG ; FTP SERVER
    21 TCP FTPSERVE            ; FTP SERVER
    23 TCP INTCLIEN            ; TELNET SERVER
    25 TCP SMTP23              ; SMTP SERVER
   111 TCP PORTMP23            ; PORTMAP SERVER
   111 UDP PORTMP23            ; PORTMAP SERVER
   515 TCP LPSRV23             ; LP SERVER
  2049 UDP VMNFS23             ; NFS SERVER

                                                MORE...
```

Figure 45. PROFILE TCPIP File, Continued

```
device dhatblk claw 410 AUSBLK DHATBLKM NONE 26 26 4096 4096
LINK ra1 IP 0 dhatblk

device dhat  claw A50 AUSESC DHATCHAN none 26 26 4096 4096
LINK ra2 IP 0 dhat

; the local host's Internet addresses
HOME
   13.0.0.2      ra1
   14.0.0.2      ra2

; ROUTING INFORMATION (IF YOU ARE NOT USING THE ROUTED SERVER)
 GATEWAY
;  Network      First hop     Driver  Packet size  Subn mask  Subn value
;  1             =             tcpip   DEFAULTSIZE  0.255.255.128 0.0.0.0
  13            =             ra1      4096        0
  14            =             ra2      4096        0
; Start all the interface
  start dhatblk
  start dhat
```

Figure 46. PROFILE TCPIP File, Continued

As far as IBM TCP/IP Version 2 Release 3 for VM is concerned, some definitions are needed in the PROFILE TCPIP. Your definition must match the syntax.

- A *DEVICE* statement:

  Syntax

  DEVICE device_name CLAW address HOST PSCA NOTE read_buffers write_buffers read_size write_size

  - **device_name**: Name of the device. Maximum length is 16 characters. The same name must be specified in the link statement.
  - **CLAW**: Keyword indicating that the CLAW mode is used.
  - **address**: The hexadecimal subchannel address for communication. It must be even, and automatically a pair is assigned.
  - **HOST**: Name of the host system. This name must match the Host Name field of the *Change/Show Characteristics of a 370 Parallel Channel Adapter* SMIT panel.
  - **PSCA**: Name of the system unit. This name must match the Adapter Name field of the *Change/Show Characteristics of a 370 Parallel Channel Adapter* SMIT panel.
  - **NONE**: Reserved.
  - **read_buffers**: Decimal number of buffers to allocate to the read channel program. The default is 20. Each of the buffers uses real storage.
  - **write_buffers**: Decimal number of buffers to allocate to the write channel program. The default is 20. Each of the buffers uses real storage.
  - **read_size**: Size of the read buffers. It must be less than or equal to the transmit buffer size specified in the system unit. The default is 4096.
  - **write_size**: Size of the read buffers. It must be less than or equal to the receive buffer size specified in the system unit. The default is 4096.

- A *LINK* statement:

  Syntax

  LINK link_name IP 0 device_name

  - **link_name**: Unique assigned link name
  - **IP**: Keyword
  - **device_name**: Comes from the corresponding device statement

- A *GATEWAY* statement:

  Syntax

  GATEWAY network first_hop link_name max_packet_size ...

  - The GATEWAY statement is the same as for any other device or link.
  - **max_packet_size**: The value must not exceed the write_size on the DEVICE statement.

7.1.4.4 HOSTS LOCAL File

The HOSTS LOCAL file is regarded as the same as /etc/hosts on the RS/6000. This file contains the names and the Internet addresses for the hosts you use most often. It is the input file for makesite command which generates the HOSTS SITEINFO and HOSTS ADDRINFO to TCPMAINT's 592 minidisk.

```
; HOSTS LOCAL
; -----------
;
; Note:  The NET and GATEWAY statements are not used by the TCP/IP for
;        VM applications.  However, some socket calls require the NET
;        entries.  For added performance, if your programs do not need
;        the NET and GATEWAY statements, delete them before running
;        the MAKESITE program.
;
;
 HOST : 13.0.0.1   : DHATBLKM ::::
 HOST : 13.0.0.2   : AUSBLK   ::::
 HOST : 14.0.0.1   : DHATCHAN ::::
 HOST : 14.0.0.2   : AUSESC   ::::
;HOST : 9.3.13.131 : AUSHAT2  ::::
;
```

Figure 47. HOSTS LOCAL File

7.2 ESCON Connection Test by TCP/IP

This section describes the ESCON connection from AIX to VM using TCP/IP.

7.2.1 VM Configuration for ESCON

The following subsections discuss the options that need to be defined on the VM machine for ESCON connection using TCP/IP.

7.2.1.1 IOCP Definition

In order to make the channel connection between a mainframe and a workstation, you need the IOCP definition for the workstation on the mainframe IOCDS (Input Output Configuration Data Set). Below is the IOCP definition for the channel and the device (workstation) for ESCON channel.

```
CHPID    PATH=((25)),TYPE=CNC,SWITCH=01
CNTLUNIT CUNUMBR=0250,PATH=(25),UNITADD=((50,032)),UNIT=SCTC,LINK=(CA)
IODEVICE ADDRESS=(A50,016),CUNUMBR=(0250),UNIT=SCTC
```

7.2.1.2 Real I/O Device Definition

We are using VM/ESA Release 2,in which case you don't need to define the device to VM CP (Control Program). If you are using VM/ESA R1 or VM/XA, you must define the device in the HCPRIO Assemble file as follows:

```
RDEVICE DEVNO=(A50,32),DEVTYPE=CTCA(or 3088)
```

If you are defining the device in HCPRIO or DMKRIO(VM/SP), you must generate new CP nucleus and reIPL the VM system.

7.2.1.3 VM/CMS User ID for a TCP/IP User

There are some special requirements for a user ID that enable it to use the TCP/IP client programs. The following example shows a regular VM/CMS user with access to TCP/IP.

Directory Entry "USER1 DIRECT"

```
USER USER1 passwd 4M 6M G
 ACCOUNT 1 USER1
 OPTION ECMODE
 IPL CMS PARM AUTOLOG
 CONSOLE 009 3215 T OPERATOR
 SPOOL 00C 2540 READER *
 SPOOL 00D 3540 PUNCH A
 SPOOL 00E 1403 A
 LINK MAINT    190 190 RR
 LINK MAINT    19E 19E RR
 LINK TCPMAINT 592 592 RR                          1
 MDISK 191 3380 200 010 VMESC1 MR Rpw Wpw Mpw      2
```

1. All TCP/IP client programs that are important to the regular user reside on this minidisk.
2. If the VM/CMS user wants to allow remote users to access this 191 minidisk via FTP or the Network File System (NFS), minidisk passwords are required. (Not required if you use the Resource Access Control Facility, RACF.)

PROFILE EXEC File

```
/*****************************************************/
/* example profile exec for a TCP/IP user            */
/*****************************************************/
trace e
'cp set pf12    retrieve'
'cp spool con  start to *'
'cp spool print rscs cl a'
'cp tag dev prt aus20058 itso 99 sysout=a'
'acc 592 r'                                          1
'set ldrtbls 10'                                     2
'cruntime'                                           3
'global loadlib edclink'                             4
'global txtlib rpclib commtxt ibmlib cmslib edcbase' 4
```

1 This statement accesses the TCP/IP minidisk. A link statement in the directory, or a previous link statement in the profile, is required.

2 The default number of loader tables might be too low for some clients (FTP for example). To do FTP for a big file, you may need 20 or more.

3 The cruntime procedure establishes the link and access to the C/370 minidisk.

4 Some client programs, or your own written TCP/IP applications, may need access to the C/370 libraries. This is optional for a regular TCP/IP user, but it is required for TCP/IP programming and the execution of the TCP/IP API programs.

7.2.1.4 RACF Considerations

Effective user verification can be offered by the Resource Access Control Facility (RACF). RACF uses the user ID and a system-encrypted password to perform its identification and verification.

This section only discusses the TCP/IP servers that can use RACF to validate user access and the way they should be customized to use RACF.

FTP Interface to RACF: If you use VM/SP with RACF, additional customization is required. RACF has an alternate user ID feature, which allows the FTP server to act as a surrogate (alternate) user ID for other users. This means that the FTP server can access the required disk on behalf of the user ID which is entered during the FTP session establishment. If RACF is installed on your VM system, you must modify the FTP server exit, FTPDEXIT EXEC, to include the 'RACF' option:

```
00085 parms = 'RACF'              /* Optional server arguments */
```

In order to authorize FTPSERVE to access a user's minidisk, you should use the FTPPERM EXEC, or issue the appropriate RACF commands.

Note: The FTPPERM EXEC assumes that a VMBATCH RACF class exists and that FTPSERVE is defined to this class (the command is RDEF VMBATCH FTPSERVE UACC(NONE) Please check this with the RACF administrator. Actually, the FTPPERM EXEC issues the following RACF commands:

```
PERMIT userid 'CLASS(VMBATCH)' access 'ID('vm')'
```

The access keyword may be either:

- DELETE (removes the surrogate capability for FTPSERVE). You can also use the FTPPERM DEL userid command.
- ACCESS (CONTROL) (allows FTPSERVE to act as a surrogate user ID). You can also use the FTPPERM ADD userid command.

This command is included in a loop so that the vm keyword can stand for all the FTP servers that you defined in your system. You must code all the names of the different FTP servers in the FTPPERM EXEC:

```
00016 ftpserve = 'FTPSERVE FTPSERV1 FTPSERV2'
```

All these servers should be defined to the class VMBATCH.

When the RACF option is chosen, when FTPSERVE is started, the FTPDRACF EXEC is called. The EXEC calls the RACFLINK EXEC with the INIT option. Depending on the VM/SP release level, RACFLINK (INIT will copy either SURROGAT EXEC or ALTERNAT EXEC to $ALTUSER EXEC, which will be used to perform either the surrogate or the alternate user ID. The ALTERNAT MODULE and the RPIVAL MODULE are loaded. Finally, the FTP server is started with the command srvrftp racf.

When a TCP/IP user establishes a connection in order to transfer files with the VM FTP server, a user ID and a password must be entered. The user ID and the password will be validated via the VALIDATE EXEC which calls the PRIVAL MODULE. If the user ID and password are valid, then the RACFLINK EXEC is called to perform the link.

The following is the result of a RACF query. The class queried is VMBATCH which is the class required to act as alternate. In this example, FTPSERVE is authorized to act as alternate for TCPMAINT.

```
id
FTPSERVE AT RACYESA  VIA RJE    06/17/92 09:21:30 EDT        WEDNESDAY
Ready; T=0.01/0.01 09:21:30
ICH70001I FTPSERVE LAST ACCESS AT 09:15:48 ON WEDNESDAY, JUNE 17, 1992
RPITMP001I RACF/VM SESSION ESTABLISHED. TO TERMINATE ENTER "END"
RPITMP002I ENTER RACF COMMAND OR "END" TO EXIT
rl vmbatch * all

CLASS      NAME
-----      ----
VMBATCH    TCPMAINT

LEVEL  OWNER      UNIVERSAL ACCESS  YOUR ACCESS  WARNING
-----  --------   ----------------  -----------  -------
 00    TCPMAINT        NONE            CONTROL    NO
```

7.2.2 AIX Configuration for ESCON TCP/IP

The AIX configuration for ESCON TCP/IP for VM is equivalent to the set up for MVS. Chapter 3, "RISC System/6000 S/390 ESCON Channel Connection" on page 15 and the subsequent sections show the setup in detail. Sections 3.1.2, "Planning Information Required" on page 17 and the set up 3.1.3, "ESCON Channel Implementation" on page 19 are equivalent. Figure 10 on page 28, Figure 11 on page 29 and Figure 12 on page 29 include the VM definitions for

reference. The VM ESCON channel was taken from MCA slot 16 with an IP address on the mainframe of 14.0.0.2. (host name AUSESC) and 14.0.0.1. on the RS/6000 (host name DHATCHAN).

The VM system passes through an ESCD for ESCON, and therefore careful reference to the subchannel addresses was made. This was explained in 3.1.3.9, “Defining ESCON Subchannels” on page 24.

7.2.3 FTP Test Between the VM Host and the RS/6000

Our test for ESCON has been done in four categories.

- FTP get file from RS/6000 to VM user minidisk (191) initiated by the VM side
- FTP put file from VM user minidisk (191) to RS/6000 initiated by the VM side
- FTP get file from VM user minidisk (191) to RS/6000 initiated by the RS/6000 side
- FTP put file from RS/6000 to VM user minidisk (191) initiated by the RS/6000 side

7.2.3.1 FTP Get File by a VM User

A file transfer is invoked using the `ftp` command. The Internet address, or a symbolic name, can be used as a parameter to indicate the remote system. FTP then tries to establish a TCP connection to the remote TCP/IP host on the reserved port, 21.The remote FTP server sends a welcome message and asks for a valid user ID and password. If the password is correct, the control connection is established, and the user can then use all the FTP subcommands and interact with the remote server.

Figure 48 on page 107 is the user console message resulting from the FTP get command.

```
Ready; T=0.32/0.40 15:04:14
ftp dhatchan
VM TCP/IP FTP V2R3
Connecting to DHATCHAN 14.0.0.1, port 21
220 dhatrisk FTP server (Version 4.29 Tue Aug 25 14:47:35 PDT 1992) ready.
USER (identify yourself to the host):
hyuk
>>>USER hyuk
331 Password required for hyuk.
Password:

>>>PASS ********
230 User hyuk logged in.
Command:
get /tmp/ftp/ftp1.test ftp1.test
>>>PORT 14,0,0,2,4,49
200 PORT command successful.
>>>RETR /tmp/ftp/ftp1.test
150 Opening ASCII mode data connection for /tmp/ftp/ftp1.test (37107547 bytes).
2752452 bytes transferred.
5504982 bytes transferred.
8224710 bytes transferred.
10944470 bytes transferred.
13696945 bytes transferred.
16416763 bytes transferred.
19070901 bytes transferred.
21790645 bytes transferred.
24477629 bytes transferred.
27230172 bytes transferred.
29949930 bytes transferred.
32669678 bytes transferred.
35422165 bytes transferred.
226 Transfer complete.
37565937 bytes transferred in 138.707 seconds. Transfer rate 270.83 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
Ready; T=52.35/58.37 15:08:07
```

Figure 48. FTP Get File Test from VM User (ESCON)

DHATCHAN is the hostname of the RS/6000 with IP address 14.0.0.1. The TCPIP machine searchs the HOSTS LOCAL file and connects user to the IP address.

We have monitored the system performance during this FTP job by VM RTM (Real Time Monitor), which displayed the system activity in Figure 49 on page 108. There were no other heavy users on the system during this test.

```
 VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 15:05:50 END 15:06:21
           *
 <USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
 VMFTP1     42 4.2  37   34 .00  284  286   .1    0   100     4M VUB,---,SIMW
 TCPIP      25  10  15   27 .00 2523 2955   .0  381   100    32M VUS,QDS,DISP
 SYSTEM    2.8 2.8 .00  .00 .00    0  852   .0  553 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                      *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
01F6 3380 VMESC2    1027  34 .00 .00   .00   37  1  32    9    0    6    3  12
0A51 CTCA            838  27 .00 .00   .00    0  1 1.2    0    0    0    0 .56
0502 GRAF              6   0 .00 .00   .00    0  1 .00    0    0    0    0 .00
0501 GRAF              2   0 .00 .00   .00    0  1 .00    0    0    0    0 .00

 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   71  15  54  29 2.8 .00  235  71  0 .00 .00    0  18    0   0    0 0.630
<-..   15 3.6  11  85 .71 .00   53  77 .. .00 .00    0  21    0   0    0 1.286
```

Figure 49. VM FTP Get File RTM Screen Output (ESCON)

What you will be interested in are %CPU, IOREQST, SHARE, and WSS (Working Set size). We will deal with them later in this chapter.

We can see that the FTP user VMFTP1 consumes the most CPU cycle, and the IOREQST for DASD 1F6, in which the VMFTP1 191 minidisk resides, has the biggest number, and the CTCA channel, A51, is very busy.

7.2.3.2 FTP Put File by a VM User

Figure 50 on page 109 is the console message of FTP put command initiated by an FTP user.

```
Ready; T=0.01/0.01 14:59:13
ftp dhatchan
VM TCP/IP FTP V2R3
Connecting to DHATCHAN 14.0.0.1, port 21
220 dhatrisk FTP server (Version 4.29 Tue Aug 25 14:47:35 PDT 1992) ready.
USER (identify yourself to the host):
hyuk
>>>USER hyuk
331 Password required for hyuk.
Password:

>>>PASS ********
230 User hyuk logged in.
Command:
put ftp1.test /tmp/ftp/ftp1.test
>>>SITE VARrecfm
500 'SITE VARRECFM': command not understood.
>>>PORT 14,0,0,2,4,46
200 PORT command successful.
>>>STOR /tmp/ftp/ftp1.test
150 Opening ASCII mode data connection for /tmp/ftp/ftp1.test.
13369342 bytes transferred.
27033598 bytes transferred.
226 Transfer complete.
37565937 bytes transferred in 55.713 seconds. Transfer rate 674.28 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
```

Figure 50. FTP Put File by VM FTP User (ESCON)

As you see in Figure 50, the transfer rate of put is a lot faster than that of get.

During this put test, we can see the TCPIP machine consumes the most CPU cycle, which means the machine involves the file transfer very actively, as the following RTM screen shows.

```
 VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 15:01:17 END 15:01:47
            *
 <USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
 TCPIP      51  21  29   45 .00 2523 2955   .0  381   100    32M VUS,QDS,DISP
 VMFTP1     25 5.1  19   22 .00  278  288   .2    0   100     4M VUB,---,SIMW
 SYSTEM    4.4 4.4 .00  .00 .00    0  852   .0  553 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                    *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
0A51 CTCA          1347  44 .00 .00   .00    0  1 7.4    1    0    0    1 6.4
01F6 3380 VMESC2    666  22 .00 .00   .00   37  1  53   24    0   12   12  26
0502 GRAF            11   0 .00 .00   .00    0  1 .00    0    0    0    0 .00
0501 GRAF             2   0 .00 .00   .00    0  1 .00    0    0    0    0 .00

 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   79  26  49  21 4.4 .00  375  89  0 .00 .00    0  57    0   0    0 0.352
<-..   14 3.4  10  86 .68 .00   51  77 .. .00 .00    0  21    0   0    0 1.218
```

Figure 51. VM FTP Put file RTM screen output(ESCON)

7.2.3.3 FTP Get file by an RS/6000 user

The following is the console message of an FTP get command initiated by an RS/6000 FTP user.

```
# ftp ausesc
Connected to ausesc.
220-FTPSERVE IBM VM V2R3 at AUSESC.AUSTIN.IBM.COM, 14:29:47 CDT WEDNESDAY 08
/09/95
220 Connection will close if idle for more than 5 minutes.
Name (ausesc:toshio): vmftp1
331 Send password please.
Password:
230 VMFTP1 logged in; working directory = VMFTP1 191
ftp> get ftp1.test
200 Port request OK.
150 Sending file 'ftp1.test'
250 Transfer completed successfully.
37568031 bytes received in 56.54 seconds (648.9 Kbytes/s)
ftp> quit
221 Quit command received. Goodbye.
#
```

Figure 52. FTP Get File from RS/6000 FTP User (ESCON)

From the test result, we can tell the transfer rate is very similar to that of an FTP put, (674.3 Kbytes/sec), initiated by the VM side. But the active users participating in the FTP are somewhat different, as you see in the following RTM screen for this FTP test.

```
 VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 14:30:56 END 14:31:27
           *
 <USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
 TCPIP      44  18  26   31 .00 2473 2905   .0  381   100    32M VUS,QDS,DORM
 FTPSERVE   31 4.8  27   20 .00  163  175   .0  369   100     6M VUC,DSC,IDLE
 SYSTEM    3.5 3.5 .00  .00 .00    0  924   .0  553 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                       *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
0A51 CTCA            977  31 .00 .00   .00    0  1 6.9    2    0    0    1 6.1
01F6 3380 VMESC2     681  21 .00 .00   .00   34  1  47   21    0    9   12  26
0501 GRAF              2   0 .00 .00   .00    0  1 .00    0    0    0    0 .00


 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   82  24  55  18 3.5 .00  294  86  0 .00 .00    0  13    0   0    0 1.066
<-..   18 4.2  13  82 .83 .00   63  76 .. .00 .00    0  22    0   0    0 1.478
```

Figure 53. RTM Screen on FTP Get File from RS/6000 FTP User (ESCON)

The FTPSERVE machine on VM takes part in this file transfer, while the TCPIP machine is still using the most CPU. The A51 channel is the busiest device. The %CPU is 82 percent which is the highest one from the ESCON FTP test.

7.2.3.4 FTP Put File by an RS/6000 User

As the last test for ESCON FTP test, we tried to upload the large file to the VM host as follows.

```
# ftp ausesc
Connected to ausesc.
220-FTPSERVE IBM VM V2R3 at AUSESC.AUSTIN.IBM.COM, 14:38:27 CDT WEDNESDAY 08
/09/95
220 Connection will close if idle for more than 5 minutes.
Name (ausesc:toshio): vmftp1
331 Send password please.
Password:
230 VMFTP1 logged in; working directory = VMFTP1 191
ftp> put ftp1.test
200 Port request OK.
150 Storing file 'ftp1.test'
250 Transfer completed successfully.
37568031 bytes sent in 141.3 seconds (259.6 Kbytes/s)
ftp> quit
221 Quit command received. Goodbye.
#
```

Figure 54. FTP Put File from RS/6000 FTP User (ESCON)

The transfer rate for this upload test is 259.6 Kbytes/sec, which is similar to the upload test (VM FTP get) initiated by VM FTP user.

Now, on the RTM screen, we discovered that the FTPSERVE machine is taking the lead. The FTP server on RS/6000 tosses the file transfer, and the VM FTPSERVE machine has the most control of writing the file on the VM user's 191 disk.

```
 VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 14:39:32 END 14:40:03
             *
 <USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
 FTPSERVE   50 4.1  46   33 .00  177  179   .0  369   100     6M VUC,DSC,SIMW
 TCPIP      21 8.8  12   20 .00 2473 2905   .0  381   100    32M VUS,QDS,DISP
 SYSTEM    2.6 2.6 .00  .00 .00    0  927   .0  553 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                      *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
01F6 3380 VMESC2    1019  33 .00 .00   .00   34  1  26    7    0    4    3  12
0A51 CTCA            617  20 .00 .00   .00    0  1 .88    0    0    0    0 .40
0501 GRAF              2   0 .00 .00   .00    0  1 .00    0    0    0    0 .00


 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   75  13  59  25 2.6 .00  199  70  0 .00 .00    0  55    0   0    0 0.698
<-..   16 3.7  11  84 .74 .00   56  76 .. .00 .00    0  21    0   0    0 1.319
```

Figure 55. RTM Screen on FTP Put File from RS/6000 FTP User (ESCON)

7.3 BLKMUX Connection Test by TCP/IP

7.3.1 VM Configuration for BLKMUX

The AIX configuration for BLKMUX TCP/IP for VM is equivalent to the set up for MVS. Please refer to 4.1.2.6, “Defining BLKMUX Subchannels” on page 40 and to the subsequent sections showing the set up in detail. Section 4.1.1, “Planning Information Required” on page 35 and section 4.1.2, “BLKMUX Channel Implementation” on page 37 are equivalent.

7.3.1.1 IOCP Definition

In order to make the BLKMUX channel connection between a mainframe and a workstation, you need the IOCP definition for the workstation on mainframe IOCDS, as is the case in ESCON. Below is the IOCP definition for the channel and the device (workstation) for the BLKMUX channel.

```
CHPID    PATH=((20,04)),TYPE=BL
CNTLUNIT CUNUMBR=0200,PATH=(20),UNITADD=((00,032)),SHARED=N,
             PROTOCL=S4,UNIT=3088
IODEVICE ADDRESS=(400,032),CUNUMBR=(0200),UNIT=CTC
```

7.3.1.2 Real I/O Device Definition

We use VM/ESA Release 2, in which case you don′t need to define the d evice to the VM CP(Control Program). If you are using VM/ESA R1 or VM/XA, you must define the device in the HCPRIO Assemble file as follows.

```
RDEVICE DEVNO=(400,32),DEVTYPE=CTCA(or 3088)
```

If you are defining the device in HCPRIO or DMKRIO (VM/SP), you must generate THE new CP nucleus and reIPL the VM system.

Other definitions for the TCP/IP user ID directory, user Profile Exec and RACF considerations are the same.

7.3.2 RS/6000 Configuration for BLKMUX Connection

Please refer to section Chapter 4, “ RS/6000 S/370 Block Multiplexer Channel Connection” on page 33.

7.3.3 FTP Test Between the VM Host and the RS/6000

The Austin test for BLKMUX was done in four categories as was the previous ESCON test.

- FTP get file from the RS/6000 to the VM user minidisk (191) initiated by the VM side
- FTP put file from the VM user minidisk (191) to the RS/6000 initiated by the VM side
- FTP get file from the VM user minidisk (191) to the the RS/6000 initiated by the RS/6000 side
- FTP put file from the RS/6000 to the VM user minidisk (191) initiated by the RS/6000 side

The FTP test results for BLKMUX was very similar to that of ESCON. The following test screens are shown without detailed explanations in an effort to reduce redundancy.

7.3.3.1 FTP Get File by a VM User

The following screen is the user console message resulting from the FTP get command.

```
Ready; T=0.31/0.36 16:04:52
ftp dhatblkm
VM TCP/IP FTP V2R3
Connecting to DHATBLKM 13.0.0.1, port 21
220 dhatrisk FTP server (Version 4.29 Tue Aug 25 14:47:35 PDT 1992) ready.
USER (identify yourself to the host):
hyuk
>>>USER hyuk
331 Password required for hyuk.
Password:

>>>PASS ********
230 User hyuk logged in.
Command:
get /tmp/ftp/ftp2.test ftp2.test
>>>PORT 13,0,0,2,4,3
200 PORT command successful.
>>>RETR /tmp/ftp/ftp2.test
150 Opening ASCII mode data connection for /tmp/ftp/ftp2.test (37105453 bytes).
2686956 bytes transferred.
5373904 bytes transferred.
8093649 bytes transferred.
10584005 bytes transferred.
13238201 bytes transferred.
15925239 bytes transferred.
17760181 bytes transferred.
20348912 bytes transferred.
22970326 bytes transferred.
25591775 bytes transferred.
28278745 bytes transferred.
30965720 bytes transferred.
33652688 bytes transferred.
36306926 bytes transferred.
226 Transfer complete.
37563843 bytes transferred in 145.760 seconds. Transfer rate 257.71 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
Ready; T=52.92/59.03 16:08:47
```

Figure 56. FTP Get File by VM FTP User (BLKMUX)

As you see in the following RTM screen for this file transfer, the user, VMFTP1, is doing a lot more work than other virtual machines, as seen in the previous VM FTP get test for ESCON.

```
VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 16:06:33 END 16:07:04
          *
<USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
VMFTP1     40 4.1  36   32 .00  283  285   .4    0   100     4M VUB,---,SIMW
TCPIP      26  10  15   32 .00 2526 2959   .0    0   100    32M VUS,DSC,DISP
SYSTEM    3.1 3.1 .00  .00 .00    0  605   .0  558 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                      *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
01F6 3380 VMESC2     986  32 .00 .00   .00   37  1  30    9    0    5    3  11
0411 CTCA            981  32 .00 .00   .00    0  1 1.4    0    0    0    0 .69
0502 GRAF             10   0 .00 .00   .00    0  1 .00    0    0    0    0 .00
0501 GRAF              2   0 .00 .00   .00    0  1 .00    0    0    0    0 .00

 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   71  15  53  29 3.2 .00  250  73  0 .00 .00    0  51    0   0    0 0.825
<-..   22 6.1  15  78 1.1 .00   84  81 .. .00 .00    0  46    3   0    0 1.825
```

Figure 57. VM FTP Get File RTM Screen Output (BLKMUX)

7.3.3.2 FTP Put File by a VM User

Please take a look at the following screen resulting from uploading the file by VM FTP user.

```
Ready; T=0.01/0.01 16:01:26
ftp dhatblkm
VM TCP/IP FTP V2R3
Connecting to DHATBLKM 13.0.0.1, port 21
220 dhatrisk FTP server (Version 4.29 Tue Aug 25 14:47:35 PDT 1992) ready.
USER (identify yourself to the host):
hyuk
>>>USER hyuk
331 Password required for hyuk.
Password:

>>>PASS ********
230 User hyuk logged in.
Command:
put ftp1.test /tmp/ftp/ftp2.test
>>>SITE VARrecfm
500 'SITE VARRECFM': command not understood.
>>>PORT 13,0,0,2,4,1
200 PORT command successful.
>>>STOR /tmp/ftp/ftp2.test
150 Opening ASCII mode data connection for /tmp/ftp/ftp2.test.
13434878 bytes transferred.
26804222 bytes transferred.
226 Transfer complete.
37563843 bytes transferred in 55.805 seconds. Transfer rate 673.13 Kbytes/sec.
Command:
quit
>>>QUIT
221 Goodbye.
Ready; T=11.15/14.12 16:04:28
```

Figure 58. FTP Put File from VM FTP User (BLKMUX)

As you see the following RTM screen for this file transfer, the TCPIP server is doing a lot of work, and the CTCA device is very busy, as seen in the previous VM FTP put test for ESCON.

```
 VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 16:03:30 END 16:04:01
           *
 <USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
 TCPIP      45  19  26   39 .00 2506 2952   .0    0   100    32M VUS,DSC,DISP
 VMFTP1     22 4.6  18   20 .00  267  281   .5    0   100     4M VUB,---,IDLE
 SYSTEM    3.4 3.4 .00  .00 .00    0  548   .0  558 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                     *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
0411 CTCA          1212  40 .00 .00   .00    0  1  18    4    0    0    4  17
01F6 3380 VMESC2    663  22 .00 .00   .00   37  1  49   23    0   10   12  26
0502 GRAF             8   0 .00 .00   .00    0  1 .00    0    0    0    0 .00
01CF 3380 VMESC3      2   0 .00 .00   .00    6  1 .18   29    7   27    2 .01

 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   75  25  46  25 3.5 .00  300  88  0 .00 .00    0  46    0   0    0 0.989
<-..   21 7.1  13  79 1.2 .00   87  87 .. .00 .00    0  46    5   0    0 0.741
```

Figure 59. VM FTP Put File RTM Screen Output (BLKMUX)

7.3.3.3 FTP Get File by an RS/6000 FTP User

Please take a look at the following screen resulting from downloading the file by an AIX user.

```
# ftp 13.0.0.2
Connected to 13.0.0.2.
220-FTPSRV23 IBM VM V2R3 at AUSESC.AUSTIN.IBM.COM, 17:37:07 CDT WEDNESDAY 08
/09/95
220 Connection will close if idle for more than 5 minutes.
331 Send password please.
230-VMFTP1 logged in; working directory = VMFTP1 191 (ReadOnly)
230 write access currently unavailable due to other links
ftp> dir
200 Port request OK.
125 List started OK
FTP1     TEST     V         80     458390       9171  8/09/95 17:32:14 FTP19
1
HYUK     TEST1    V         80          9          1  7/20/95 13:48:04 FTP19
1
KOH      PRO      V        110       2021         27  8/01/95 18:06:24 FTP19
1
PROFILE  EXEC     V         24          9          1  8/09/95 15:20:32 FTP19
1
TEST     FTP      F         80          9          1  7/20/95 13:43:21 FTP19
1
VMFTP    BLKMUX   F        132        105          4  8/09/95 16:10:02 FTP19
1
VMFTP    ESCON    F        132         97          4  8/09/95 15:08:59 FTP19
1
VMFTP1   NETRC    V         42          2          1  8/07/95 14:20:17 FTP19
1
250 List completed successfully.
ftp> get ftp2.test
200 Port request OK.
150 Sending file 'ftp2.test'
250 Transfer completed successfully.
37561749 bytes received in 57.93 seconds (633.2 Kbytes/s)
ftp> quit
221 Quit command received. Goodbye.
```

Figure 60. FTP Get File from RS/6000 FTP User Screen (BLKMUX)

When you get the user ID and password authorization, you can look at the VM user's files on the 191 disk. You can choose any one of them to download to the hard disk. The figure below shows the system activity resulting from this download.

```
 VM/ESA CPU9221 SERIAL 000774   30M DATE 08/09/95 START 17:38:32 END 17:39:03
           *
 <USERID> %CPU %CP %EM ISEC PAG  WSS  RES   UR PGES SHARE VMSIZE TYP,CHR,STAT
 TCPIP      44  18  26   29 .00 2251 2683   .0  541   100    32M VUS,DSC,DISP
 FTPSERVE   32 4.8  27   21 .00  432  450   .0    0   100     6M VUC,DSC,SIMW
 SYSTEM    3.4 3.4 .00  .00 .00    0  721   .0  555 .....     2G SYS,

<--- DEVICE ---> <----- DEVICE  RDEV  DATA ------> <-- MEASUREMENT FACILITY ->
                      *
 DEV TYPE VOLSER IOREQST SEC  %Q %ER R %LK  LNK PA %UT  ACC  FPT  DCT   CN %CN
0411 CTCA           878  29 .00 .00   .00    0  1  18    6    0    0    6  17
01F6 3380 VMESC2    689  22 .00 .00   .00   34  1  42   19    0    7   11  26
0502 GRAF            15   0 .00 .00   .00    0  1 .00    0    0    0    0 .00
02AB 3380 VMESC1     12   0 .00 .00   .00   67  1 .29    7    1    5    2 .08

 <----------- CPU STATISTICS --------> <-- VECTOR ---> <STORAGE><XSTORE>
  NC %CPU %US %EM %WT %SY %SP  XSI %SC NV %VT %OT RSTR %ST PSEC %XS XSEC   TTM
-> 1   83  24  56  17 3.4 .00  290  83  0 .00 .00    0  49    0   0    0 0.750
<-..  6.1 1.6 4.2  94 .38 .00   25  77 .. .00 .00    0  23    0   0    0 4.390
```

Figure 61. RTM Screen on FTP Get File from RS/6000 FTP User (BLKMUX)

7.3.3.4 FTP Put File by an RS/6000 FTP User

Please take a look at the following screen which resulted from uploading the file by an AIX user.

```
dhatrisk:/u/toshio> ftp 13.0.0.2
Connected to 13.0.0.2.
220-FTPSRV23 IBM VM V2R3 at AUSESC.AUSTIN.IBM.COM, 09:59:39 CDT FRIDAY 08/18/95
220 Connection will close if idle for more than 5 minutes.
Name (13.0.0.2:toshio): vmftp1
331 Send password please.
Password:
230 VMFTP1 logged in; working directory = VMFTP1 191
ftp> put /tmp/ftp/large1 ftp1.test
200 Port request OK.
150 Storing file 'ftp1.test'
250 Transfer completed successfully.
37579288 bytes sent in 149.4 seconds (245.6 Kbytes/s)
ftp>
```

Figure 62. FTP Put File from RS/6000 FTP User Screen (BLKMUX)

The following RTM screen displays the system activity for this upload.

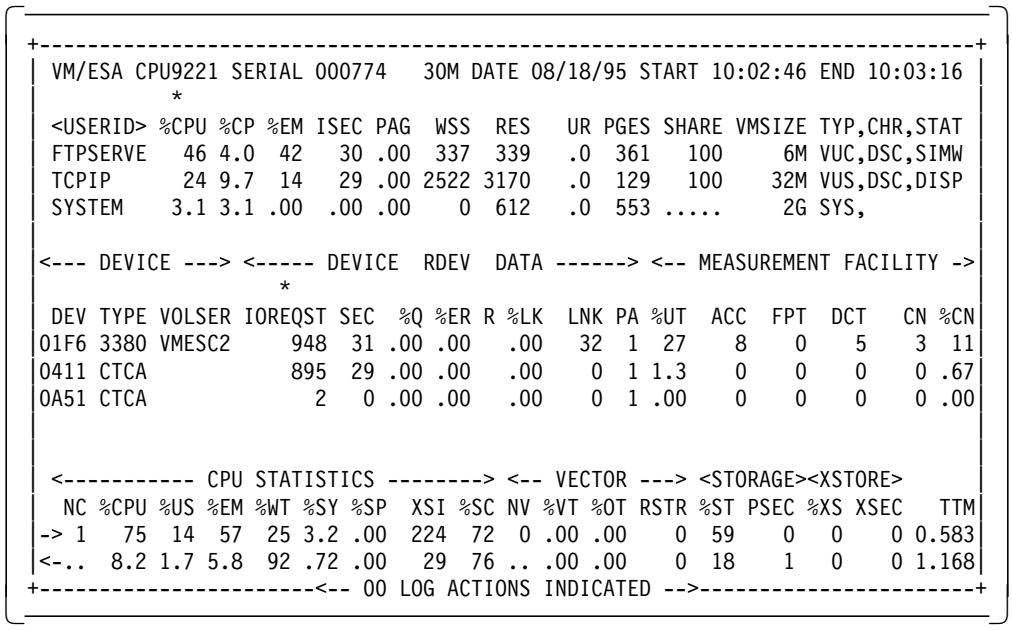


Figure 63. RTM Screen on FTP Put File from RS/6000 FTP User (BLKMUX)

7.3.4 Performance Tuning Guideline

This summarizes what we learned about throughput as a result of the Austin tests.

7.3.4.1 Test Summary

Before we talk about some tuning points for the FTP, let's take a look at the FTP test summary table as follows.

Table 7. Test Result from VM Host and RS/6000 on TCP/IP File Transfer.

Channel Type	Initiating User	FTP Direction	Transfer Rate	Most Active Users	Busiest Devices
ESCON	VM	upload	270 Kbytes/s	VMFTP1, TCPIP	VMESC2, CTCA
		download	674 Kbytes/s	TCPIP, VMFTP1	CTCA, VMESC2
	AIX	upload	259 Kbytes/s	FTPSERVE, TCPIP	VMESC2, CTCA
		download	648 Kbyte/s	TCPIP, VMFTP1	CTCA, VMESC2
BLKMUK	VM	upload	257 Kbytes/s	VMFTP1, TCPIP	VMESC2, CTCA
		download	673 Kbytes/s	TCPIP, VMFTP1	CTCA, VMESC2
	AIX	upload	245 Kbytes/s	FTPSERVE, TCPIP	VMESC2, CTCA
		download	633 Kbyte/s	TCPIP, VMFTP1	CTCA, VMESC2

The table shows that the transfer rate for the ESCON and BLKMUX is very similar in spite of the different device characteristics. The FTP program is utilizing CPU cycles in its negotiation instead of just sending the data. Even if we use faster media to transfer data, the result will, most likely, be the same.

You can compare this with the use of CLIO under MVS, which gets relatively near channel bandwidth speeds because it is mainly using channel protocol after the initial TCP/IP handshake. Please see Chapter 6, " IBM Client Input Output/Sockets (CLIO)" on page 65.

Also, the uploading of the file is very slow because of the test environment as there is only one channel path to the 3380 DASDs to which the FTP program writes the file. The slow I/O speed makes the transfer rate slow down.

Also, the most active users depends on the transfer type. For example, when a VM user initiates the upload of a file, the user and TCPIP machine are most active, but the FTPSERVE machine and TCPIP machine is most active when the AIX user initiates uploading.

Lastly, the busiest devices are also varied according to the transfer type.

7.3.4.2 FTP Performance Tuning Guide for VM System

There can be various ways of improving the system performance. Let's focus on the key points for tuning from our tests.

1. Using socket programs

   A lot of tests from labs show that FTP using socket programs can increase the transfer rate significantly. Please refer to the related TCP/IP Socket Programming Guide manuals and redbooks.

2. Improving Disk I/O Rate

   The I/O bottleneck on the VM user's 3380 DASD resulted in the poor file transfer performance. The following ways can be used to improve the disk I/O rate.

   - Use faster devices, such as the 3990 and 3390
   - Create more paths to DASDs (3990 supports 4-path dynamic channel select and reconnect function)
   - Use memory for disk caching
   - Use cached controllers
   - Move minidisks across multiple paths

3. Improving transfer rate and throughput

   We can improve the throughput by adjusting the data buffers in the PROFILE TCPIP file on the TCPIP 191 disk with the following methods.

   - Using DATABUFFERPOOLSIZE

     We recommend that you use 32768 (32 K) bytes. The default number of regular data buffers is 160. Running out of data buffers causes the abnormal ending of active connections.

   - Using the SMALLDATABUFFERPOOLSIZE

     Small data buffers hold 2048 bytes of data, in contrast to regular data buffers, which hold 8192 bytes or more. They are used by each Telnet

server's TCP connections. If there is no small data buffer pool defined, Telnet will use regular data buffers and waste the storage.

- Using the LARGEENVELOPEPOOLSIZE

  A large envelope is use only if a packet does not fit in a small envelope. Running out of these buffers causes TCP/IP to drop outgoing and incoming packets. The resulting retransmission of lost packets lowers performance.

- Working with Packet Size

  The Maximum Transmission Unit (MTU) can be no larger than the larger envelope size. You can change the MTU using the GATEWAY statement entry pertaining to the network connection in the PROFILE TCPIP file.

  If you are using routed, you will need to change the MTU by using the BSDROUTINGPARMS statement instead of the GATEWAY statement.

4. Improving CPU Utilization

   - Larger buffer sizes and larger MTU size can improve CPU utilization
   - Locked or reserved pages for TCP/IP
   - Favored status to TCP/IP servers

     For example, if an AIX user initiates the upload of a file to VM host, the FTPSERVE machine becomes very busy. If you want make the machine work faster, you can provide more CPU cycle to the machine from MAINT or OPERATOR (user) by issuing following commands:

     ```
     'set quickdsp on FTPSERVE'
     'set share FTPSERVE absolute 20%'
     ```

7.4 SNA Connection

The RS/6000 connected directly by channel to mainframe is considered as a PU type 2.1 peripheral node. A 3270 emulation or dependent LU 6.2 communication is an example of a peripheral node. It is defined to Host VTAM like a 3174 terminal control unit. Please also refer to Chapter 5, "SNA Channel Connectivity Using AIX SNA Server/6000" on page 49 throughout this chapter as the latter refers to the AIX setup with MVS which is very similar to VM.

Following is the requirement of both software and hardware for direct channel attachment implementation by VM/VTAM.

- VM/SP Release 5 (5664-173) or VM/XA Release 2 (5664-308)
- VM/ESA V1.1.1 or higher (our environment is V1.2.2)
- VM/VTAM V3.3 or higher (our environment is V3.4.1)
- S/370, S/390 or ES/9000 (in our case ES/9221-150 4 MIPS)
- One tape drive
- 3270 terminal
- One Block Multiplexer channel or one ESCON channel

The requirements of both software and hardware for direct channel attachment implementation by AIX SNA Server/6000 is shown in Chapter 5, "SNA Channel Connectivity Using AIX SNA Server/6000" on page 49 and 5.1.1, "SNA Channel for AIX Packaging of the Files" on page 49.

The following figure shows the overall VTAM configuration environment for our test.

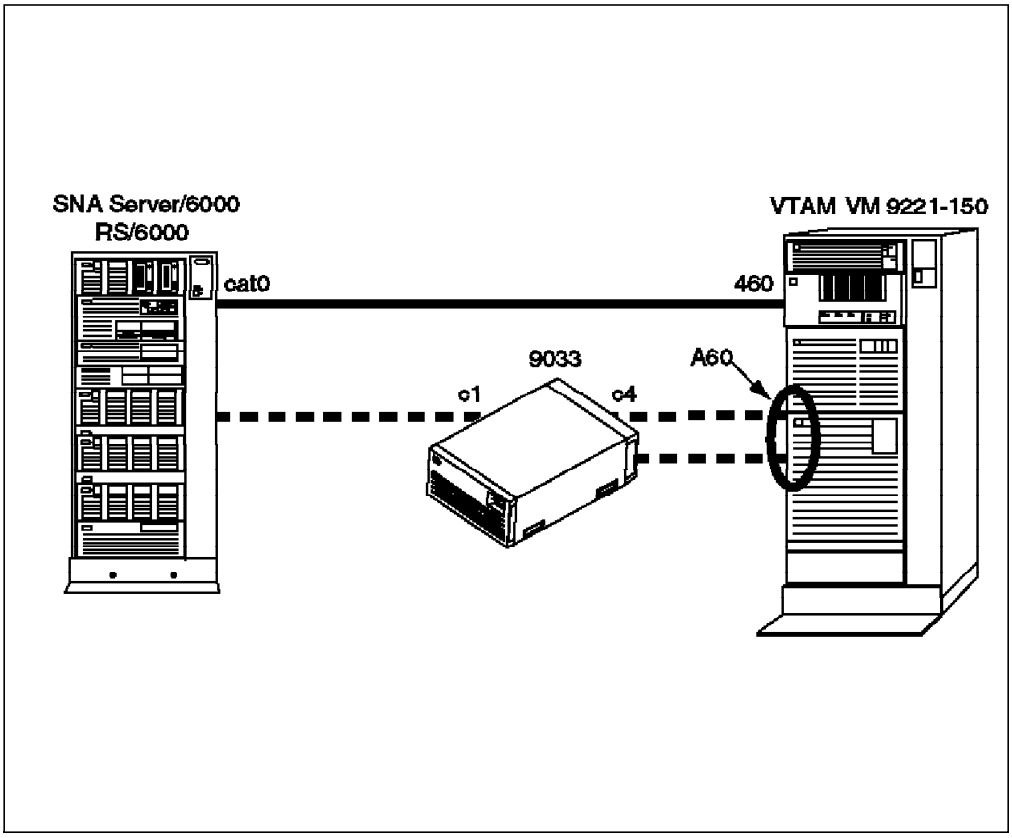


Figure 64. SNA Channel Attach Test Configuration.

A60 is the channel address of ESCON channel, and 462 is that of BLKMUX channel.

7.5 SNA Configuration Overview

The VM SNA subsystem is composed of Group Control System (GCS) group. GCS is a component of VM, like CMS, and provides the operating system environment for running SNA applications such as VTAM. There are GCS systems such as:

- VTAM (Virtual Telecommunication Access Method) - Controls data flow between SNA network devices and programs running in other group machines.
- VSCS (VTAM SNA Console Support) - This is the VTAM component that let's SNA-connected terminals function as Virtual Machine consoles.
- RSCS (Remote Spooling Communications Subsystem) - Designed as a GCS application, it runs in a group Virtual Machine and relies on VTAM to transfer information through the SNA network. RSCS can also spool files and transmit messages through non-SNA links.
- CCS (SNA/Console Communication Services) - SNA/CCS provides full VM/ESA console capabilities to operators (users) of SNA terminals. It's not a GCS component, but a VM CP component that closely works with VTAM subsystems.

We are not going to look further at all the GCS subsystems because it would be somewhat out of the topic. We are going to focus on the VTAM machine which communicates directly with SNA Server/6000 on RS/6000. For more information for setting other GCS group subsystems, please refer to the *VM/ESA Installation Manual*, the *VM/ESA Planning Guide*, the *RSCS V3.2 Planning and Administration Guide*, and the *VM/ESA Group Control System Reference* manuals. The quickest way is to discuss it with your VM system programmer.

Let's take a brief look at the VTAM server set up configuration files. You may need these definitions all through this chapter for reference.

7.5.1 VM/VTAM Configuration

7.5.1.1 Typical Sequence of a VM/VTAM Start Up

In VM, the following command in Profile GCS, which the VTAM machine has in its 191 disk, starts the VTAM network. When the VTAM is autologged by AUTOLOG1 machine, this file is executed automatically, and it brings up the SNA network.

```
'VTAM  START  LIST=11'
```

The following figure shows the overall VTAM start up procedure.

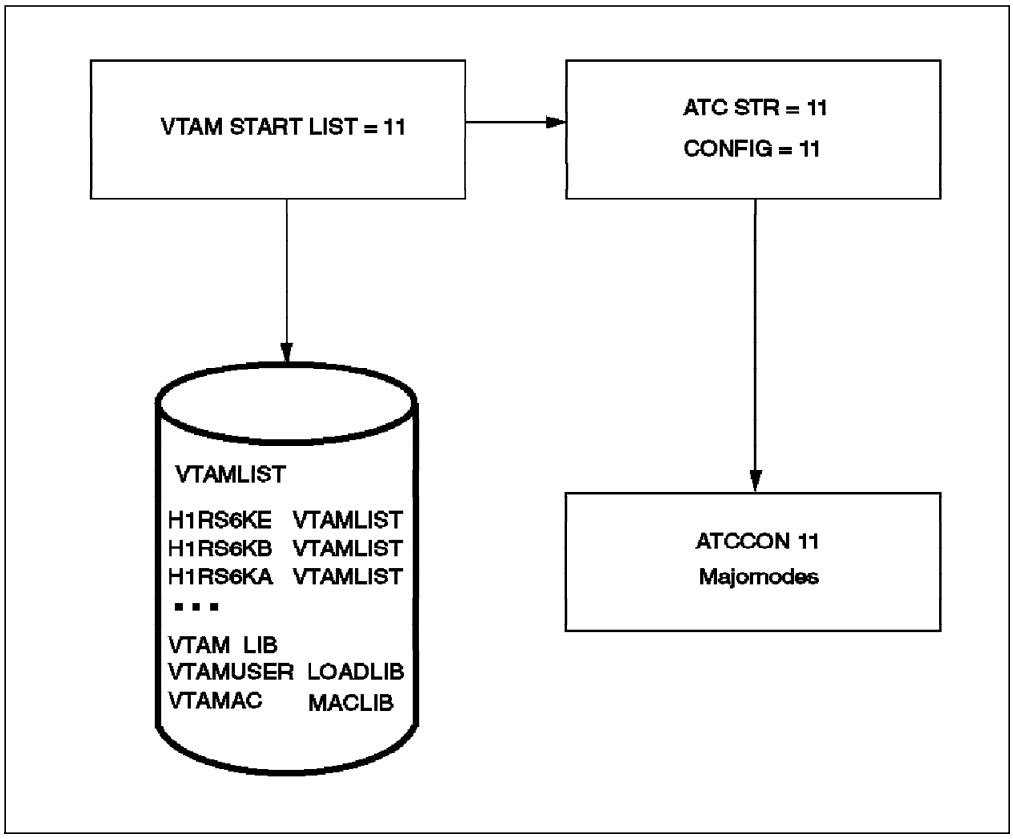


Figure 65. VM Start Up Procedure.

Here is a brief explanation of the VTAM startup procedure.

1. When the VTAM machine logs on, the PROFILE GCS file is automatically started.

2. The VTAM START LIST=11 command within the file starts the ATCSTR11 VTAMLST file which contains the VTAM start options.
3. The CONFIG=11 parameter points to the configuration list, ATCCON11, that VTAM will use to start its resources.
4. Finally, VTAM gets the information from the configuration list, major nodes definitions, and loadlibs and holds overall control of the network.

7.5.1.2 VTAM Virtual Machine Directory

The directory for VTAM machine looks like this:

```
USER VTAM password 24M 32M ABG                                  07201058
 OPTION MAXCONN 400 QUICKD DIAG98                               07201058
 SHARE REL 800                                                         58
 IUCV *CCS P M 10                                                      58
 IUCV ANY P M 0                                                        58
 ACCOUNT 8000                                                          58
 IPL GCS PARM AUTOLOG                                                  58
 MACHINE XA                                                            58
 NAMESAVE GCS VTAM                                                     58
 CONSOLE 01F 3215 W OPERATOR                                           58
 SPOOL 00C 2540 READER A                                               58
 SPOOL 00D 2540 PUNCH  A                                               58
 SPOOL 00E 1403 A                                                      58
 LINK MAINT 190 190 RR                                                 58
 LINK VTAMAINT 114 191 RR                                              58
 LINK VTAMAINT 298 192 RR                                              58
 LINK VTAMAINT 49A 29A RR                                              58
```

Figure 66. VTAM User Directory

- MAXCONN defines the maximum number of IUCV connections allowed for VSCS connections to *CCS (1 IUCV connection = 1 terminal session). IUCV(Inter User Communication Vehicle) is the CP-added facility to handle the interface between the VSCS code and the Virtual Machines in VM, such as CMS and RSCS.
- The QUICKD option enables VTAM to get into the dispatch list for the CP immediately when it has work to do.
- The DIAG98 option enables VTAM to do its own I/O and all paging and spooling.
- IUCV *CCS allows communication between VSCS and the CP system services. VTAM restricts priority messages of queued users to 10 at a time in this case.
- IUCV ANY allows IUCV sessions with any other Virtual Machines.
- NAMESAVE GCS VTAM authorizes the VTAM machine to access the GCS and VTAM Named Saved Systems (NSS) which have been designated as restricted. The NSSs are the saved segments which are loaded to main memory when the host system IPLs, and you must make the NSS for GCS and VTAM in order to bring up the SNA system successfully.
- The VTAMAINT machine is the installing user ID for VTAM. has all the configuration files on the 298 disk and the VTAM Libraries on the 49A disk.

7.5.1.3 Profile GCS File on the VTAM 191 Disk

Below is the Profile GCS file which is executed when the VTAM logs on. You can think of it as the Profile Exec for VTAM.

```
parse source . . exec_name
arg list_value . '(' options
if list_value = '' then
  LIST_VALUE='11'
'CP ENABLE SNA'                 /* ENABLE SNA COMMUNICATIONS     */
'ACC 29A F/F'                   /* VTAM run disk                 */
'GLOBAL LOADLIB VTAMUSER VTAM VSCS'
'LOADCMD VTAM ISTINVOO'
'LOADCMD VSCS DTISLCMD'
'VTAM START LIST=11'
    - - - -
'VSCS START'/* INITIALIZE VSCS              */
    - - - -
exit 0
```

Figure 67. Profile GCS

It starts the ATCSTR11 VTAMLST file and, if everything goes well, starts VSCS for user communication.

7.5.1.4 ATCSTR11 VTAMLST file on VTAMAINT's 298 disk

```
CONFIG=11,                                                      X
HOSTSA=01,                                                      X
NETID=USIBM01,                                                  X
SSCPID=4671,                                                    X
SSCPNAME=TH1VTAM,                                               X
HOSTPU=SA1PUS,                                                  X
DYNLU=NO,                                                       X
BSBUF=(144,,,,,),                                               X
XDBUF=(110,,,,,),                                               X
   - - -                                                          0551520
NOPROMPT
```

Figure 68. ATCSTR11 VTAMLST

This file has the startup list and options, like buffers and defines, items such as NETID, and the name of VTAM.

7.5.1.5 ATCCON11 VTAMLST File on VTAMAINT's 298 Disk

VTAM looks for a file called ATCCON11 VTAMLIST.

```
*     THIS MEMBER IS THE CONFIGURATION LIST FOR AUSESC
         H1NVA,                                                    X
         H1PATH,                                                   X
         H1RSCSA,                                                  X
         H1RS6KB,                                                  X
         H1RS6KE,                                                  X
         H1VSCSA,                                                  X
         H1AVSA,                                                   X
         H1V3274,                                                  X
      HATCDRM
```

Figure 69. ATCCON11 VTAMLST

H1RS6KB is the name of local major node for our RS/6000 workstation connected to BLKMUX channel; H1RS6KE is for ESCON channel.

7.5.1.6 USSTAB ASSEMBLE File on VTAMAINT's 298 Disk

The USS (Unsupported System Services) table allows users to simplify logon procedure. A user can just type VM or L on the VTAM logon screen when using the USS table as follows.

```
USSHATS  USSTAB TABLE=STDTRANS,FORMAT=V3R2                          USS00010
*********************************************************************  USS00090
*  STANDARD ENTRY TO SUPPORT FORMATTED LOGONS                         USS00100
*********************************************************************  USS00110
LOGON    USSCMD   CMD=LOGON,FORMAT=PL1                              USS00120
         USSPARM  PARM=APPLID                                       USS00130
         USSPARM  PARM=LOGMODE                                      USS00140
         USSPARM  PARM=DATA                                         USS00150
*********************************************************************  USS00160
*  STANDARD ENTRY TO SUPPORT LOGOFF                                   USS00170
*********************************************************************  USS00180
LOGOFF   USSCMD   CMD=LOGOFF,FORMAT=PL1                             USS00190
         USSPARM  PARM=APPLID                                       USS00200
         USSPARM  PARM=TYPE,DEFAULT=UNCOND                          USS00210
         USSPARM  PARM=HOLD,DEFAULT=YES                             USS00220
*********************************************************************  USS00670
*   SPECIAL ENTRY TO SUPPORT SHORTHAND FORMATTED LOGONS               USS00680
*********************************************************************  USS00690
VM       USSCMD  CMD=VM,REP=LOGON,FORMAT=BAL USS MSG 12 TEST        USS00700
L        USSCMD  CMD=L,REP=LOGON,FORMAT=BAL                         USS00710
         USSPARM REP=APPLID,PARM=P1                    1ST POSTIONAL  USS00720
         USSPARM REP=APPLID,PARM=A                                  USS00730
         USSPARM REP=APPLID,PARM=APPLID                             USS00740
         USSPARM REP=DATA,PARM=P2                                   USS00750
         USSPARM REP=DATA,PARM=DATA                                 USS00760
         USSPARM REP=DATA,PARM=D                                    USS00770
         USSPARM REP=LOGMODE,PARM=P3                                USS00780
         USSPARM REP=LOGMODE,PARM=LOGMODE                           USS00790
         USSPARM REP=LOGMODE,PARM=MODE                              USS00800
END      USSEND                                                     USS03950
         END                                                        USS03960
further for the detail connection information that follows.
```

Figure 70. USSHATS Assemble

On the other hand, when using the formatted logon procedure, a user must type all the logon parameters as follows.

```
Logon APPLID(name) DATA(user data) LOGMODE(name)
```

7.5.1.7 LOGMODE Assemble File on VTAMAINT's 298 Disk

The SNA-attached RS/6000 terminals must use the correct logmode table to get the 3270 session via the SNA protocol. We used the following logmode in the relevant LOGMODE Assemble file.

```
D3270M0Q MODEENT LOGMODE=D3270M0Q,                                    x
               TYPE=X'01',                                            x
               FMPROF=X'03',                                          x
               TSPROF=X'03',                                          x
               PROPROT=X'B1',                                         x
               SECPROT=X'90',                                         x
               COMPROT=X'3080',                                       x
               SSNDPAC=X'00',                                         x
               SRCVPAC=X'00',                                         x
               RUSIZES=X'87C7',                                       x
               PSNDPAC=X'00',                                         x
               PSERVIC=X'028000000000000000000300'
```

Figure 71. D3270M0Q Logmode for RS/6000 SNA Connection

7.5.1.8 Tips for Putting the USS Code and Logmode into the VTAM Loadlib

The USS Assemble file and the MTAWD Assemble file must be assembled and linkedited into the loadlib that VTAM machine uses. We are using the VTAMUSER LOADLIB. Logon to the VTAMAINT machine, and issue the following commands:

```
'acc 298 a'
'acc 29a b'
'global maclib vtamac vtambld
'hasm usshats'
'hasm mtawd'
'vmflked vtamuser'
'acc 29a b'
```

The VTAMUSER LKEDCTRL file will be used for in this linkedit procedure The VTAMUSER LOADLIB will be built, and it looks like this:

```
%LEPARMS REUS
 INCLUDE MTAWD
 NAME MTAWD(R)
 INCLUDE USSHATS
 NAME USSHATS(R)
```

Figure 72. VTAMUSER LKEDCTRL File

7.6 ESCON Connection Test by SNA

This section describes the steps necessary to establish the VM to AIX ESCON connection for SNA.

7.6.1 VM Configuration for ESCON

These are the steps required to define ESCON for SNA on a VM system.

7.6.1.1 IOCP Definition

Unquestionably, you need the IOCP definition for the RS/6000 on mainframe in order to make the channel connection between a mainframe and a workstation. Below is the IOCP definition for the channel and the device (workstation) for ESCON channel.

```
CHPID    PATH=((25)),TYPE=CNC,SWITCH=01
CNTLUNIT CUNUMBR=0260,PATH=(25),UNITADD=((60,032)),UNIT=3174,LINK=(CB)
IODEVICE ADDRESS=(A60,032),CUNUMBR=(0260),UNIT=3791L,STADET=Y
```

7.6.1.2 Real I/O Device Definition

.We used VM/ESA Release 2, in which case you don′t need to define the device to VM CP (Control Program). If you are using VM/ESA R1 or VM/XA, you must define the device in the HCPRIO Assemble file as follows.

```
RDEVICE DEVNO=(A60,32),DEVTYPE=3705,ADAPTER=TYPE4,MODEL=G6
```

If you are defining the device in HCPRIO or DMKRIO (VM/SP), you must generate a new CP nucleus and reIPL the VM system.

7.6.1.3 Major Node Definition for ESCON Channel Device

The following VTAMLST is the VTAM resource definition for RS/6000 as a local SNA major node connected to ESCON channel.

```
H1RS6KE  VBUILD TYPE=LOCAL
VM6KE    PU CUADDR=A60,                                          X
               XID=YES,                                          X
               MAXBFRU=5,                                        X
               VPACING=7,                                        X
               PACING=7,                                         X
               PUTYPE=2,                                         X
               DELAY=0.00,                                       X
               ISTATUS=ACTIVE,                                   X
               USSTAB=USSHATS,                                   X
               MODETAB=MTAWD,                                    X
               DLOGMOD=D3270MOQ
TH1ECP   LU    LOCADDR=0
TH1E02   LU    LOCADDR=2
TH1E03   LU    LOCADDR=3
TH1E04   LU    LOCADDR=4
TH1E05   LU    LOCADDR=5
```

Figure 73. H1RS6KE VTAMLST (BLKMUX)

RS6KE is the PU name of this local major node. TH1ECP, TH1E02 and so on, are LU names for the terminals.

7.6.1.4 VTAM Operation for RS/6000 SNA Major Node Start (ESCON)

After the configuration definitions for AIX SNA channel connection are finished, you can issue the following command from VTAM. Remember, this can be working only after the AIX Link Station is in starting state, after the *sna -start link-station* command is issued from the AIX administrator.

```
vary on a60
attach a60 vtam
vtam vary net,activate,id=h1rs6ke
```

After the successful vary on for h1rs6ke, you may issue the following display command to see the status for the nodes.

```
Ready;
vtam d net,id=h1rs6ke,e
Ready;
IST097I DISPLAY ACCEPTED
IST075I NAME = H1RS6KE, TYPE = LCL SNA MAJ NODE
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST084I NETWORK NODES:
IST089I VM6KE    TYPE = PHYSICAL UNIT    , ACTIVE     ,CUA=A60
IST089I TH1E02   TYPE = LOGICAL UNIT     , ACTIVE
IST089I TH1E03   TYPE = LOGICAL UNIT     , ACTIVE
IST089I TH1E04   TYPE = LOGICAL UNIT     , ACTIVE
IST089I TH1E05   TYPE = LOGICAL UNIT     , ACTIVE
IST314I END
```

Figure 74. VTAM Display for RS/6000 SNA Major Node

Now you are ready to go for the SNA connection for ESCON between the VM host and the RS/6000 by channel.

7.6.2 AIX Configuration for SNA ESCON

Chapter 5, “SNA Channel Connectivity Using AIX SNA Server/6000” on page 49 explains the detailed set up for SNA over ESCON on the RS/6000. Please refer to Figure 10 on page 28 and to Figure 11 on page 29 for the subchannel addressing, and make sure that the TH1ECP is the definition on the RS/6000 and that VM6KE is the definition on the mainframe. The actual SNA emulation was through HCON, and specifically, the subchannel set name was called HCON4. Figure 9 on page 27 shows more details, and the previous sections explained how we set up and tested the connectivity.

7.7 BLKMUX Connection Test by SNA

This section describes the steps necessary to establish the VM to AIX BLKMUX connection for SNA.

7.7.1 VM Configuration for BLKMUX

These are the steps required to define BLKMUX for SNA on a VM system.

7.7.1.1 IOCP Definition

Below is the IOCP definition for the channel and the device (workstation) for the ESCON channel.

```
CHPID    PATH=((20)),TYPE=BL
CNTLUNIT CUNUMBR=0204,PATH=(20),UNITADD=((60,32)),UNIT=3174,SHARED=N,
             PROTOCL=S4
IODEVICE ADDRESS=(460,32),CUNUMBR=(0204),UNIT=3791L,STADET=N,TIMEOUT=Y
```

7.7.1.2 Real I/O Device Definition

.We used VM/ESA Release 2, in which case you don't need to define the device to the VM CP (Control Program). If you are using VM/ESA R1 or VM/XA, you must define the device in the HCPRIO Assemble file as follows.

```
RDEVICE DEVNO=(460,32),DEVTYPE=3705,ADAPTER=TYPE4,MODEL=G6
```

If you are defining the device in HCPRIO or DMKRIO (VM/SP), you must generate new a CP nucleus and reIPL the VM system.

7.7.1.3 Major Node Definition for the ESCON Channel Device

The following VTAMLST is the VTAM resource definition for the RS/6000 as a local SNA major node connected to BLKMUX channel.

```
H1RS6KB  VBUILD TYPE=LOCAL
VM6KB    PU CUADDR=460,                                          X
               XID=YES,                                          X
               MAXBFRU=5,                                        X
               VPACING=7,                                        X
               PACING=7,                                         X
               DELAY=0.00,                                       X
               USSTAB=USSHATS,                                   X
               MODETAB=MTAWD,                                    X
               DLOGMOD=D3270M0Q
TH1BCP   LU    LOCADDR=0
TH1B02   LU    LOCADDR=2
TH1B03   LU    LOCADDR=3
TH1B04   LU    LOCADDR=4
TH1B05   LU    LOCADDR=5
```

Figure 75. H1RS6KB VTAMLST (BLKMUX)

RS6KB is the PU name of this local major node, and TH1BCP, TH1B02 and so on, are the LU names for the terminals.

7.7.1.4 VTAM Operation for RS/6000 SNA Major Node Start (BLKMUX)

You can refer to the previous operation procedure for ESCON, and when you issue the VTAM display command, it will show you the following screen.

```
Ready;
vtam d net,id=h1rs6kb,e
Ready;
IST097I DISPLAY ACCEPTED
IST075I NAME = H1RS6KE, TYPE = LCL SNA MAJ NODE
IST486I STATUS= ACTIV, DESIRED STATE= ACTIV
IST084I NETWORK NODES:
IST089I VM6KB    TYPE = PHYSICAL UNIT    , ACTIVE     ,CUA=460
IST089I TH1BO2   TYPE = LOGICAL UNIT     , ACTIVE
IST089I TH1BO3   TYPE = LOGICAL UNIT     , ACTIVE
IST089I TH1BO4   TYPE = LOGICAL UNIT     , ACTIVE
IST089I TH1BO5   TYPE = LOGICAL UNIT     , ACTIVE
IST314I END
```

Figure 76. VTAM Display for RS/6000 SNA Major Mode (BLKMUX)

The PU VM6KB is active, and you can enjoy the conversation between the VM host and the AIX system by the SNA protocol.

7.7.2 AIX Configuration for SNA BLKMUX

Chapter 5, “SNA Channel Connectivity Using AIX SNA Server/6000” on page 49 explains the detailed set up for SNA over BLKMUX on the RS/6000. Please refer to Figure 10 on page 28 and Figure 11 on page 29 for the subchannel addressing, and make sure that TH1BCP is the definition on the RS/6000 and that VM6KB is the definition on the mainframe.

Chapter 8. Throughput Figures and Tuning

This manual describes our experiences connecting AIX to mainframes via S/370 channels. It is not intended to be a performance or tuning guide.

The figures below are for the Austin systems we worked on during this residency, including a collation of previous work done by testing and development at Austin. The details are therefore very specific and are to be used as a guide **ONLY** for throughput that can be achieved with the specific hardware and software that we used. This is very important when discussing performance because the data pipe between mainframes and RS/6000′s can be very complex.

8.1 Throughput Bottlenecks

We are aware that it is dangerous to provide performance figures, and we trust the reader′s will carefully compare their systems with ours before deciding on the data throughput that is possible in their systems. Also, the technology changes very fast in terms of connectivity throughput, and therefore, newer architectures may eradicate any data bottlenecks in the systems.

On the whole, we had slow DASD (3380) and slow RS/6000 disk (2 GB internal), which were big factors in the limitations of our system. The channel bandwidth of up to 10 MB/S was therefore not achieved, except in memory-to-memory transfer results that we obtained through CLIO. This, however, does show that ESCON can be driven at nearly full bandwidth provided that certain factors are taken into account. These factors include use use of fast disks, such as the the 9333, the new SSA disk technology on the RS/6000 and DASD stiping on the MVS side.

The performance possible in a benchmarking environment (see 6.4.3.8, “Maximum Throughput” on page 90) is therefore a lot higher than we achieved in our environment.

8.2 File Transfer Protocol (FTP) Performance Tuning Guide

There can be various ways of improving the system performance. Let′s focus on the key points for tuning from our tests.

1. Using socket programs

   A lot of tests from laboratories show that FTP, using socket programs, can increase the transfer rate significantly. Please refer to the related TCP/IP Socket Programming Guide manuals and redbooks.

2. Improving disk I/O rate

   The I/O bottleneck on the mainframe user′s 3380 DASD resulted in the poor file transfer performance. There are a number of ways to improve the disk I/O rate.

   - Use faster devices, such as the 3990.
   - Create more paths to DASD. The 3990 supports 4-path dynamic channel select and reconnect functions.
   - Use memory for disk caching.

- Use cached controllers.
- Move minidisks across multiple paths.

3. Improving transfer rate and throughput

   We can improve the throughput by adjusting the data buffers in the PROFILE.TCPIP file.

   - Using **DATABUFFERPOOLSIZE**

     We recommend that you use 32768 (32K) bytes. The default number of regular data buffers is 160. Running out of data buffers causes the abnormal ending of active connections.

   - Using the **SMALLDATABUFFERPOOLSIZE**

     Small data buffers hold 2048 bytes of data, in contrast to regular data buffers,which hold 8192 bytes or more. They are used by each Telnet server TCP connection. If there is no small data buffer pool defined, Telnet will use regular data buffers and waste the storage.

   - Using the **LARGEENVELOPEPOOLSIZE**

     A large envelope is used only if a packet does not fit in a small envelope. Running out of these buffers causes TCP/IP to drop outgoing an incoming packets. The resulting retransmission of lost packets lowers performance.

   - Working with Packet Size

     The MTU can be no larger than the largest envelope size. You can change the MTU by using the GATEWAY statement entry pertaining to the network connection in the Profile Tcpip file.

     If you are using routed, you will need to change the by MTU using the BSDROUTINGPARMS statement instead of the GATEWAY statement.

4. Improving CPU Utilization

   - Larger buffer and MTU sizes can improve CPU utilization
   - Locked or reserved pages for TCP/IP
   - Favored status to TCP/IP servers for VM

     For example, if an AIX user initiates the upload of a file to a VM host, the FTPSERVE machine becomes very busy. If you want make the machine work faster, you can provide more CPU cycle to the machine from MAINT or OPERATOR (user) by issuing the following commands:

     ```
     'set quickdsp on FTPSERVE'
     'set share FTPSERVE absolute 20%'
     ```

8.3 Performance Measurements

The RS/6000 performance characteristics were taken with `vmstat` and `iostat` commands during the file transfer process. The ideal buffer sizes were obtained after a lot of experimentation, and the MTU size of 4096 was the best. The ADSM results are one year old and are left as a comparison. The CLIO results are for a memory-to-memory transfer.

Table 8. Block Multiplexer Channel Adapter

Utility	Performance	9121 % Utilization	RISC/System 6000 CPU %
MVS reads: connection over BLKMUX:			
CLIO (1)	4.2 MB	9	51
CLIO (2)	8.4 MB	10	71
CLIO (3)			
CLIO (4)			
FTP/BIN	1.0 MB	60	33
FTP/ASCII	1.1 MB	67	43
ADSM ARCH	0.9 MB	66	25
ADSM BUP	0.9 MB	68	28

Table 9. Block Multiplexer Channel Adapter

Utility	Performance	9121 % Utilization	RISC/System 6000 CPU %
MVS writes: connection over BLKMUX:			
CLIO (1)	2.1 MB	10	38
CLIO (2)	4.1 MB	10	43
CLIO (3)			
CLIO (4)			
FTP/BIN	1.1 MB	51	45
FTP/ASCII	1.2 MB	53	43
ADSM RET	0.8 MB	80	32
ADSM RES	0.9 MB	68	28

Table 10. Block Multiplexer Channel Adapter

Utility	Performance	9221 % Utilization	RISC/System 6000 CPU %
VM reads: connection over BLKMUX:			
CLIO (1)	*	*	*
CLIO (2)	*	*	*
CLIO (3)	*	*	*
CLIO (4)	*	*	*
FTP/BIN	0.8 MB	71	29
FTP/ASCII	0.7 MB	72	38
WDSF ARCH	0.7 MB	80	28
WDSF BUP	0.7 MB	80	26

Table 11 (Page 1 of 2). Block Multiplexer Channel Adapter

Utility	Performance	9221 % Utilization	RISC/System 6000 CPU %
VM writes: connection over BLKMUX:			
CLIO (1)	*	*	*
CLIO (2)	*	*	*

Table 11 (Page 2 of 2). Block Multiplexer Channel Adapter			
Utility	**Performance**	**9221 % Utilization**	**RISC/System 6000 CPU %**
CLIO (3)	*	*	*
CLIO (4)	*	*	*
FTP/BIN	1.0 MB	68	24
FTP/ASCII	0.7 MB	56	23
WDSF RET	0.7 MB	82	22
WDSF RES	0.7 MB	79	19

Table 12. ESCON Control Unit Adapter (FC 2756)			
Utility	**Performance**	**9121 % Utilization**	**RISC/System 6000 CPU %**
MVS reads: connection over ESCON:			
CLIO (1)	8.5 MB	11	92
CLIO (2)	12.5 MB (two channels)	10	100
CLIO (3)	14.3 MB (three channels)	10	100
CLIO (4)	15.2 MB	10	100
FTP/BIN	1.9 MB	71	50
FTP/ASCII	1.5 MB	74	76
ADSM ARCH	1.2 MB	66	25
ADSM BUP	1.2 MB	70	42

Table 13. ESCON Control Unit Adapter (FC 2756)			
Utility	**Performance**	**9121 % Utilization**	**RISC/System 6000 CPU %**
MVS writes: connection over ESCON:			
CLIO (1)	8.4 MB	13	33
CLIO (2)	11.4 MB (two channels)	11	75
CLIO (3)	13.4 MB (three channels)	10	80
CLIO (4)			
FTP/BIN	2.1 MB	69	57
FTP/ASCII	2.2 MB	76	67
ADSM ARCH	0.7 MB	82	39
ADSM BUP	0.7 MB	75	39

Table 14 (Page 1 of 2). ESCON Control Unit Adapter (FC 2756)			
UTILITY	**Performance**	**9221 % Utilization**	**RISC/System 6000 CPU %**
VM reads: connection over ESCON:			
CLIO (1)	*	*	*
CLIO (2)	*	*	*
CLIO (4)	*	*	*
FTP/BIN	0.7 MB	74	43
FTP/ASCII	0.6 MB	76	50

Table 14 (Page 2 of 2). ESCON Control Unit Adapter (FC 2756)			
UTILITY	**Performance**	**9221 % Utilization**	**RISC/System 6000 CPU %**
WDSF ARCH	0.9 MB	85	36
WDSF BUP	0.9 MB	85	36

Table 15. ESCON Control Unit Adapter (FC 2756)			
Utility	**Performance**	**9221 % Utilization**	**RISC/System 6000 CPU %**
VM reads: connection over ESCON:			
CLIO (1)	*	*	*
CLIO (2)	*	*	*
CLIO (4)	*	*	*
FTP/BIN	0.7 MB	63	42
FTP/ASCII	0.7 MB	62	40
WDSF RES	0.5 MB	79	39
ADSM BUP	0.5 MB	77	38

Appendix A. TCPIP and CLIO Configuration Files

This appendix contains the configuration files for the systems used during the residency in Austin.

A.1 SYS1.PROCLIB.TCPIP

This is the TCPIP.PROCLIB file used in Austin.

```
****** ***************************** Top of Data ******************************
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>           your edit profile using the command RECOVERY ON.
 000001 //TCPIP PROC MODULE='TCPIP',PARMS='/'
 000002 //*********************************************************************
 000003 //*                                                                   *
 000004 //*         TCP/IP for MVS Version 3, Release 1, Level 0              *
 000005 //*                                                                   *
 000006 //*         5655-HAL (C) Copyright IBM Corp. 1989, 1994.              *
 000007 //*         All rights reserved.                                      *
 000008 //*         US Government Users Restricted Rights -                   *
 000009 //*         Use, duplication or disclosure restricted                 *
 000010 //*         by GSA ADP Schedule Contract with IBM Corp.               *
 000011 //*         Refer to Copyright Instructions                           *
 000012 //*         Form Number G120-2083.                                    *
 000013 //*                                                                   *
 000014 //*  Change log:                                                      *
 000015 //*     07/26/94  J Wenzlaff                                          *
 000016 //*               - copy from TCPIP.V3R1.SEZAINST(TCPIPROC)           *
 000017 //*               - change profile to tcpip.v3r1.aushat3.profile      *
 000018 //*               - change tcpdata to sys1.tcpparms(tcpdata)          *
 000019 //*     09/28/94  J Wenzlaff                                          *
 000020 //*               - change profile to sys1.tcpparms(aushat2)          *
 000021 //*                                                                   *
 000022 //*     10/20/94  J Wenzlaff                                          *
 000023 //*               - copy from hat2                                    *
 000024 //*               - change profile to sys1.tcpparms(aushat4)          *
 000025 //*               - change tcpip.v3r1 to tcpip.v3r1m0 for ga code     *
 000026 //*                                                                   *
 000027 //*     10/21/94  J Wenzlaff                                          *
 000028 //*               - copy from hat4                                    *
 000029 //*               - change profile to sys1.tcpparms(aushata)          *
 000030 //*                                                                   *
 000031 //*********************************************************************
 000032 //TCPIP EXEC PGM=MVPMAIN,
 000033 //             PARM='&MODULE,ERRFILE(SYSERR),HEAP(512),&PARMS',
 000034 //             REGION=9999K,TIME=1440
 000035 //*     The C runtime libraries should be in the system's link list
 000036 //*         or add them to the STEPLIB definition here.  If you add
 000037 //*         them to STEPLIB, they must be APF authorized.
 000038 //STEPLIB  DD DSN=TCPIP.V3R1M0.SEZATCP,DISP=SHR
 000039 //SYSMDUMP DD SYSOUT=*
 000040 //*********************************************************************
 000041 //*  The SYSPRINT, SYSERR, SYSERROR, and SYSDEBUG DDs can specify
 000042 //*  a dataset name or SYSOUT. All of these DDs support the use
 000043 //*  of alternate datasets, which are specified by replacing th e
 000044 //*  DD name prefix SYS with SY1, SY2, or SY3.
 000045 //*  Output will cycle from the primary to the alternate datasets
```

```
000046 //*  and around again as each data set fills.
000047 //*********************************************************************
000048 //*       SYSPRINT contains runtime diagnostics from TCPIP.
000049 //SYSPRINT DD SYSOUT=*
000050 //*       SYSERR contains runtime diagnostics from Pascal.
000051 //SYSERR   DD SYSOUT=*
000052 //*       SYSERROR contains error messages from TCPIP that occurred
000053 //*        while processing the PROFILE and OBEYFILEs.
000054 //SYSERROR DD SYSOUT=*
000055 //*       SYSDEBUG receives output that is generated when the TRACE
000056 //*        parameter is specified in the PROFILE dataset.
000057 //SYSDEBUG DD SYSOUT=*
000058 //*SY1DEBUG DD DSN=TCPIP.TRACE.ALT1,DISP=SHR
000059 //*SY2DEBUG DD DSN=TCPIP.TRACE.ALT2,DISP=SHR
000060 //*SY3DEBUG DD DSN=TCPIP.TRACE.ALT3,DISP=SHR
000061 //*
000062 //*       TNDBCSCN is the configuration dataset for TELNET DBCS
000063 //*       transform mode.
000064 //*TNDBCSCN DD DSN=TCPIP.V3R1M0.SEZAINST(TNDBCSCN),DISP=SHR
000065 //*       TNDBCSXL contains binary DBCS translation table codefiles
000066 //*       used by TELNET DBCS Transform mode.
000067 //*TNDBCSXL DD DSN=TCPIP.V3R1M0.SEZAXLD2,DISP=SHR
000068 //*       TNDBCSER receives debug output from TELNET DBCS Transform
000069 //*       mode, when TRACE TELNET is specified in the PROFILE data set.
000070 //*TNDBCSER DD SYSOUT=*
000071 //*
000072 //*      The dataset containing the configuration parameters fo r
000074 //*      statement.  If the PROFILE DD statement is not present, a
000075 //*      hierarchical name search and dynamic allocation will be
000076 //*      performed.  Please see "Understanding TCP/IP Data Set Names"
000077 //*      in the Customization and Administration Guide for more
000078 //*      information.
000079 //*      The dataset name on the PROFILE DD statement can be an y
000080 //*      sequential data set or a member of a partitioned data set (PDS)
000081 //*PROFILE  DD DISP=SHR,DSN=TCPIP.V3R1M0.SEZAINST(SAMPPROF)
000082 //PROFILE  DD DISP=SHR,DSN=SYS1.TCPPARMS(AUSHATA)
000083 //*
000084 //*      SYSTCPD explicitly identifies which data set is to be
000085 //*      used to obtain the parameters defined by TCPIP.DATA.
000086 //*      The SYSTCPD DD statement should be placed in the TSO logon
000087 //*      procedure or in the JCL of any client or server executed
000088 //*      as a background task. The data set can be any sequential
000089 //*      data set or a member of a partitioned dataset (PDS).
000090 //*
000091 //*      For more information please see "Understanding TCP/IP Data Set
000092 //*      Names" in the Customization and Administration Guide.
000093 //*SYSTCPD  DD DISP=SHR,DSN=TCPIP.V3R1M0.SEZAINST(TCPDATA)
000094 //SYSTCPD  DD DISP=SHR,DSN=SYS1.TCPPARMS(TCPDATA)
```

A.2 SYS1.TCPPARMS.AUSHATA

This is the PROFILE.TCPIP file used in Austin.

```
; CHANGE LOG
;   10/21/94 Jim Wenzlaff
;            - copy from tcpip.profile.tcpip for v3r1
;            -add DATASETPREFIX for TCPIP V3.
;  This is a sample configuration file for the TCPIP address space.
;  The POOL sizes listed are the default values and need not be
;  specified. The device configuration statements MUST be changed
;  to match your hardware and software configuration.
;  The BEGINVTAM section must be changed to match your VTAM
;  configuration. It is recommended that you establish a
;  TCP maintenance userid and all occurrences of TCPMAINT be
;  changed to that userid. For more information about this
;  file, see "Configuring the TCPIP Address Space" and
;  "Configuring the Telnet Server" in the Installation and
;  Maintenance Manual.
;
ACBPOOLSIZE                  1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                  200
DATABUFFERPOOLSIZE           300 32768
ENVELOPEPOOLSIZE             750
IPROUTEPOOLSIZE              500
LARGEENVELOPEPOOLSIZE        300 32768
RCBPOOLSIZE                  50
SCBPOOLSIZE                  256
SKCBPOOLSIZE                 256
SMALLDATABUFFERPOOLSIZE      750
TCBPOOLSIZE                  512
TINYDATABUFFERPOOLSIZE       500
UCBPOOLSIZE                  100
;
; NOTRACE SCREEN
; MORETRACE PCCA
; MORETRACE CLAW
; RETRACE UDP IPDOWN IPUP
; Inform the following users of serious errors
INFORM
    A795849
    RICH
;   OPERATOR TCPMAINT
ENDINFORM
; Obey the following users for restricted commands
OBEY
    sue rich A795849 routed
;   OPERATOR TCPMAINT SNMPD SNMPQE ROUTED
ENDOBEY
; Flush the arp tables every 5 minutes
ARPAGE 5
; The SYSCONTACT and SYSLOCATION statements are used for SNMP.
;
; SYSCONTACT is the contact person for this managed node and how to
; contact this person.  Used for VM agent MIB variable sysContact
SYSCONTACT
   MAIN OPERATOR (823-6300)
ENDSYSCONTACT
; SYSLOCATION is the physical location of this node.  Used for VM
; agent MIB variable sysLocation
SYSLOCATION
```

```
   DHAT SYSTEM ROOM BLDG 901
ENDSYSLOCATION
;
; You can specify DATASETPREFIX in the PROFILE.TCPIP and
; TCPIP.DATA datasets.  The character string specified as a
; parameter on DATASETPREFIX takes precedence over both the distributed
; or modified dataset prefix name as changed by the EZAPPRFX
; installation job.  If this statement is used in a profile or
; configuration dataset that is allocated to a client or a server, then
; that client or server dynamically allocates additional required datasets
; using the value specified for DATASETPREFIX as the dataset name
; prefix.  The DATASETPREFIX parameter can be up to 26 characters long,
; and the parameter must NOT end with a period.
;
; For more information please see "Understanding TCP/IP Data Set
; Names" in the Customization and Administration Guide.
;
DATASETPREFIX TCPIP.V3R1M0  ; uncomment for tcpip v3
;
; Set Telnet timeout to 10 minutes
INTERNALCLIENTPARMS TIMEMARK 600 ENDINTERNALCLIENTPARMS
;
AUTOLOG
 ;  FTPSERVE    ; FTP Server
    FTPSERVP    ; FTP Server
 ;  FTPSERVa    ; FTP Server
 ;  FTPSERVb    ; FTP Server
 ;  FTPSERVc    ; FTP Server
 ;  NAMESRV     ; Domain Name Server
    PORTMAP     ; Portmap server
 ;  ROUTED      ; RouteD Server
 ;  SMTP        ; SMTP Server
    lpserve     ; lp Server
 ;  SNMPD       ; SNMP Agent Server
 ;  SNMPQE      ; SNMP Client Address space
 ;  TCPIPX25    ; X25
    MVSNFS      ; Network File System Server
ENDAUTOLOG
PORT
 ; Values from RFC 1010, "Assigned numbers"
 ;  20 TCP FTPSERVE  NOAUTOLOG ; FTP Server
 ;  21 TCP FTPSERVE            ; FTP Server
    20 TCP FTPSERVP  NOAUTOLOG ; FTP Server
    21 TCP FTPSERVP            ; FTP Server
 ;  20 TCP FTPSERVa  NOAUTOLOG ; FTP Server
 ;  21 TCP FTPSERVa            ; FTP Server
 ;  20 TCP FTPSERVb  NOAUTOLOG ; FTP Server
 ;  21 TCP FTPSERVb            ; FTP Server
 ;  20 TCP FTPSERVc  NOAUTOLOG ; FTP Server
 ;  21 TCP FTPSERVc            ; FTP Server
    23 TCP INTCLIEN            ; TELNET Server
    25 TCP SMTP                ; SMTP Server
 ;  53 TCP NAMESRV             ; Domain Name Server
 ;  53 UDP NAMESRV             ; Domain Name Server
   111 TCP PORTMAP             ; Portmap Server
   111 UDP PORTMAP             ; Portmap Server
;  161 UDP SNMPD               ; SNMP Agent
;  162 UDP SNMPQE              ; SNMPQE Agent
   515 tcp lpserve             ; lp server
```

```
   520 UDP ROUTED              ; RouteD Server
  2049 UDP MVSNFS              ; NFS Server
; Hardware definitions:
;
; change ca1/cat1 for arco test; it was network 11
  DEVICE CA0 CLAW 3d0 PETEMVS PETERISC NONE 26 26 4096 4096 ;           t3
  DEVICE dhat1 CLAW 400 DHATMVS CATODHA NONE 26 26 4096 4096 ; ca1 on dh
  DEVICE RPERF CLAW 910 MVSESC DHATRISC NONE 26 26 4096 4096 ; dhatrisk es0
  DEVICE CA3 CLAW C50 MVSBLK1 MVSBLK NONE 26 26 4096 4096 ; dhatchan ca t0
  DEVICE aix41 CLAW a20 MV3 PINOLAE3   NONE 26 26 4096 4096 ; pinola es3
  DEVICE AIX411 CLAW BC4 DHATMVS1 AIX411 NONE 26 26 4096 4096 ; aix_41 es1
  DEVICE AIX412 CLAW BC6 DHATMVS2 AIX412 NONE 26 26 4096 4096 ; aix_41 es2
  DEVICE MES2 CLAW B40 MV2 PINOLAE2 NONE 26 26 4096 4096 ; ES2 ON pinola
  DEVICE PNL  CLAW B20 MV1 PINOLA NONE 26 26 4096 4096 ; ES1 ON pinola
  DEVICE LAB2 CLAW CC2 G9121G POLY NONE 40 40 4096 4096 ; POLYESTER ES2
  DEVICE AIX413 CLAW CE0 DHAT TEST NONE 26 26 4096 4096 ; AIX_41 ES3
  DEVICE CA5 CLAW E00 MVS PINOLAC0 NONE 26 26 4096 4096 ; pinola ca0
  DEVICE CA1 CLAW D02 MV1 PINOLAC1 NONE 26 26 4096 4096 ; pinola ca1
  DEVICE CA2 CLAW D04 MV2 PINOLAC2 NONE 26 26 4096 4096 ; pinola ca2
  DEVICE CA6 CLAW E06 MV3 PINOLAC3 NONE 26 26 4096 4096 ; pinola ca3

; new links
  LINK CAT0 IP 0 CA0
  LINK CAT1 IP 0 CA1
  LINK CAT2 IP 0 CA2
  LINK CAT3 IP 0 CA3
  LINK CAT5 IP 0 CA5
  LINK CAT6 IP 0 CA6
  LINK laba IP 0 lab2
  LINK PRF IP 0 RPERF
  LINK AIX IP 0 AIX41
  LINK AIX1 IP 0 AIX411
  LINK AIX2 IP 0 AIX412
  LINK AIX3 IP 0 AIX413
  LINK dhat IP 0 dhat1
  LINK es2 IP 0 mes2
  LINK PIN IP 0 PNL

HOME
; Local host's Internet addresses
    10.0.0.2      CAT0
    11.0.0.2      CAT1
    14.0.0.2      CAT2
    21.0.0.2      CAT3
    13.0.0.2      dhat
    15.0.0.2      AIX
    16.0.0.2      AIX1
    17.0.0.2      AIX2
    20.0.0.2      AIX3
    18.0.0.2      PRF
    23.0.0.2      laba
    25.0.0.2      es2
    35.0.0.2      CAT5
    36.0.0.2      CAT6
    37.0.0.2      PIN
GATEWAY
 ; Network  First hop   Driver  Packet size  Subnet mask   Subnet value
    10           =         CAT0  4096         0
    11           =         CAT1  4096         0
```

```
    14          =         CAT2  4096         0
    21          =         CAT3  4096         0
    15          =         AIX   4096         0
    16          =         AIX1  4096         0
    17          =         AIX2  4096         0
    20          =         AIX3  4096         0
    18          =         PRF   4096         0
    23          =         laba  4096         0
    25          =         es2   4096         0
    35          =         CAT5  4096         0
    36          =         CAT6  4096         0
    37          =         PIN   4096         0
;   9           19.0.0.1  lab   4096         0
    13          =         dhat  4096         0
;
; all unknown destinations are routed through 60.2.2.254
; DEFAULTNET    10.0.0.1     CAT0   4096          0.255.255.0 0
; DEFAULTNET    19.0.0.1     lab    4096          0.255.255.0 0
; ; RouteD Routing information (if you are using the ROUTED server)
; ; If you are using RouteD, uncomment all the lines below for
; ; 'BSDROUTINGPARMS', and comment out all the lines for the 'GATEWAY'
; ; statement.
;
; ;  link     maxmtu   metric   subnet mask     dest addr
; BSDROUTINGPARMS false
;    X25LA     1024       0     255.255.255.0   0
;    ETH1      1500       0     255.255.255.0   0
;    ETH2      1500       0     255.255.255.0   0
;    PCN1      2000       0     255.255.255.0   0
;    TR1       2000       0     255.255.255.0   0
;    TR2       2000       0     255.255.255.0   0
;    TR3       2000       0     255.255.255.0   0
;    HCH1      1018       0     255.255.255.0   0
;    X25NPL1 DEFAULTSIZE  0     255.255.255.0   0
;    TESTLINK  1500       0     255.255.0.0     129.34.12.6
;    YORKTOWN  1500       0     255.0.0.0       0
; ENDBSDROUTINGPARMS
;
  TRANSLATE
; Define the VTAM parameters required for the TELNET server
ASSORTEDPARMS
TCPIPSTATISTICS
ENDASSORTEDPARMS
BEGINVTAM
    ; Define logon mode tables to be the defaults shipped with the lates
    ; level of VTAM
3278-3-E NSX32702 ; 32 line screen - default of NSX32702 is 24 line sc
3279-3-E NSX32703 ; 32 line screen - default of NSX32702 is 24 line sc
3278-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line sc
3279-4-E NSX32704 ; 48 line screen - default of NSX32702 is 24 line sc
3278-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 colu
3279-5-E NSX32705 ; 132 column screen - default of NSX32702 is 80 colu
    ; Define the LUs to be used for general users
DEFAULTLUS
    THOTCP01 THOTCP02 THOTCP03 THOTCP04 THOTCP05
    THOTCP06 THOTCP07 THOTCP08 THOTCP09 THOTCP10
    THOTCP11 THOTCP12 THOTCP13 THOTCP14 THOTCP15
    THOTCP16 THOTCP17 THOTCP18 THOTCP19 THOTCP20
    THOTCP21 THOTCP22 THOTCP23 THOTCP24 THOTCP25
```

```
    THOTCP26 THOTCP27 THOTCP28 THOTCP29 THOTCP30
ENDDEFAULTLUS
DEFAULTAPPL THOTSO ; Set the default appl.for all TELNET session
LINEMODEAPPL THOTSO ; Send all line mode terminals directly to TSO
ALLOWAPPL THOTSO* DISCONNECTABLE ; Allow all users access to TSO appl.
             ; TSO is multiple applications all beginning with TSO so u
             ; the * to get them all. If a session is closed, disconnec
             ; the user rather than log off the user.
; RESTRICTAPPL IMS ; Only three users may use IMS
;   USER USER1 ; Allow user1 access  uncommented 2/11/94 - rich
;   LU TCPIMS01 ; Assign USER1 LU TCPIMS01
;   USER USER2 ; Allow user2 access from the default LU pool
;   USER USER3 ; Allow user3 access from three telnet sessions, each wit
;              ; different reserved LU.
;     LU TCPIMS31 LU TCPIMS32 LU TCPIMS33
; ALLOWAPPL * ; Allow all applications that have not been previously
             ; specified to be accessed
ENDVTAM
  START PNL
  START CA0
  START CA1
  START CA2
  START CA3
  START CA5
  START CA6
  START lab2
  START RPERF
  START AIX41
  START AIX411
  START AIX412
  START AIX413
  START dhat1
  START mes2
```

A.3 TCPPARMS.FTPDATA

This is the FTP.DATA file used in Austin.

```
****** ***************************** Top of Data ******************************
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>           your edit profile using the command RECOVERY ON.
 000001 ; CHANGE LOG
 000002 ;   07/26/94 JW Wenzlaff
 000003 ;            -copy from TCPIP.V3R1.SEZAINST(FTPDATA)
 000004 ;
 000005 ;   10/20/94 JW Wenzlaff
 000006 ;            -change RETPD from 30 to 0
 000007 ;
 000008 ;*******************************************************************
 000009 ;                                                                  *
 000010 ;   Name of File:      FTP.DATA                                     *
 000011 ;                                                                  *
 000012 ;   This file, FTP.DATA, is used to specify default file and disk  *
 000013 ;   parameters used by the FTP client and server.  Different       *
 000014 ;   FTP.DATA files might be in effect, depending on where          *
 000015 ;   the FTP client and server are executing.                       *
 000016 ;                                                                  *
 000017 ;   Syntax Rules for the FTP.DATA Configuration File:              *
 000018 ;                                                                  *
```

```
 000019 ;   (a) All characters to the right of and including a ; will be      *
 000020 ;       treated as a comment.                                          *
 000021 ;                                                                      *
 000022 ;   (b) Blanks and <end-of-line> are used to delimit tokens.           *
 000023 ;                                                                      *
 000024 ;   (c) The format for each statement is:                               000025 ;
*
 000026 ;       parameter value                                                *
 000027 ;                                                                      *
 000028 ;   (d) abbreviations are not permitted in the FTP.DATA file           *
 000029 ;                                                                      *
 000030 ;                                                                      *
 000031 ;**********************************************************************
 000032 ;
 000033 ;
 000034 ; File and disk parameters
 000035 ;
 000036 Primary         100        ; Primary allocation is 5 tracks
 000037 Secondary       50         ; Secondary allocation is 2 tracks
 000038 Directory       15         ; PDS allocated with 15 directory blocks
 000039 Lrecl           128        ; Logical record length is 128 bytes
 000040 BlockSize       6144       ; Block size is 6144 bytes
 000041 AutoRecall      true       ; Migrated HSM files recalled automatically
 000042 AutoMount       true       ; Nonmounted volumes mounted automatically
 000043 DirectoryMode   false      ; Use all qualifiers (Datasetmode)
 000044 ;Volume         volume     ; Volume serial number for allocation
 000045 SpaceType       TRACK      ; Datasets allocated in tracks
 000046 Recfm           FB         ; Fixed blocked record format
 000047 ;DcbDSN         model.dcb  ; Dataset name used as model for all ocation
 000048 ;UnitName       SYSDA      ; Unit name used for allocation
 000049 Filetype        SEQ        ; File Type = SEQ (default)
 000050 SMF             70         ; The SMF record type to be used
 000051 RETPD           0          ; New dataset expiration date is 30  days
 000052 MGMTCLASS       TCPMGMT    ; SMS management class for new data sets
 000053 RDW             false      ; Do not retain RDWs as data
 000054 NCP             20         ; 15 I/O buffers
```

A.4 TCPIP.V3R1M0.HOSTS.LOCAL

This is the HOSTS.LOCAL file used in Austin.

```
; To update, refer to the use of the MAKESITE command in the TCP/IP
; Planning and Customization manual.
;
;
; change log:
; 12/16/93 jw wenzlaff
;          - created
; 12/21/93 jw wenzlaff
;          - rename POLYESTER to POLYESCON
; 12/22/93 jw wenzlaff
;          - add POLYPCA and MVSPCA
;
;
HOST : 9.3.72.11 : ESCONTEST ::::
HOST : 11.0.0.1 : POLYPCA ::::
HOST : 21.0.0.1 : MVSBLK ::::
HOST : 21.0.0.2 : MVSBLK1 ::::
```

```
HOST : 11.0.0.2 : MVSPCA ::::
;
HOST : 18.0.0.2 : MVSESC ::::
HOST : 18.0.0.1 : DHATRISC ::::
HOST : 10.0.0.2 : MVSPCADHO ::::
HOST : 10.0.0.1 : DHATPCAO ::::
;
HOST : 13.0.0.2 : MVSFVT ::::
HOST : 13.0.0.1 : FVTPCA ::::
```

A.5 CLIO.SFCFSAMP.FCFDEFS

This file allows the installation to specify how JCL is located and generated.

```
# START COPYRIGHT
# ********************************************************************
#
#     This module is "RESTRICTED MATERIALS OF IBM"
#     Licensed Materials - Property of IBM
#
#     5799-FET
#     5648-129
#     (C) COPYRIGHT IBM CORP. 1993, 1995
#     All rights reserved.
#     US Government Users Restricted Rights - Use,
#     duplication or disclosure restricted by GSA ADP
#     Schedule Contract with IBM Corp.
#
# ********************************************************************
# END COPYRIGHT
# This file allows the installation to specify how JCL is located and
# generated.
# When PARSRV starts a program on MVS the following steps are taken:
#
# 1. Read the file(s) $(USERDEF), where the user can set his own defaults.
# 2. Process the first existing file in $(USERJCL). This file normally
#    contains JCL for a particular program, but can also contain
#    instructions as in $(USERDEF).
# 3. If step 2 produced no JCL, expand the sections mentioned in
#    $(SECTIONS).
# 4. If $(CLIOENV) has value YES, expand a special section
#    which allows a slave to contact its master.
#
# Note: The variables USERDEF and USERJCL may hold more than one
# dataset name. The names must be separated by blanks.
#
# The INSTDEF, USERDEF and USERJCL files hold assignment statements,
# keyword statements and plain text. The latter is scanned for
# variables and added to the JCL. The keywords are:
#    .EXPAND <sections>      # expand the sections mentioned
#    .EXPAND                 # same as .EXPAND $(SECTIONS)
#    .INCLUDE file(s)        # include the files mentioned
#    .INLINE <delim>         # start an inline section
#    .INLINE                 # same as .INLINE $(.DELIM)
#    delim                   # end of inline section
#    .ECHO line              # echo line to stderr
#    .DEBUG flag(s)          # Update debug flags
#
# Any output that would be produced by INSTDEF is suppressed. The
```

```
# USERJCL file (and files included by it) is entered with .INLINE
# active.
#
# In an inline section, assignment and keyword statements are treated
# as plain text.  Variables are expanded unless the value of $(INLINE)
# is NOSUBST.  The inline section ends when the delimiter is found on
# a line by itself, starting in column one.
#
# To prevent recursion, .INCLUDE of a file that is active will be
# ignored. The offending list of files is written to stderr.
#
# The possible arguments of the .DEBUG keyword are:
#    JOB       - print a message when submitting a job
#    NOJOB     - opposite of JOB
#    ECHO      - echo JCL to stdout
#    NOECHO    - opposite of ECHO
#    SUBST     - indicate how variables are substituted
#    NOSUBST   - opposite of SUBST
#    CURFILE   - print a message when opening a file
#    NOCURFILE - opposite of CURFILE
#    OFF       - turn off ECHO, SUBST and CURFILE   (default)
#
# Variables:
# ----------
# Variables are referenced by surrounding the name with $( and ). Any
# spaces will be taken as part of the name. The name may itself be
# specified by a variable, e.g.  $($(VAR)EXT). Almost any character
# is permitted in a name. Names are case sensitive. It is possible
# to define a variable in a way that leads to recursion. This is
# detected, and the value returned is **RECUR**.
# If the value of a variable is a valid arithmetic expression, it is
# replaced by the numerical value of the expression.
# Undefined variables return an empty string.
# Note: Prefixing the name with '?' when referencing a variable
# will return the first blank-delimited string in the value. This
# provides an easy mechanism for overriding default dataset names.
# See the definition of PROGLIB for an example.
#
# Sections:
# ---------
# A section is a group of JCL cards. Inline sections are defined by
# using the .INLINE keyword. All other sections have a name. Multiple
# lines can be assigned to the name by treating it as a stem variable,
# i.e. by appending .1, .2, etc. to the name and assigning values to
# the resulting variables. Several examples are shown below.
# Note: Many sections have default values. If name.1 is redefined, the
# whole section is redefined. In some cases it may be desirable to
# redefine only parts of the section and keep the rest. This can be
# achieved by treating the desired component, e.g. name.2, as a stemmed
# variable and assigning values to name.2.1, name.2.2, etc.
#
# Examples:
# Define a new section SECT:
# SECT.1 = "//DD1      DD  DISP=SHR,DSN=$(USER).NEWSAMP.LIB"
# SECT.2 = "//DD2      DD  DISP=SHR,DSN=$(USER).SAMPLE.LIB"
#
# Redefine STEPLIB.1 and keep STEPLIB.2, STEPLIB.3 etc.
# STEPLIB.1.1 = "//STEPLIB  DD   DISP=SHR,DSN=$(USER).TEST.LOAD"
# STEPLIB.1.2 = "//         DD   DISP=SHR,DSN=$(PROGLIB)"
```

```
#
# Special sections:
# -----------------
#    STC               - Can be used for started tasks
#    CLIOENV           - Enables slave to contact master
#    C*A*N             - JCL for a program which cancels a job
#
# Assignments:
# ------------
# The format of an assignment statement is:
#   lhs = rhs
# Leading and trailing spaces surrounding lhs and rhs are removed.
# Any variables in lhs are expanded. Variables in rhs are expanded unless
# they are surrounded by quotes.  One layer of quotes is stripped off.
# Quotes can be single or double.  Single quotes inside double quotes,
# and vice versa, or quotes inside parenthesis are treated like a
# normal character. Unbalanced quotes are retained.
# The consequence of this is that the evaluation of variables inside
# quotes is delayed until the new variable is expanded, effectively
# turning it into a macro. This is very useful when defining defaults.
# A semicolon can be used to separate multiple assignment statements
# on the same line.
#
# Examples:
#    c = "'name.tst$(a)'"         # Value of $(c) is 'name.tst'
#    a = 5; b = $(a) + 1          # Value of $(b) is 6
#    a$(b) = $(a) * $(b)          # Value of $(a6) is 30
#    a = 7                        # Value of $(c) is 'name.tst7'
#    a = '$(a)'                   # Value of $(c) is 'name.tst**RECUR**'
#    t = Don't remove quote       # The quote is kept
#    s = first second
#    t = $(?s)                    # Value of t is first
#    t = $(s)                     # Value of t is first second
#
# Special variables:
# ------------------
# There are some special variables:
#    PROJECT           - Used to construct filenames. For user files, it
#                        can be overridden by setting $(PROG).PROJECT to
#                        the desired value in INSTDEF. Default is CLIO.
#    SECTIONS          - This variable defines which sections are
#                        expanded by an explicit .EXPAND ( with no args )
#              or when default JCL is produced. ( Step 3 above.)
#                        Default is START JOB JES EXEC STEPLIB SYSPRINT END.
#    $(PROG).SECTIONS  - When producing default JCL, it is checked if
#                        this variable exists. If it does, it is used
#              instead of $(SECTIONS).
#    CLIOENV           - If YES, append the special section CLIOENV at the
#              end of the job. This section allows the slave
#              to contact its master.
#              The builtin default is YES.
#    $(PROG).CLIOENV   - If it exists, this variable replaces $(CLIOENV).
#    INLINE          - Affects variable expansions in inline sections
#                 SUBST   - variables are expanded (default)
#                 NOSUBST - variables are not expanded
#    CURFILE           - Name of the file currently being processed
#    USER              - Current MVS userid
#    PW                - Current MVS password
#    PROG              - Used to select program to be run
```

```
#    $PARM             - Holds argument string from OPNSRV.
#    APPL              - Set to CLIO by PARSRV.
#    JOBLETTERS        - Defines the set of characters that are appended
#                        to $(USER) to produce the jobname.
#    JC                - Current element of $(JOBLETTERS)
#    MSTPORT           - Master's TCP/IP port number
#    MSTHID            - Master's TCP/IP Internet address
#    DATE              - Current date   ( day dd/mm/yy )
#    TIME              - Current time   ( hh:mm:ss )
#    NAME              - May hold the name of the current owner.
#                        Default is $(USER) or $($(USER).NAME) if it exists.
#    $(USER).NAME      - Name corresponding to $(USER).

#
#    S T A R T   O F   I N S T A L L A T I O N   D E F A U L T S
#
 ECHO "Processing INSTDEF: $(CURFILE) $(DATE) $(TIME)."
 DEBUG CURFILE JOB               # Echo message when opening files

### Variables that are set in each invocation before processing
### user files
# USER     # MVS userid
# PW       # MVS password for $(USER)
# PROG     # Identifies program to be run
# $PARM    # Command line arguments  ( PARM field in JCL )
# APPL     # Set to CLIO by PARSRV
# JC       # Set to next character in $(JOBLETTERS) if it exists
#          # else next character in "BCDEFGHIJKLMNOPQRTUVWXYZ0123456789"
# MSTPORT  # Master's port number         ( from PARSRV )
# MSTHID   # Master's Internet address    ( from PARSRV )
# MSTUSER  # Master's user id
# MSTPID   # Master's process id
# PROJECT  = "$($(PROG).PROJECT)"  # If defined in INSTDEF
# NAME     = "$($(USER).NAME)"     # If defined in INSTDEF

######## CLIO/S dataset defaults. Set these to match the installation
######## of your CLIO/S data sets.
  CLIOHLQ   = SYS1               # HLQ of CLIO datasets
  PROJECT   = CLIO               # Default project
  CLIODSN   = $(CLIOHLQ)         # CLIO datasets

######## Job letters that can be used in job names. Set these to values
######## that can be used in job names.
  JOBLETTERS= BCDEFGHIJKLMNOPQRTUVWXYZ0123456789

######## Variables in the job card. Set these to match the values needed
######## for your installation.
# $NAME    ="$(USER)$(JC)"
  $NAME    ="$(USER)$(JC)"
# $ACCT    = EPAC,1000
# $PID     = "$($(USER).NAME)"
# $CLASS   = CLASS=X,
# $MSGCLASS= MSGCLASS=R,
# $MSGLEVEL= MSGLEVEL=(1,1),
# $REGION  = REGION=0K,
  $REGION  = REGION=2M,
# $NOTIFY  =
# $TYPRUN  =
# $TIME    =
```

```
  $TIME    = TIME=1000,
# $USER    ="USER=$(USER),"
# $PW      ="PASSWORD=$(PW)"

######## STEPLIB datasets. Set these to match the corresponding data
######## sets at your installation.
# PROGLIB  ="$($(PROG).PROGLIB) $(USER).$(PROJECT).LOAD"
  PROGLIB  = SYS1.SFCFLOAD
  SEDCLINK = SYS1.SEDCLINK             # C370 SEDCLINK
  SEQALINK = SYS1.SEQALINK             # PLI SEQALINK
  SAMPRUN2 = VSPASCAL.VSPV1R2.SAMPRUN2 # VSPASCAL SAMPRUN2
  VSF2LOAD = SYS1.VSF2LOAD             # VSF2LOAD

######## Define files holding defaults and JCL for specific programs.
# INSTDEF   = "dd:DEFAULTS"               # Initial file processed
  INSTDEF   = "'$(CLIODSN).SFCFSAMP(FCFDEFS)'" # For PARREFR, if used
  INSTJCL   = "'$(CLIODSN).JCLGEN($(PROG))'"
  USERFILE  = "$(USER).$(PROJECT).JCLGEN"   # For user use
  USERFIL2  = "$(USER).API.CNTL"       # For compatibility
  USERJCL1  = "'$(USERFILE)($(PROG))'" # For user use
  USERJCL2  = "'$(USERFIL2)($(PROG))'" # For user use
  USERDEF   = "'$(USERFILE)($DEF$)' '$(USERFIL2)($DEF$)'"
  USERJCL   = "$(USERJCL1) $(USERJCL2) $(INSTJCL)"

######## JCL for TAPESRV
TAPESRV.SECTIONS = START JOB JES TAPESRV END
TAPELOAD = "$(CLIODSN).SFCFLOAD"
TAPESRV.1 ="//TAPESRV  EXEC PGM=$(PROG),PARM='NOSTAE,NOSPIE'"
TAPESRV.2 ="//STEPLIB  DD   DISP=SHR,DSN=$(TAPELOAD)"
TAPESRV.3 ="//         DD   DISP=SHR,DSN=$(SEQALINK)"
TAPESRV.4 ="//         DD   DISP=SHR,DSN=$(SEDCLINK)"
TAPESRV.5 ="//         DD   DISP=SHR,DSN=$(SAMPRUN2)"
TAPESRV.6 = //SYSUDUMP DD   SYSOUT=*,DCB=BLKSIZE=3429
TAPESRV.7 = //SYSPRINT DD   SYSOUT=*
TAPESRV.8 = //FTO6FO01 DD   SYSOUT=*

######## JCL for FCFFSERV
FCFFSERV.SECTIONS = START JES FCFFSERV END

######## JCL for CLFTP
CLFTP.SECTIONS = START JES CLFTP END

######## JCL for CLPLINK
CLPLINK.SECTIONS = START JES CLPLINK END

FCFFLD = "$(CLIODSN).SFCFLOAD"
FCFFSERV.1 = "//$($NAME) JOB ($($ACCT)),'$($PID)',$($CLASS)REGION=17M,"
FCFFSERV.2 = "//   $($MSGCLASS)$($MSGLEVEL)$($NOTIFY)$($TYPRUN)"
FCFFSERV.3 = "//   $($TIME)$($USER)$($PW)"
FCFFSERV.4 ="//FCFFSERV PROC"
FCFFSERV.5 ="//FCFFSERV EXEC PGM=IKJEFT01"
FCFFSERV.6 ="//STEPLIB  DD   DISP=SHR,DSN=$(FCFFLD)"
FCFFSERV.7 ="//SYSIN    DD   DUMMY"
FCFFSERV.8 ="//INSPLOG  DD   SYSOUT=4,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)"
FCFFSERV.9 ="//REXXLST  DD   DSN=$(CLIODSN).SFCFSAMP,DISP=SHR"
FCFFSERV.10 = //SYSUDUMP DD   SYSOUT=4
FCFFSERV.11 = //SYSPRINT DD   SYSOUT=4
FCFFSERV.12 = //SYSTSPRT DD   SYSOUT=4,DCB=BLKSIZE=3429
FCFFSERV.13 = //    PEND
```

```
FCFFSERV.14 = //P1  EXEC FCFFSERV
FCFFSERV.15 = //SYSTSIN  DD   *
FCFFSERV.16 ="   CALL '$(FCFFLD)(FCFFSERV)' '$(PARM2)'"


CLFLD = "$(CLIODSN).SFCFLOAD"
CLFTP.1 = "//$($NAME) JOB ($($ACCT)),'$($PID)',$($CLASS)REGION=17M,"
CLFTP.2 = "//   $($MSGCLASS)$($MSGLEVEL)$($NOTIFY)$($TYPRUN)"
CLFTP.3 = "//   $($TIME)$($USER)$($PW)"
CLFTP.4 = "//CLFTP EXEC PGM=$(PROG),"
CLFTP.5 = "// PARM='$(PARM2)'"
CLFTP.6 = "//STEPLIB  DD   DISP=SHR,DSN=$(CLFLD)"
CLFTP.7 = //SYSUDUMP DD   SYSOUT=4
CLFTP.8 = //SYSPRINT DD   SYSOUT=4

CLPLD = "$(CLIODSN).SFCFLOAD"
CLPLINK.1 = "//$($NAME) JOB ($($ACCT)),'$($PID)',$($CLASS)REGION=17M,"
CLPLINK.2 = "//   $($MSGCLASS)$($MSGLEVEL)$($NOTIFY)$($TYPRUN)"
CLPLINK.3 = "//   $($TIME)$($USER)$($PW)"
CLPLINK.4 = "//CLPLINK EXEC PGM=$(PROG),"
CLPLINK.5 = "// PARM='$(PARM2)'"
CLPLINK.6 = "//STEPLIB  DD   DISP=SHR,DSN=$(CLPLD)"
CLPLINK.7 = //SYSUDUMP DD   SYSOUT=4
CLPLINK.8 = //SYSPRINT DD   SYSOUT=4

######## This is the default JCL produced
# JOB.1     = "//$($NAME) JOB ($($ACCT)),'$($PID)',$($CLASS)$($REGION)"
# JOB.2     = "//   $($MSGCLASS)$($MSGLEVEL)$($NOTIFY)$($TYPRUN)"
# JOB.3     = "//   $($TIME)$($USER)$($PW)"
# EXEC.1    = "//$(PROG)  EXEC PGM=$(PROG)$(COND)$(PARM)"
  STEPLIB.1 = "//STEPLIB  DD   DISP=SHR,DSN=$(?PROGLIB)"
  STEPLIB.2 = "//         DD   DISP=SHR,DSN=$(SEDCLINK)"
  STEPLIB.3 = "//         DD   DISP=SHR,DSN=$(SAMPRUN2)"
  STEPLIB.4 = "//         DD   DISP=SHR,DSN=$(VSF2LOAD)"
# SYSPRINT.1=  //SYSPRINT DD   SYSOUT=*
# SYSPRINT.2=  //FTO6FO01 DD   SYSOUT=*

######## Assign names to some userids. Default is "$(USER)".
# NAME         = "$(USER)"             # Default name
  EPAC201.NAME = O. Skudal
  EPAC202.NAME = T. Rugland
  EPAC203.NAME = O. Gjerde
  EPAC312.NAME = F. Harloff

######## Sundry defaults
# .DELIM    = /*PROCESS*/         # End of inline block
# INLINE    = SUBST               # Substitute inline variables
# CLIOENV   = "$($(PROG).CLIOENV) YES"  # Expand CLIOENV DD

######## Define the default sections
# SECTIONS = START JOB JES EXEC STEPLIB SYSPRINT END
#
# Add ARGS to SECTIONS if arguments are wanted in a file
  ARGSDD    = "ARGS    "                    # Name of DD card
  ARGS.1    = "//$(ARGSDD) DD   *"
  ARGS.2    = "$(PROG) $($PARM)"
  ARGS.3    = "/*"
```

```
######## EXEC PARM field
# PARM     =",PARM='$($PARM)'"
PARM2    ="$($PARM)"
# Use PARM3 instead of PARM or PARM2 when a CLIOENV DD statement is
# not generated for FCFFSERV, CLFTP, and CLPLINK. This will occur
# when CLIOENV = NO.
PARM3    =",PARM='-c $(MstPort),$(MstHid) $(PARM2)'"

######## JCL for WTOSRV
WTOSRV.SECTIONS = $(TAPESRV.SECTIONS)

######## Sections for the CLIOCAN program
# C*A*N.SECTIONS = C*A*N;C*A*N.CLIOENV = NO    # Supress CLIOENV DD

# This should work if INTRDR has system authority (JES2:$T RDI,A=3 )
# JTYPE = "$($PARM)/100000;JOBNO = "$($PARM)-$(JTYPE)*100000"
# LETTER.2=J;LETTER.3=S;LETTER.4=T   # JOB/STC/TSO
# C*A*N.1 = "/*$C $(LETTER.$(JTYPE))$(JOBNO)"

######## JCL for running CLIOCAN
  C*A*N.1 = "//$($CANNAME) JOB ($($CANACCT)),'$($CANPID)',$($CANCLASS)"
  C*A*N.2 = "//   $($CANMSGCLASS)$($CANMSGLEVEL)$($CANNOTIFY)"
  C*A*N.3 = "//   $($TIME)$($USER)$($PW)"

######## Using the authorized program
# C*A*N.4 = "//FCFCAN   EXEC PGM=$(CANPROG),PARM='$($PARM)'"
# C*A*N.5 = "//STEPLIB  DD   DISP=SHR,DSN=$(CANPROGLIB)"
# C*A*N.6 = "//SYSPRINT DD   SYSOUT=*"

######## Using the REXX version of CLIOCAN
  C*A*N.4 = "//FCFCANC  EXEC PGM=IKJEFT01"
  C*A*N.5 = "//SYSPROC  DD   DISP=SHR,DSN=$(CLIODSN).REXX"
  C*A*N.6 = "//SYSTSPRT DD   SYSOUT=*"
  C*A*N.7 = "//SYSTSIN  DD   *"
  C*A*N.8 = "%FCFCANC $($PARM) $(USER)"  #Job ID and USERID
  C*A*N.9 = "/*"

######## Variables in the job card for CLIOCAN
# $CANNAME     = FCFCAN
# $CANACCT     = CLIO,1000
# $CANPID      = FCFCAN
  $CANCLASS    = CLASS=A,
  $CANMSGCLASS = MSGCLASS=4,
  $CANMSGLEVEL = MSGLEVEL=(0,0),
  $CANREGION   = REGION=2M,

  $CANPROG     = FCFCAN                  # Program name
  $CANPROGLIB  = SYS1.APFLINK            # Needs APF Authorization

######## Sections which can be used for started tasks ( $T RDI,A=3 )
# STC.1    = "/*$VS,'S $(PROG).$(USER),PRM=($(MSTPORT),$(MSTHID))'"
#
# The following should be used for FCFFSERV, CLFTP, and CLPLINK.
# STC.1    = "/*$VS,'S $(PROG).$(USER),PRM=($(PARM3))'"
#
#### Try to run TAPESRV as a started task
# TAPESRV.SECTIONS = STC; TAPESRV.CLIOENV = NO
# WTOSRV.SECTIONS  = STC; WTOSRV.CLIOENV = NO
#
```

```
# Note: This presumes that you have JCL for TAPESRV in one of your
#    PROCLIB datasets.  (Try 'S TAPESRV' from the console or sdsf if
#    you want to make sure TAPESRV is there.)
#
#    If you want the task name to be FCFTAPES, try the following:
#    TAPESRV.SECTIONS = TAPESTC; TAPESRV.CLIOENV = NO
#    TAPESTC.1= "/*$VS,'S FCFTAPES.$(USER),PRM=($(MSTPORT),$(MSTHID))'"
#
# Ensure that you have no TAPESRV member in JCLGEN which overrides this
# definition.
#
# Any C program that runs as a started task should have the following
# code at the beginning of it.
# #if defined(MVS)
#    Clioargc = argc;   /* argc is argument count passed to main() */
#    Clioargv = argv;   /* argv is argument ptrs passed to main() */
# #endif
#
#### Try to run FCFIDUM as a started task. The FCFIDUM proc will be
#    started rather than submitting JCL. The FCFIDUM proc must be
#    available in one of your PROCLIB datasets. (Try 'S FCFIDUM' from
#    the console or sdsf if you want to make sure FCFIDUM is there.)
# FCFIDUM.SECTIONS = STC; FCFIDUM.CLIOENV = NO
#
# The following JCL should not be uncommented. It is an example PROC
# that can be used to start FCFIDUM. Any PROC used to start a started
# task must be included in the system PROCLIB concatenation.
#
#   Example PROC to start FCFIDUM:
#   //FCFIDUM  PROC PRM=
#   //*--------------------------------------------------------------
#   //*
#   //* FUNCTION:
#   //*   STARTED PROGRAM FCFIDUM.
#   //*
#   //* INSTRUCTIONS:
#   //*   1. IF NECESSARY, CHANGE SYS1.SFCFLOAD TO BE THE NAME OF THE
#   //*      FILE CONTAINING THE FCFIDUM LOAD MODULE.
#   //*   2. CHANGE USER.CLIOENV TO BE THE NAME OF THE FILE CONAINING
#   //*      CLIOENV VARIABLES.
#   //*
#   //*--------------------------------------------------------------
#   //*
#   //FCFIDUM  EXEC PGM=FCFIDUM,PARM=&PRM
#   //STEPLIB  DD DSN=SYS1.SFCFLOAD,DISP=SHR
#   //CLIOENV  DD DSN=USER.CLIOENV,DISP=SHR
#   //SYSPRINT DD SYSOUT=4
#   //   PEND

######## Variables that are redefined before generating default JCL
# SECTIONS = "$($(PROG).SECTIONS)" # If defined
```

A.6 CLIO.SFCFSAMP.FCFREXX

This is the CLIO.SFCFSAMP.FCFREXX file used in Austin. This REXX exec is provided as sample code with the CLIO/S product.

```
/*********************************************************************
*
*     This module is "RESTRICTED MATERIALS OF IBM"
*     Licensed Materials - Property of IBM
*
*     5648-129
*     (C) COPYRIGHT IBM CORP.  1995
*     All rights reserved.
*     US Government Users Restricted Rights - Use,
*     duplication or disclosure restricted by GSA ADP
*     Schedule Contract with IBM Corp.
*
*********************************************************************/
/* End copyright. */
/*********************************************************************
*
* This REXX exec is provided as sample code with the CLIO/S product.
* When called by FCFPARSR, it produces JCL statements, writes them
* into a JES internal reader, and submits the JCL by closing the reader.
* The submitted JCL starts a servant process requested by a CLIO/S
* master process.
*
* The JCL created by this exec will work 'as is' if your installation
* placed CLIO/S components in the default datasets. If not, the STEPLIB
* variables must be modified accordingly.
*
* This exec must return a value > 0, otherwise FCFPARSR will report
* back to the master process that the servant creation process failed.
*
* See the CLIO/S User's Guide for more information on this exec.
*
*********************************************************************/

numargs = arg()                         /* How many arguments?        */
do i = 1 to numargs                     /* Save the arguments.        */
   p.i = arg(i)
   end
call initialize                         /* Do some preparation.       */

/*********************************************************************
* Now set some variables that will be used during JCL generation.
* Change these as desired.
*********************************************************************/
account_info = '(572260,6MM,202-3)'
class        = 'CLASS=A'
region       = 'REGION=2M'
msgclass     = 'MSGCLASS=4'
msglevel     = 'MSGLEVEL=(1,1)'
NOTIFY       = 'NOTIFY=' || DEMO2
user         = 'USER=' || DEMO2
time         = 'TIME=1000'
passwd       = 'PASSWORD=' || COU1RT
jobname = DEMO2||substr(jobstring,(SRVNUM-1)//length(jobstring)+1,1)
                                        /* Choose jobletter based on  */
```

```
                                        /*   SRVNUM, append to userid */
                                        /*   to create jobname.       */
/*********************************************************************
* The following STEPLIB datasets should reflect your own installation.
*********************************************************************/
clioload = 'CLIO.SFCFLOAD'        /* CLIO/S load modules             */
cliosamp = 'CLIO.SFCFSAMP'        /* CLIO/S sample code              */
tapeload = 'CLIO.SFCFLOAD'        /* CLIO/S load modules             */
sedclink = 'SYS1.SEDCLINK'        /* C runtime libraries             */
seqalink = 'SYS1.SIBMLINK'        /* PL1 runtime libraries           */
vsf2load = 'SYS1.VSF2LOAD'        /* Fortran runtime libraries       */

/*********************************************************************
*
* Call the appropriate JCL generation routine, based on the name of
* the application to be started.
*
* Valid JCL statements must be produced. All statements must be 80
* characters or less in length.
*
* JCL statements are placed into the stem variable 'jcl'.  The entire
* stem will then be written into the JES internal reader in one
* operation. (This is not a requirement. It would also be possible to
* write statements one at a time into the reader as they are created.)
*
* In this sample code, different methods of filling the stem variables
* with JCL statements are demonstrated.
*
* The JCL produced closely matches that produced by the older CLIO/S
* JCL generation routine.
*
* For started tasks, see the routine 'stask' below. It is not used by
* any of the default code, and will have to adapted for your own
* environment.
*
*********************************************************************/
select;
  when SRVPROG == 'TAPESRV'  /* CLIO/S Tape server.                */
     then call tapesrv

  when SRVPROG == 'FCFFSERV' /* CLIO/S FTP server.                 */
     then call fcffserv

  when SRVPROG == 'CLPLINK'  /* CLIO/S Pipe link.                  */
     then call clplink

  when SRVPROG == 'CLFTP'    /* CLIO/S FTP client.                 */
     then call clftp

  when SRVPROG == 'C*A*N'    /* CLIO/S Cancel routine.             */
     then call fcfcanc

  otherwise call default     /* Default JCL.                       */
  end;                       /* end select                         */

/*********************************************************************
*
* Write the JCL statements into the JES internal reader.
*
```

```
*********************************************************************/
'EXECIO * DISKW FCFJCL ( STEM JCL.'      /* Write JCL to FCFJCL.     */
if rc <> 0 then do
   say 'Write error:' rc
   exit 0        /* Will cause FCFPARSR to report failure to master.  */
   end

return jcl_lines  /* WARNING: return value must be > 0, otherwise    */
                  /*    FCFPARSR will report failure back to master  */
                  /*    process.                                     */
exit

/*********************************************************************/
default:
/*********************************************************************
*
* This subroutine calls another subroutine to store the JCL strings.
* The 'j' subroutine maintains it's own count, so we don't have to.
*
*********************************************************************/
call j("//"jobname  "JOB" account_info",'"MVSUSER"',"class","region",")
call j("//   "msgclass","msglevel","notify",")
call j("//   "time","user","passwd)
call j("//"SRVPROG"   EXEC PGM="SRVPROG",")
call j("// PARM='"SRVPARMS"'")
call j("//STEPLIB  DD   DISP=SHR,DSN="clioload)
call j("//         DD   DISP=SHR,DSN="sedclink)
call j("//         DD   DISP=SHR,DSN="vsf2load)
call j("//SYSPRINT DD   SYSOUT=*")
call j("//SYSUDUMP DD   SYSOUT=*")
call j("//SYSDUMP  DD   SYSOUT=*")

if datatype(start_env, 'W') = 1        /* Any environment variables? */
   then do
   call j("//CLIOENV  DD   *")         /* Create CLIOENV DD card,    */
   do i = start_env to numargs         /*   write variables into it. */
      call j(p.i)
      end i
   call j("/*")                        /* Terminate the DD card.     */
   end

call j(rexxtag)                        /* Information only.          */

call print_jcl                 /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
j: /* Add a string to 'jcl.' array, update count of JCL lines created.*/

parse arg argstring
   jcl_lines = jcl_lines + 1
   jcl.jcl_lines = argstring

return

/*********************************************************************/
fcffserv:
/*********************************************************************
```

```
*
* This subroutine determines the position of each JCL string explicitly.
* Be sure to keep the numbering correct if this code is modified.
*
*********************************************************************/

SRVPARMS2 = "-c "MstPort","MstHid SRVPARMS /* Use SRVPARMS2 instead of
                              SRVPARMS if CLIOENV will not be passed
                              through the JCL (ie if add_env is not
                              called below). */
jcl.1  = "//"jobname  "JOB" account_info","'"MVSUSER"'","class","
jcl.2  = "//   REGION=17M,"msgclass","msglevel","notify","
jcl.3  = "//   "time","user","passwd
jcl.4  = "//"SRVPROG"  PROC"
jcl.5  = "//"SRVPROG"  EXEC PGM=IKJEFT01"
jcl.6  = "//STEPLIB  DD   DISP=SHR,DSN="clioload
jcl.7  = "//SYSIN    DD   DUMMY"
jcl.8  = "//INSPLOG  DD   SYSOUT=4,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)"
jcl.9  = "//REXXLST  DD   DSN="cliosamp",DISP=SHR"
jcl.10 = "//SYSUDUMP DD   SYSOUT=*"
jcl.11 = "//SYSPRINT DD   SYSOUT=*"
jcl.12 = "//SYSTSPRT DD   SYSOUT=4,DCB=BLKSIZE=3429"
jcl.13 = "//    PEND"
jcl.14 = "//P1  EXEC "SRVPROG
jcl.15 = "//SYSTSIN  DD   *"
jcl.16 = "   CALL '"clioload"("SRVPROG")' '"SRVPARMS"'"
jcl_lines = 16

call add_env                    /* Add environment variables */
call print_jcl                  /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
clplink:
/*********************************************************************
*
* This subroutine determines the position of each JCL string explicitly.
* Be sure to keep the numbering correct if this code is modified.
*
*
*********************************************************************/
SRVPARMS2 = "-c "MstPort","MstHid SRVPARMS /* Use SRVPARMS2 instead of
                              SRVPARMS if CLIOENV will not be passed
                              through the JCL (ie if add_env is not
                              called below). */
jcl.1  = "//"jobname  "JOB" account_info","'"MVSUSER"'","class","
jcl.2  = "//   REGION=17M,"msgclass","msglevel","notify","
jcl.3  = "//   "time","user","passwd
jcl.4  = "//"SRVPROG"  EXEC PGM="SRVPROG","
jcl.5  = "// PARM='"SRVPARMS"'"
jcl.6  = "//STEPLIB  DD   DISP=SHR,DSN="clioload
jcl.7  = "//SYSUDUMP DD   SYSOUT=*"
jcl.8  = "//SYSPRINT DD   SYSOUT=*"
jcl_lines = 8

if CLPLINKX <> "CLPLINKX" then
  do
    /* CLPLINK provides a mechanism to pass information to JCL
```

```
       generation through the CLPLINKX variable. The value for CLPLINKX
       is set to whatever was passed in though the -X option on the
       CLPLINK command. You will only enter this section of the code
       if the -X option was used. You can use the values passed in
       CLPLINKX to generate additional JCL. */
  end

call add_env                    /* Add environment variables */
call print_jcl                  /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
clftp:
/*********************************************************************
*
* This subroutine determines the position of each JCL string explicitly.
* Be sure to keep the numbering correct if this code is modified.
*
*********************************************************************/
SRVPARMS2 = "-c "MstPort","MstHid SRVPARMS /* Use SRVPARMS2 instead of
                            SRVPARMS if CLIOENV will not be passed
                            through the JCL (ie if add_env is not
                            called below). */
jcl.1  = "//"jobname  "JOB" account_info",'"MVSUSER"','"class","
jcl.2  = "//   REGION=17M,"msgclass","msglevel","notify","
jcl.3  = "//   "time","user","passwd
jcl.4  = "//"SRVPROG"  EXEC PGM="SRVPROG","
jcl.5  = "// PARM='"SRVPARMS"'"
jcl.6  = "//STEPLIB  DD   DISP=SHR,DSN="clioload
jcl.7  = "//SYSUDUMP DD   SYSOUT=*"
jcl.8  = "//SYSPRINT DD   SYSOUT=*"
jcl_lines = 8

call add_env                    /* Add environment variables */
call print_jcl                  /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
tapesrv:
/*********************************************************************
*
* This subroutine reads the REXX source for comments containing the
* JCL statements, then substitutes !-delimited variables with their
* values. In this implementation '!' can only be used a delimiter,
* and there can be no comment delimiters in the jcl.
*
*********************************************************************/
jcl_in = _sigl() + 1    /* _sigl returns line number of caller       */
/** start of comment ************************************************
//!jobname!  JOB !account_info!,!MVSUSER!,!class!,!region!,!
//   !msgclass!,!msglevel!,!notify!,
//   !time!,!user!,!passwd!
//!SRVPROG!   EXEC PGM=!SRVPROG!,PARM='NOSTAE,NOSPIE'
//STEPLIB  DD   DISP=SHR,DSN=!clioload!
//         DD   DISP=SHR,DSN=!sedclink!
//SYSUDUMP DD   SYSOUT=*,DCB=BLKSIZE=3429
```

```
//SYSPRINT DD   SYSOUT=*
*** end of comment ************************************************/

line = sourceline(jcl_in)              /* Read the FCFREXX source.   */
if pos('/*', line) = 0 then say 'Comment expected'

maxlines = sourceline()                /* Last line in program.      */
do until jcl_in >= maxlines            /* Stop at end of file.       */
   jcl_in = jcl_in + 1
   line = strip(sourceline(jcl_in), 'T')
   if pos('*/', line) <> 0 then leave  /* Stop at end of comment.    */
   if pos('!', line) <> 0 then line = change(line)
   jcl_lines = jcl_lines + 1
   jcl.jcl_lines = line
   end

if datatype(start_env, 'W') = 1        /* Any environment variables? */
   then do
   jcl_lines = jcl_lines + 1
   jcl.jcl_lines = "//CLIOENV  DD   *" /* Create CLIOENV DD card,    */
   do i = start_env to numargs         /*   write variables into it. */
      jcl_lines = jcl_lines + 1
      jcl.jcl_lines = p.i
      end i
   jcl_lines = jcl_lines + 1
   jcl.jcl_lines = "/*"                /* Terminate the DD card.     */
   end

jcl_lines = jcl_lines + 1
jcl.jcl_lines = rexxtag                        /* Info only.        */

call print_jcl                 /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
fcfcanc:
/*********************************************************************
*
* This subroutine determines the position of each JCL string explicitly.
* Be sure to keep the numbering correct if this code is modified.
*
* Notes about FCFCANC:
*    -  Default MSGCLASS is Z, meaning job output is discarded
*    -  The JCL generated here calls FCFCANC, which is another
*       REXX exec. You could just as well insert FCFCANC code here.
*       This was not done in order to match the previous method.
*********************************************************************/
msgclass = 'MSGCLASS=Z'
jcl.1  = "//FCFCANC JOB" account_info",'"ID"',"class","region","
jcl.2  = "//   "msgclass","msglevel","notify","
jcl.3  = "//   "time","user","passwd
jcl.4  = "//FCFCANC   EXEC PGM=IKJEFT01"
jcl.5  = "//SYSPROC  DD   DISP=SHR,DSN="cliosamp
jcl.6  = "//SYSTSPRT DD   SYSOUT=*"
jcl.7  = "//SYSTSIN  DD   *"
jcl.8  = "%FCFCANC" SRVPARMS
jcl.9  = "/*"
```

```
jcl_lines = 9

jcl_lines = jcl_lines + 1
jcl.jcl_lines = rexxtag                          /* Info only.        */

call print_jcl                  /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
stask:
/*********************************************************************
*
*  Run the servant process as a started task. None of the defaults use
*  this method, but it is provided here as a template. This template
*  must be modified before use. See the User's Guide for discussion of
*  servants as started tasks.
*
*********************************************************************/
jcl.1  = "//FCFSTASK JOB" account_info",'"ID"',"class","region","
jcl.2  = "//   "msgclass","msglevel","notify","
jcl.3  = "//   "time","user","passwd
jcl.4  = "/*$VS,'S "SRVPROG"."MVSUSER",PRM=("MstPort","MstHid")'"

/* Started tasks don't get the CLIOENV DD card. See User's Guide.    */

call print_jcl                  /* Copy JCL to SYSTSPRT, if desired. */

return

/*********************************************************************/
_sigl: return sigl   /* get the source line number of calling line    */
/*********************************************************************/
change:  /* Substitute !-delimited variables with their values.       */
parse arg line
do until pos('!', line) = 0
   parse var line first '!' var '!' rest
   if var <> '' then line = first || value(var) || rest
   else line = first
   end
return line

/*********************************************************************/
add_env:   /* Add environment variables to jcl.                       */
if datatype(start_env, 'W') = 1        /* Any environment variables? */
   then do
   jcl_lines = jcl_lines + 1
   jcl.jcl_lines = "//CLIOENV  DD   *" /* Create CLIOENV DD card,    */
   do i = start_env to numargs         /*   write variables into it. */
      jcl_lines = jcl_lines + 1
      jcl.jcl_lines = p.i
      end i
   jcl_lines = jcl_lines + 1
   jcl.jcl_lines = "/*"                /* Terminate the DD card.     */
   end

jcl_lines = jcl_lines + 1
jcl.jcl_lines = rexxtag                          /* Info only.        */
```

```
return
/*********************************************************************/
print_jcl: /* Copy JCL to SYSTSPRT, but hide password.                */

say copies('-', 80) /* eyecatching separator */

do i = 1 to jcl_lines
   if (pos('PASSWORD='||MVSPW, jcl.i)) > 0
      then do until pos('PASSWORD='||MVSPW, temp) == 0
      if length(MVSPW) = 0 then leave
      temp = jcl.i
      parse var temp first 'PASSWORD=' rest
      lpw = length(MVSPW)
      rest = copies('*', lpw) || substr(rest, lpw+1)
      temp = first || 'PASSWORD=' || rest
      say temp
      end
   else say jcl.i               /* Copy JCL to SYSTSPRT, if desired. */
   end

return
/*********************************************************************/
initialize:

parse source . . . . . environment . /* Just for information,      */
                                     /*    could be 'MVS' or 'TSO'.*/
jcl_lines = 0
jobstring = 'BCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'
rexxtag = "//* JCL generated by FCFREXX on" date('U') "at" time('C')

do i = 1 to numargs                  /* Process the arguments.     */
   if substr(p.i, 1, 7) == 'MstPort'   /* Start of Environment data, */
      then start_env = i             /*    capture for later use.  */
   if substr(p.i, 1, 11) == 'SRVPARMS = ' /* Check for SRVPARMS.     */
      then do
        SRVPARMS = substr(p.i, 13, length(p.i)-13) /* Use substr so  */
                                     /*    that any quotes in      */
                                     /*    SRVPARMS will not cause */
                                     /*    confusion.              */
      end
   else if substr(p.i, 1, 9) == 'CLPLINKX=' /* Check for CLPLINKX    */
      then do
        CLPLINKX = substr(p.i, 10, length(p.i)-9) /* Use substr so   */
                                     /*    that any quotes or      */
                                     /*    arithmetic symbols in   */
                                     /*    CLPLINKX will not cause */
                                     /*    confusion.              */
      end
   else if substr(p.i, 1, 9) == 'CLFTPSRV=' /* Check for CLFTPSRV    */
      then do
        CLFTPSRV = substr(p.i, 10, length(p.i)-9) /* Use substr so   */
                                     /*    that any quotes or      */
                                     /*    arithmetic symbols in   */
                                     /*    CLFTPSRV will not cause */
                                     /*    confusion.              */
      end
   else do
     interpret p.i                   /* Execute the parameter,     */
                                     /*    if desired.             */
```

```
    end
  end
return

    .
```

A.7 SYS1.PARMLIB.IEFSSN00

This is the SYS1.PARMLIB.IEFSSN00 file used in Austin.

```
********************************* Top of Data ****************************
 ==MSG> -Warning- The UNDO command is not available until you change
 ==MSG>           your edit profile using the command RECOVERY ON.
 000001 SMS,IGDSSIIN,'ID=00,PROMPT=NO'
 000010 JES2,,,PRIMARY
 000020 BP01                                    BATCHPIPES/MVS
 001900 TNF,MVPTSSI                             TCP/IP
 002000 VMCF,MVPXSSI,MVSESC                     TCP/IP
 002100 RACF,IRRSSI00,'#'                       RACF 1.9.2
 002200 CLAW,CLWXSSI,SYS1.PARMLIB(CLAWPARM)     MVS/CLAW
******************************** Bottom of Data ***************************
```

A.8 SYS1.PROCLIB.PARSV2

The is the renamed FCFPARSR file used in Austin.

```
//FCFPARSR EXEC PGM=FCFPARSR
//STEPLIB  DD   DISP=SHR,DSN=CLIO.SFCFLOAD DATA S                       ET
//         DD   DISP=SHR,DSN=SYS1.SEDCLINK    C/370 OR LE/370 RUNTIME
//*        DD   DISP=SHR,DSN=SYS1.SEQALINK    INSPECT DATASET, OPTIONAL
//*        DD   DISP=SHR,DSN=SYS1.SAMPRUN2 VSPASCAL RUNTIME
//SYSPRINT DD   SYSOUT=4,DCB=BLKSIZE=3429
//SYSTSPRT DD   SYSOUT=4,DCB=BLKSIZE=3429
//INSPLOG  DD   SYSOUT=4,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//DDTERM   DD   DISP=SHR,DSN=CLIO.SFCFSAMP(FCFDDT)
//*DEFAULTS DD   DISP=SHR,DSN=CLIO.SFCFSAMP(FCFDEFS)
//REXXJCL  DD   DISP=SHR,DSN=CLIO.SFCFSAMP             (FCFREXX)
//*                                          REXX JCL GENERATION METHOD
```

A.9 LSCFG of Test RS/6000.

this is the output of the lscdg command.

```
INSTALLED RESOURCE LIST WITH VPD

The following resources are installed on your machine.

  sysplanar0        00-00             CPU Planar

        Part Number.................052G5691
        EC Level....................00D26537
        Processor Identification....00000111
        ROS Level and ID............IPLVER1.2 LVL0.00,08185543
        Processor Component ID......040200800000000E
        Device Specific.(Z0)........012055
        Device Specific.(Z1)........024057
        Device Specific.(Z2)........032054
```

```
      Device Specific.(Z3).....041052
      Device Specific.(Z4).....CD234D
      Device Specific.(Z5).....ED234D
      Device Specific.(Z6).....0A2356
      Device Specific.(Z7).....2A2356
      Device Specific.(Z8).....4A2356
      Device Specific.(Z9).....6A2356
      ROS Level and ID.....OCS(08000F00)
      ROS Level and ID.....SEEDS(00000000)

fpa0            00-00           Floating Point Processor
mem1            00-0D           64 MB Memory Card
mem0            00-0H           64 MB Memory Card
ioplanar0       00-00           I/O Planar

      EC Level.....39

bus0            00-00           Microchannel Bus
sio0            00-00           Standard I/O Planar

      Part Number..... 52G5814
      EC Level.....D26536
      Serial Number.....00000092
      FRU Number..... 52G5826
      Manufacturer.....IBM97N
      Displayable Message.....STANDARD I/O
      Device Driver Level.....00
      Diagnostic Level.....00
      ROS Level and ID.....0000
      Displayable Message..... STANDARD SCSI
      Device Driver Level.....00
      Diagnostic Level.....02
      Loadable Microcode Level....0044
      ROS Level and ID.....0054
      Read/Write Register Ptr.....0100

fda0            00-00-0D        Standard I/O Diskette Adapter
fd0             00-00-0D-00     Diskette Drive
ppa0            00-00-0P        Standard I/O Parallel Port Adapter
lp0             00-00-0P-00     IBM 4201 Model 3 Proprinter III
scsi0           00-00-0S        Standard SCSI I/O Controller
cd0             00-00-0S-00     CD-ROM Drive

      Manufacturer.....IBM
      Machine Type and Model......CD-ROM DRIVE:XM
      ROS Level and ID.....0242
      Device Specific.(Z0).....058001015B000088

rmt0            00-00-0S-50     5.0 GB 8mm Tape Drive

      Manufacturer.....EXABYTE
      Machine Type and Model......IBM-8505
      Device Specific.(Z1).....46LA
      Part Number.....46G2665
      Serial Number.....06021465
      Device Specific.(LI).....00000001
      EC Level.....896774
      FRU Number.....17G1715
      Device Specific.(Z0).....0180020283000010
```

```
rmt1            00-00-0S-60      150 MB 1/4-Inch Tape Drive

      Manufacturer.............TANDBERG
      Machine Type and Model...... TDC 3600
      Part Number.............21F8627
      Device Specific.(Z0).....018001002E000000
      Device Specific.(Z1).....U07:
      Device Specific.(Z2).....49

sa0             00-00-S1         Standard I/O Serial Port 1
tty0            00-00-S1-00      Asynchronous Terminal
sa1             00-00-S2         Standard I/O Serial Port 2
tty1            00-00-S2-00      Asynchronous Terminal
tok0            00-01            Token-Ring High-Performance Adapter

      Network Address..........10005AA8BA6D
      Displayable Message.....TOKEN RING
      EC Level................C24551
      FRU Number..............022F9380
      Manufacturer............VENOCLT96G
      Part Number.............074F8653
      Serial Number...........044697
      ROS Level and ID........0000
      Loadable Microcode Level....00

fddi0           00-02            FDDI Primary Card, Single Ring Fiber

      Network Address..........10005AB807C5
      Displayable Message..... FDDI RING B
      EC Level................C73876
      FRU Number.............. 81F9003
      Manufacturer............VEN0857049
      Part Number............. 31G9393
      Serial Number...........001968
      ROS Level and ID........0000
      Loadable Microcode Level....01

cat0            00-04            370 Parallel Channel Adapter

      Part Number.............02G7418
      EC Level................C26858
      FRU Number..............02G7425
      Manufacturer............IBM982
      Serial Number...........002285
      ROS Level and ID........9000
      Loadable Microcode Level....0000
      Device Driver Level.....0000
      Diagnostic Level........0000
      Displayable Message.....SYSTEM/370 HOST INTERFACE A

escon0          00-05            ESCON Channel Adapter

      Part Number............. 99F3765
      Serial Number...........
      EC Level................C49490
      Manufacturer............IBM982
      ROS Level and ID........03
      Loadable Microcode Level....00
```

```
      Device Driver Level.........00
      Diagnostic Level............00
      FRU Number.................. 43G0235
      Device Specific.(ZF)........ 43G0236
      Displayable Message.........GRAPE

esca0             00-05             ESCON Adapter(Control Unit Image)
chna0             00-07             IBM S/370 Channel Emulator/A Adapter
scsi1             00-08             SCSI I/O Controller

      Device Driver Level.........00
      Diagnostic Level............03
      Displayable Message.........SCSI-2
      EC Level....................D25855
      FRU Number.................. 52G5484
      Manufacturer................IBM97N
      Part Number.................. 52G5483
      Serial Number...............00013855
      Loadable Microcode Level....00A0
      ROS Level and ID............0005
      Read/Write Register Ptr.....0100

hdisk0            00-08-00-00       2.0 GB SCSI Disk Drive

      Manufacturer................IBM
      Machine Type and Model......0664M1H
      Part Number.................86F0101
      ROS Level and ID............35203530
      Serial Number...............00006214
      EC Level....................895118
      FRU Number..................86F0118
      Device Specific.(Z0)........000002029F00001E
      Device Specific.(Z1)........86F0620
      Device Specific.(Z2)........0980
      Device Specific.(Z3)........92323
      Device Specific.(Z4)........0002
      Device Specific.(Z5)........22
      Device Specific.(Z6)........895108

hdisk1            00-08-00-10       2.0 GB SCSI Disk Drive

      Manufacturer................IBM
      Machine Type and Model......0664M1H
      Part Number.................86F0101
      ROS Level and ID............35203530
      Serial Number...............00006133
      EC Level....................895118
      FRU Number..................86F0118
      Device Specific.(Z0)........000002029F00001E
      Device Specific.(Z1)........86F0620
      Device Specific.(Z2)........0980
      Device Specific.(Z3)........92323
      Device Specific.(Z4)........0002
      Device Specific.(Z5)........22
      Device Specific.(Z6)........895108

hdisk4            00-08-00-20       Other SCSI Disk Drive

      Manufacturer................IBM
```

```
      Machine Type and Model......0664
      Part Number.................
      ROS Level and ID............35202020
      Serial Number...............00000000
      EC Level....................
      FRU Number..................
      Device Specific.(Z0)........000002029F00001E
      Device Specific.(Z1)........
      Device Specific.(Z2)........
      Device Specific.(Z3)........
      Device Specific.(Z4)........
      Device Specific.(Z5)........
      Device Specific.(Z6)........

ioplanar1         00-10             I/O Planar

      EC Level....................30

bus1              00-10             Microchannel Bus
scsi2             00-12             SCSI I/O Controller

      Device Driver Level.........00
      Diagnostic Level............03
      Displayable Message.........SCSI-2
      EC Level....................D25855
      FRU Number.................. 52G5484
      Manufacturer................IBM97N
      Part Number.................. 52G5483
      Serial Number...............00005167
      Loadable Microcode Level....00A0
      ROS Level and ID............0005
      Read/Write Register Ptr.....0100

hdisk3            00-12-00-20       1.37 GB SCSI Disk Drive

      Manufacturer................IBM
      Machine Type and Model......ST41600N
      Part Number.................2437585
      ROS Level and ID............31310300
      Serial Number...............00077028
      EC Level....................835324
      Device Specific.(Z0)........000002028300001A
      Device Specific.(Z1)........70899604
      Device Specific.(Z2)........09VE
      Device Specific.(Z3)........92134

rmt2              00-12-00-50       2.3 GB 8mm Tape Drive

      Manufacturer................EXABYTE
      Machine Type and Model......EXB-8200
      Part Number.................21F8842
      Device Specific.(Z0)........0180010133000000
      Device Specific.(Z1)........2680

chna1             00-13             IBM S/370 Channel Emulator/A Adapter
s3701             00-13-00          IBM S/370 Channel Emulator/A Tape
                                    Driver
cat1              00-14             370 Parallel Channel Adapter
```

```
      Part Number................. 02G7418
      EC Level....................C26858
      FRU Number.................. 02G7425
      Manufacturer................VEN1612450
      Serial Number...............004901
      ROS Level and ID............9000
      Loadable Microcode Level....0000
      Device Driver Level.........0000
      Diagnostic Level............0000
      Displayable Message......... SYSTEM/370 HOST INTERFACE

escon1            00-16             ESCON Channel Adapter

      Part Number.................99F3609
      Serial Number...............
      EC Level....................C46970
      Manufacturer................IBM982
      ROS Level and ID............00
      Loadable Microcode Level....00
      Device Driver Level.........00
      Diagnostic Level............00
      FRU Number..................99F3609
      Displayable Message.........GRAPE

esca1             00-16             ESCON Adapter(Control Unit Image)
sysunit0          00-00             System Unit

      Machine Type and Model......ME7015-990
      Serial Number...............ME02603625
      User Data...................MEtest
```

Appendix B. VM TCP/IP Configuration Files

This sections shows the VM configurations files used to configure TCP/IP.

B.1.1 Profile TCPIP

This is the PROFILE.TCPIP file used in Austin.

```
ACBPOOLSIZE                 1000
ADDRESSTRANSLATIONPOOLSIZE  1500
CCBPOOLSIZE                 150
DATABUFFERPOOLSIZE          300 32768
ENVELOPEPOOLSIZE            750
IPROUTEPOOLSIZE             300
LARGEENVELOPEPOOLSIZE       300 8192
RCBPOOLSIZE                 50
SCBPOOLSIZE                 256
SKCBPOOLSIZE                256
SMALLDATABUFFERPOOLSIZE     0
TCBPOOLSIZE                 256
UCBPOOLSIZE                 100

; Flush the arp tables every 5 minutes
ARPAGE 5
;
; The SYSCONTACT and SYSLOCATION statements are used for SNMP.
;
; SYSCONTACT is the contact person for this managed node and how to
; contact this person.  Used for VM agent MIB variable sysContact
SYSCONTACT
   DHAT HELPDESK (823-6300)
ENDSYSCONTACT

; SYSLOCATION is the physical location of this node.  Used for VM
; agent MIB variable sysLocation
SYSLOCATION
   DHAT SYSTEM ROOM BLDG 901
ENDSYSLOCATION
; Inform the following users of serious errors
INFORM
    OPERATOR TCPMAINT
ENDINFORM
; Obey the following users for restricted commands
OBEY
    OPERATOR TCPMAINT VMFTP1
ENDOBEY
; Autolog the following server machines
AUTOLOG
    FTPSRV23 FIREWRK     ; FTP SERVER
;   DSSERV   FIREWRK     ; WDSF Server
    LPSRV23  FIREWRK     ; LP SERVER

                                                    MORE...
```

Figure 77. TCP/IP Configuration, continued

```
;   NAMESRV  FIREWRK     ; DOMAIN NAME SERVER
    PORTMP23 FIREWRK     ; PORTMAP SERVER
;   REXECD   FIREWRK     ; REXEC SERVER
;   ROUTED   FIREWRK     ; ROUTED SERVER
    SMTP23   FIREWRK     ; SMTP SERVER
;   SNMPD    FIREWRK     ; SNMP VM AGENT VIRTUAL MACHINE
;   SNMPQE   FIREWRK     ; SNMP VM CLIENT VIRTUAL MACHINE
    VMNFS23  FIREWRK     ; NFS SERVER
ENDAUTOLOG

PORT
    20 TCP FTPSERVE  NOAUTOLOG ; FTP SERVER
    21 TCP FTPSERVE            ; FTP SERVER
    23 TCP INTCLIEN            ; TELNET SERVER
    25 TCP SMTP23              ; SMTP SERVER
;   53 TCP NAMESRV             ; DOMAIN NAME SERVER
;   53 UDP NAMESRV             ; DOMAIN NAME SERVER
   111 TCP PORTMP23            ; PORTMAP SERVER
   111 UDP PORTMP23            ; PORTMAP SERVER
   515 TCP LPSRV23             ; LP SERVER
  2049 UDP VMNFS23             ; NFS SERVER

device dhatblk claw 410 AUSBLK DHATBLKM NONE 26 26 4096 4096
LINK ra1 IP 0 dhatblk

device dhat  claw A50 AUSESC DHATCHAN none 26 26 4096 4096
LINK ra2 IP 0 dhat

device poly1 claw 900 DHAT9221 POLY   none 26 26 4096 4096
link poly ip 0 poly1

device laser claw 406 HOST PSCA none 26 26 4096 4096
link chan1 ip 0 laser

; the local host's Internet addresses
HOME
   10.0.0.2      chan1
   16.0.0.2      poly
   13.0.0.2      ra1
   14.0.0.2      ra2

; ROUTING INFORMATION (IF YOU ARE NOT USING THE ROUTED SERVER)
 GATEWAY
;  Network      First hop     Driver  Packet size  Subn mask  Subn value
;  1             =             tcpip   DEFAULTSIZE  0.255.255.128 0.0.0.0
;  3             =             pvttcp  DEFAULTSIZE  0.255.255.128 0.0.0.0
  10            =             chan1    4096        0
  16            =             poly     4096        0
  13            =             ra1      4096        0
  14            =             ra2      4096        0
; 9             =             TR1     DEFAULTSIZE  0.255.255.128 0.3.13.0
; 129           =             TR1     DEFAULTSIZE    0
;   9.3.13.30   =             TR1     DEFAULTSIZE     HOST
; ; RouteD Routing information (if you are using the ROUTED server)
; ; If you are using RouteD, uncomment all the lines below for
; ; 'BSDROUTINGPARMS', and comment out all the lines for the 'GATEWAY'
; ; statement.

                                                        MORE...
```

Figure 78. TCP/IP Configuration, continued

```
; ;  link     maxmtu   metric   subnet mask     dest addr
; BSDROUTINGPARMS FALSE
;   TR1    DEFAULTSIZE   0     255.255.255.128  0
; ENDBSDROUTINGPARMS
;
;
;
; TRANSLATE
; The following translate statements are used for remote Hyperchannel
; hosts.
; 193.6.0.1  HCH     FF0000001040     HCH1
;
;
;
; Start all the interface
  start dhatblk
  start dhat
  start poly1
  start laser
```

Figure 79. TCP/IP Configuration

B.1.2 PROFILE EXEC for TCPIP Virtual Machine

This is the profile exec for the TCPIP Virtual Machine.

```
/**********************************************************************/
/* Profile exec for the TCPIP Virtual Machine.                        */
/**********************************************************************/
/* Initialize common variables.                                       */
/**********************************************************************/
userexit = 'TCPIPXIT'                  /* Name of user exit EXEC.     */
serverid = 'TCPIP'                     /* TCPRUN routine for this srvr.*/
tcpexit = '*IGNORE*'                   /* Assume no user exit exists.  */
exittype = 'PRELUDE'                   /* User exit argument from here.*/
tcprun = 'TCPRUN'                      /* Name of exec to invoke server*/
owner = 'TCPMAINT'                     /* Assign default ownerid.     */
command = ''                           /* Let TCPRUN or user exit     */
parms = ''                             /* handle these assignments.   */
invoke_tcprun = 1                      /* Hope for the best.          */
stay_online = 0                        /* Assume the worst.           */
globalv = 'globalv SELECT TCPRUN'      /* Nickname for readability.   */


/**********************************************************************/
/* Set up specific to this server before calling TCPRUN.              */
/**********************************************************************/
'ATTACH A50 *'
'ATTACH A51 *'
'ATTACH 410 *'
'ATTACH 411 *'
                                                          MORE...
```

Figure 80. Profile Exec for TCPIP Virtual Machine, Continued

```
/* ---------------------------------------------------------------- */
/*    Next line has been modified for APAR PN15969                  */
/* ---------------------------------------------------------------- */
Call Set_Up_Cons owner                /* Initiate console spooling    */
'Access 198 D'                        /* Access configuration files.  */
'Access 591 E'                        /* Access the server modules.   */
'Access 592 F'                        /* Access the TCP/IP TXTLIBs.   */
/* ---------------------------------------------------------------- */
/*    Next 12 lines have been added for APAR PN19884                */
/* ---------------------------------------------------------------- */
/* ---------------------------------------------------------------- */
/* Note: Ensure that Virtual Machine can gain Write access to it's  */
/*       191 disk.                                                  */
/* ---------------------------------------------------------------- */
mprefix = 'TCP'                       /* Set Msg Prefix passed        */
'EXEC TCPDSKCK' mprefix owner         /* Check to see if write ok     */
If rc <> 0 Then                       /* Do we have 191 disk R/W?     */
  Do                                  /* No, cease & desist with init */
    saverc = rc                       /* Save retcode from TCPDSKCK   */
    'EXECIO 0 CP (STRING SP CONS STOP CLOSE' /* Close the console     */
    Exit saverc                       /* Exit with return code        */
  End                                 /* End - cease & desist w/ init */

/*********************************************************************/
/* Call user exit if it exists.                                      */
/*********************************************************************/
globalv 'PURGE'                       /* Clean up from the last time. */
globalv 'PUT OWNER COMMAND PARMS'     /* Initialize for user exit.    */

exit_exists = filehere( userexit ,    /* Does a user exit exist?      */
                        'EXEC *' )
if exit_exists then                   /* Is there a user exit?        */
  do                                  /* Yes,                         */
    tcpexit = userexit                /* Pass it's name to TCPRUN.    */
    command = 'exec' userexit exittype /* Here's the user exit command.*/
    say 'Issuing command "'command'"...'/*Tell user what we're doing.  */
    command                           /* Call exit.                   */
    invoke_tcprun = rc = 0            /* By convention, this means go.*/
    stay_online = rc = 4              /* This means don't logoff.     */
  end /* do if exit_exists                                           */

                                                          MORE...
```

Figure 81. Profile Exec for TCPIP Virtual Machine, Continued

```
/*********************************************************************/
/* Do the right thing.                                               */
/*********************************************************************/

select                                  /* What did the user want?      */
  when invoke_tcprun then               /* Did user exit return 0?      */
    do                                  /* Yes.                         */
      globalv 'GET' ,                   /* Get user's command, parms.   */
        'COMMAND PARMS OWNER'
      cmd = 'exec' tcprun serverid ,    /* Set up for this server.      */
        '/exit' tcpexit ,
        '/ownerid' owner ,
        '/command' command ,
        '/parms' parms
      say 'Stacking this command for execution:'
      say ''
      say '  ' cmd
      say ''
      queue cmd                         /* Queue tcprun for invocation. */
    end /* do to invoke tcprun.                                        */
  when stay_online then                 /* User want to stay logged on? */
    say 'Terminating server startup',   /* Issue appropriate message,   */
    'at the request of' userexit'.'     /* then leave.                  */
  otherwise                             /* Non-zero rc from user exit.  */
    say 'Aborting server startup;',     /* Tell the console.            */
      'return code' rc 'from' userexit'.'
    'execio 0 cp (string LOGOFF'        /* Logoff.                      */
end /* select                                                          */

/* ---------------------------------------------------------------- */
/*    Next 2 lines have been added for APAR PN15969                 */
/* ---------------------------------------------------------------- */
If ¬ invoke_tcprun Then                 /* Are we supposed to continue? */
  'EXECIO 0 CP (STRING SP CONS STOP CLOSE' /* No, close the console    */

exit rc                                 /* Get outta dodge.             */

/*********************************************************************/
/* FileHere returns TRUE (1) if the given file exists.               */
/*********************************************************************/
filehere: procedure
parse upper arg fileid                  /* Get the file in question.    */
typeflag = cmstype( 'HT' )              /* Don't let STATE give answer. */
'State' fileid                          /* Does the file exist?         */
found = (rc = 0)                        /* Return TRUE or FALSE answer. */
'Set CMSTYPE' typeflag                  /* Return to previous setting.  */
return found                            /* Return result.               */

                                                          MORE...
```

Figure 82. Profile Exec for TCPIP Virtual Machine, Continued

```
/*********************************************************************/
/* Set CMSTYPE to new value if given, then return the old setting.   */
/*********************************************************************/
cmstype: procedure
parse arg newflag                       /* Get new CMSTYPE setting.  */
'MakeBuf'                               /* Get a program buffer.     */
buffer = rc                             /* Save buffer identifier.   */
'Query CMSType (LIFO'                   /* Query the old setting.    */
parse pull . '=' oldflag .              /* Retrive old setting.      */
'DropBuf' buffer                        /* Get rid of program buffer.*/
if newflag <> '' then                   /* Was new value given?      */
  'Set CMSType' newflag                 /* Yes set CMSTYPE to new value.*/
return oldflag                          /* Return old value.         */


/* ----------------------------------------------------------------- */
/*    Next 17 lines have been added for APAR PN15969                 */
/* ----------------------------------------------------------------- */
Set_Up_Cons: Procedure
/* ----------------------------------------------------------------- */
/*    The routine Set_Up_Cons is responsible for performing the      */
/*    initial console spooling. It will attempt to spool the con-    */
/*    sole to the userID TCPMAINT provided that it is a valid        */
/*    userID for the system. If not, it will merely spool the        */
/*    console back to itself.                                        */
/* ----------------------------------------------------------------- */
Parse Arg dflt_owner                    /* Get the default owner ID  */
uid = dflt_owner                        /* Make local copy of argument  */
'EXECIO 0 CP (STRING LINK' uid          /* Issue a bad LINK command  */
saverc = rc - 1000                      /* Generate retcode from LINK   */
If saverc <> 22 Then                    /* Did we get a syntax error?   */
  uid = '*'                             /* No, set uid as ourselves  */
'EXECIO 0 CP (STRING SPOOL CONS STOP CLOSE'  /* In case we have re-IPL */
'EXECIO 0 CP (STRING SPOOL CONS' uid 'START' /* Start console spooling */
Return                                  /* Return to mainline code   */
```

Figure 83. Profile Exec for TCPIP Virtual Machine

B.1.3 TCPIP DATA File

This file, TCPIP DATA, is used to specify configuration information required by TCP/IP client programs. This it the one used in Austin.

```
;
;*********************************************************************
;                                                                    *
;   Name of File:           TCPIP DATA                               *
;                                                                    *
;   This file, TCPIP DATA, is used to specify configuration          *
;   information required by TCP/IP client programs.                  *
;                                                                    *
;                                                                    *
;   Syntax Rules for the TCPIP DATA configuration file:              *
;                                                                    *
;   (a) All characters to the right of and including a ';' will be   *
;       treated as a comment.                                        *
;                                                                    *
;   (b) Blanks and <end-of-line> are used to delimit tokens.         *
;                                                                    *
;   (c) The format for each configuration statement is:              *
;                                                                    *
;       <SystemName||':'>  keyword  value                            *
;                                                                    *
;       where <SystemName||':'> is an optional label which may be    *
;       specified before a keyword; if present, then the keyword-    *
;       value pair will only be recognized if the SystemName matches *
;       the node name of the system, determined by the CMS IDENTIFY  *
;       command.  This optional label permits configuration          *
;       information for multiple systems to be specified in a single *
;       TCPIP DATA file.                                             *
;                                                                    *
;*********************************************************************
; TCPIPUSERID specifies the userid of the TCP/IP Virtual Machine.
; TCPIP is the default userid.
;
TCPIPUSERID TCPIP
;
; HOSTNAME specifies the TCP host name of this VM host.  If not
; specified, the default HOSTNAME will be the node name returned
; by the CMS IDENTIFY command.
;
; For example, if this TCPIP DATA file is shared between two systems,
; OURVM and YOURVM, then the following two lines will define the
; HOSTNAME correctly on each system.
;
HOSTNAME  AUSESC
;
; DOMAINORIGIN specifies the domain origin that will be appended
; to host names passed to the resolver.  If a host name contains
; any dots, then the DOMAINORIGIN will not be appended to the
; host name.
;
                                                          MORE...
```

Figure 84. TCPIP DATA File, continued

```
DOMAINORIGIN  AUSTIN.IBM.COM
;
; NSINTERADDR specifies the Internet address of the name server.
; LOOPBACK (14.0.0.0) is the default value (your local name server).
; If a name server will not be used, then do not code an NSINTERADDR
; statement (Comment out the NSINTERADDR line below).  This will cause
; all names to be resolved via site table lookup.
;
NSINTERADDR 13.0.0.1
NSINTERADDR 17.0.0.1
NSINTERADDR  14.0.0.0
;
; NSPORTADDR specifies the foreign port of the Name Server.
; 53 is the default value.
;
NSPORTADDR 53
;
; RESOLVEVIA specifies how the Resolver is to communicate with the
; name server.  TCP indicates use of TCP virtual circuits.  UDP
; indicates use of UDP datagrams.  The default is UDP.
;
RESOLVEVIA UDP
;
; RESOLVERTIMEOUT specifies the time in seconds that the Resolver
; will wait to complete an open to the name server (either UDP or TCP).
; The default is 30 seconds.
;
RESOLVERTIMEOUT 10
;
; RESOLVERUDPRETRIES specifies the number of times the resolver
; should try to connect to the name server when using UDP datagrams.
; The default is 1.
;
RESOLVERUDPRETRIES 1
;
; TRACE RESOLVER will cause a complete trace of all queries to and
; responses from the name server or site tables to be written to
; the user's console.  This command is for debugging purposes only.
;
; TRACE RESOLVER
;
;
;*********************************************************************
;
; End of file.
;
```

Figure 85. TCPIP DATA File

B.1.4 HOSTS LOCAL File

This is the HOSTS LOCAL file used in Austin.

```
; HOSTS LOCAL
; -----------
;
; The format of this file is documented in RFC 952, "DoD Internet
; Host Table Specification".
;
; The format for entries is:
;
; NET : ADDR : NETNAME :
; GATEWAY : ADDR, ALT-ADDR : HOSTNAME : CPUTYPE : OPSYS : PROTOCOLS :
; HOST : ADDR, ALT-ADDR : HOSTNAME, NICKNAME : CPUTYPE : OPSYS : PROTOCOLS :
;
; Where:
;   ADDR, ALT-ADDR = Internet address in decimal, e.g., 26.0.0.73
;   HOSTNAME, NICKNAME = the fully qualified hostname and any nicknames
;   CPUTYPE = machine type (PDP-11/70, VAX-11/780, IBM-3090, C/30, etc.)
;   OPSYS = operating system (UNIX, TOPS20, TENEX, VM/SP, etc.)
;   PROTOCOLS = transport/service (TCP/TELNET,TCP/FTP, etc.)
;   : (colon) = field delimiter
;   :: (2 colons) = null field
; *** CPUTYPE, OPSYS, and PROTOCOLS are optional fields.
;
;
; Note:  The NET and GATEWAY statements are not used by the TCP/IP for
;        VM applications.  However, some socket calls require the NET
;        entries.  For added performance, if your programs do not need
;        the NET and GATEWAY statements, delete them before running
;        the MAKESITE program.
;
;
 HOST : 13.0.0.1   : DHATBLKM ::::
 HOST : 13.0.0.2   : AUSBLK   ::::
 HOST : 14.0.0.1   : DHATCHAN ::::
 HOST : 14.0.0.2   : AUSESC   ::::
 HOST : 9.3.13.131 : AUSHAT2  ::::
;
```

Figure 86. HOSTS LOCAL File

Appendix C. VM SNA Configuration Files

These are the SNA configurations used during the VM part of the Austin residency.

C.1.1 PROFILE GCS File

This is the PROFILE GCS file.

```
/********************************************************************/
'CP DEF GRAF 600'
'CP DEF GRAF 601'
'CP DEF GRAF 602'
'CP DEF GRAF 603'
'CP DEF GRAF 604'
'CP DEF GRAF 605'

'CP DEF GRAF 606'
'CP DEF GRAF 607'
'CP DEF GRAF 608'
'CP DEF GRAF 609'
'CP DEF GRAF 60A'
'CP DEF GRAF 60B'
'CP DEF GRAF 60C'
'CP DEF GRAF 60D'
'CP DEF GRAF 60E'
'CP DEF GRAF 60F'
'CP SET PF12 RETRIEVE'
/* Att addr for H3745B */
/*'ATT 521 *' */
/* Att addr for H3720C */
/*'ATT 522 *' */
/*'ATT 610-67F *'*/
/*
***** SET UP THE VIRTUAL CTC'S
***** FOR MULTIPLE GUESTS
*/
/*
'DEFINE CTCA 317 HATMVS4'
'COUPLE 317 TO HATMVS4 317'

'DEFINE CTCA 31D HATMVS3'
'COUPLE 31D TO HATMVS3 31D'

'DEFINE CTCA 334 HATVSE'
'COUPLE 334 TO HATVSE 334'
*/
/**
                                                          MORE...
```

Figure 87. PROFILE CGS, Continued

```
*** Title-
***      VMVTAM
***
*** Function-
***      Initialize VM/VTAM and VSCS for use.
***
*** Parameters-
***      list_value
***
*** Returns-
***      00 (VTAM has been successfully activated)
***      ¬0 (VTAM activation failed)
**/
    parse source . . exec_name
    arg list_value . '(' options
    if list_value = '' then
      LIST_VALUE='11'

/**
*** Set CP options to improve performance of VTAM Virtual Machine
**/
    'CP ENABLE SNA'                 /* ENABLE SNA COMMUNICATIONS     */

/**
*** VTAM initialization
**/
    'ACC 29A F/F'                   /* VTAM run disk                 */
/*  'ACC 543 L/L' */                /* SSP 4.1 RUN DISK              */
/*  'ACC 034 N/N' */                /* NETVIEW RUN DISK              */
/*  'FILEDEF NCPLOAD DISK NCP LOADLIB *' */
    'GLOBAL LOADLIB VTAMUSER VTAM VSCS'
    'LOADCMD VTAM ISTINV00'
    'LOADCMD VSCS DTISLCMD'
    'VTAM START LIST=11'
    rcode=rc
    if rcode¬=0 then                /* If VTAM start failure         */
      do                            /* Error, VTAM startup failed    */
        say '**ERROR** VTAM initialization failed'
        exit rcode
      end                           /* Error, VTAM startup failed    */
/**
*** VSCS initialization
**/
    'VSCS START'/* INITIALIZE VSCS              */
    RCODE=RC                        /* SAVE STARTUP RETURN CODE      */
    if rcode¬=0 then                /* If VTAM start failure         */
      do                            /* Error, VTAM startup failed    */
        say '**ERROR** VSCS initialization failed'
        exit rcode
      end                           /* Error, VTAM startup failed    */
    exit 0
    exit 0
```

Figure 88. PROFILE GCS File

C.1.2 VTAMAINT DIRECT File

This is the VTAMAINT DIRECT file.

```
USER VTAMAINT T1MEOUT 12M 32M BEG                                       VTA00010
*@VTAMAINT  OWNER:<SYSTEM>,<406084>                                     VTA00020
MACHINE XA                                                              VTA00030
 ACCOUNT 8000                                                           VTA00040
 IPL CMS PARM NOSPROF                                                   VTA00050
 CONSOLE 009 3215                                                       VTA00060
 SPOOL 00C 2540 READER *                                                VTA00070
 SPOOL 00D 2540 PUNCH  A                                                VTA00080
 SPOOL 00E 1403 A                                                       VTA00090
 LINK MAINT 190 190 RR                                                  VTA00100
 LINK MAINT 19E 19E RR                                                  VTA00110
*                                                                       VTA00120
MDISK 114 3380 11  002 VMESC3 RR                                        VTA00130
MDISK 298 3380 13  010 VMESC3 MR                                        VTA00140
MDISK 049A 3380 23  30 VMESC3 MR                                        VTA00150
```

Figure 89. VTAMAINT DIRECT

C.1.3 USSHATS ASSEMBLE File

This is the USSHATS ASSEMBLE file.

```
USSHATS  USSTAB TABLE=STDTRANS,FORMAT=V3R2                              USS00010
*********************************************************************   USS00090
*  STANDARD ENTRY TO SUPPORT FORMATTED LOGONS                           USS00100
*********************************************************************   USS00110
LOGON    USSCMD   CMD=LOGON,FORMAT=PL1                                  USS00120
         USSPARM  PARM=APPLID                                           USS00130
         USSPARM  PARM=LOGMODE                                          USS00140
         USSPARM  PARM=DATA                                             USS00150
*********************************************************************   USS00160
*  STANDARD ENTRY TO SUPPORT LOGOFF                                     USS00170
*********************************************************************   USS00180
LOGOFF   USSCMD   CMD=LOGOFF,FORMAT=PL1                                 USS00190
         USSPARM  PARM=APPLID                                           USS00200
         USSPARM  PARM=TYPE,DEFAULT=UNCOND                              USS00210
         USSPARM  PARM=HOLD,DEFAULT=YES                                 USS00220
*********************************************************************   USS00230
*  STANDARD ENTRY TO SUPPORT IBMTEST                                    USS00240
*********************************************************************   USS00250
IBMTEST  USSCMD   CMD=IBMTEST,FORMAT=BAL                                USS00260
         USSPARM  PARM=P1,DEFAULT=10                                    USS00270
         USSPARM  PARM=P2,DEFAULT=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  USS00280
*********************************************************************   USS00290
*                                                                       USS00300
*  END OF VTAM SUPPLIED USS COMMANDS                                    USS00310
*  THE FOLLOWING ENTRIES ARE APPLICATION INDEPENDENT COMMANDS FOR       USS00320
*  SPECIAL FUNCTIONS                                                    USS00330
*                                                                       USS00340

                                                          MORE...
```

Figure 90. USSHATS ASSEMBLE, Continued

```
*********************************************************************  USS00350
*   SPECIAL  ENTRY TO SUPPORT LOGOFF TYPE=COND                        USS00360
*********************************************************************  USS00370
DISC     USSCMD  CMD=DISC,FORMAT=BAL,REP=LOGOFF                        USS00380
         USSPARM PARM=APPLID                                           USS00390
         USSPARM PARM=TYPE,DEFAULT=COND                                USS00400
         USSPARM PARM=HOLD,DEFAULT=YES                                 USS00410
*********************************************************************  USS00420
*   SPECIAL ENTRY TO SUPPORT IBMTEST  (ECHO TEST)                     USS00430
*********************************************************************  USS00440
ECHO     USSCMD  CMD=ECHO,REP=IBMTEST,FORMAT=BAL                       USS00450
         USSPARM PARM=P1,DEFAULT=10                                    USS00460
         USSPARM PARM=P2,DEFAULT=ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  USS00470
*********************************************************************  USS00480
*   SPECIAL ENTRY TO PROVIDE TERMINAL ID                              USS00490
*   REQUIRES USSMSG13 TO HAVE @@LUNAME IN THE MESSAGE                 USS00500
*********************************************************************  USS00510
ID       USSCMD  CMD=ID,REP=IBMTEST,FORMAT=BAL                         USS00520
         USSPARM PARM=P1,DEFAULT=1                                     USS00530
         USSPARM PARM=P2,DEFAULT='FOR TERMINAL NODE NAME CHECK'        USS00540
*********************************************************************  USS00550
VMEXIT   USSCMD  CMD=VMEXIT,REP=IBMTEST,FORMAT=BAL                     USS00560
         USSPARM PARM=COUNT,REP=P1,DEFAULT=1                           USS00570
         USSPARM PARM=MSG,REP=P2,                                     XUSS00580
               DEFAULT='.  VMEXIT ONLY DEFINED ON VM SCREEN.'          USS00590
*********************************************************************  USS00600
*   SPECIAL  ENTRY TO SUPPORT LOGOFF FORCE                            USS00610
*********************************************************************  USS00620
FORCE    USSCMD  CMD=FORCE,REP=LOGOFF,FORMAT=PL1                       USS00630
         USSPARM PARM=APPLID                                           USS00640
         USSPARM PARM=TYPE,DEFAULT=FORCE                               USS00650
         USSPARM PARM=HOLD                                             USS00660
*********************************************************************  USS00670
*   SPECIAL ENTRY TO SUPPORT SHORTHAND FORMATTED LOGONS               USS00680
*********************************************************************  USS00690
VM       USSCMD  CMD=LERR,REP=LOGON,FORMAT=BAL USS MSG 12 TEST         USS00700
L        USSCMD  CMD=L,REP=LOGON,FORMAT=BAL                            USS00710
         USSPARM REP=APPLID,PARM=P1                     1ST POSTIONAL  USS00720
         USSPARM REP=APPLID,PARM=A                                     USS00730
         USSPARM REP=APPLID,PARM=APPLID                                USS00740
         USSPARM REP=DATA,PARM=P2                                      USS00750
         USSPARM REP=DATA,PARM=DATA                                    USS00760
         USSPARM REP=DATA,PARM=D                                       USS00770
         USSPARM REP=LOGMODE,PARM=P3                                   USS00780
         USSPARM REP=LOGMODE,PARM=LOGMODE                              USS00790
         USSPARM REP=LOGMODE,PARM=MODE                                 USS00800
*********************************************************************  USS00810
*   SPECIAL ENTRY TO SUPPORT SHORTHAND FORMATTED LOGOFF               USS00820
*********************************************************************  USS00830
LO       USSCMD  CMD=LO,REP=LOGOFF,FORMAT=BAL                          USS00840
         USSPARM PARM=APPLID                                           USS00850
         USSPARM PARM=TYPE,DEFAULT=UNCOND                              USS00860
         USSPARM PARM=HOLD,DEFAULT=YES                                 USS00870
*    NETVIEW ON AUSVHAT4 VM SYSTEM                                     USS01220
*                                                                      USS01230
                                                        MORE...
```

Figure 91. USSHATS ASSEMBLE, Continued

```
NV1      USSCMD CMD=NV1,REP=LOGON,FORMAT=BAL                              USS01240
         USSPARM PARM=APPLID,DEFAULT=TH1NV                                USS01250
         USSPARM REP=DATA,PARM=DATA           NOT USED BY NETVIEW         USS01260
         USSPARM REP=LOGMODE,PARM=P1                                      USS01270
         USSPARM REP=LOGMODE,PARM=P2                                      USS01280
         USSPARM REP=LOGMODE,PARM=MODE                                    USS01290
         USSPARM REP=LOGMODE,PARM=LOGMODE                                 USS01300
************************************************************            USS01310
*                                                                         USS01320
*    ENTRIES FOR NETWORKS OUTSIDE DHAT                                    USS01330
*                                                                         USS01340
*                                                                         USS01350
AUSNET   USSCMD  CMD=AUSNET,REP=LOGON,FORMAT=BAL                          USS01360
         USSPARM PARM=APPLID,REP=APPLID,DEFAULT=NRASAM1                   USS01370
         USSPARM REP=DATA,PARM=P1                                         USS01380
         USSPARM REP=DATA,PARM=DATA                                       USS01390
         USSPARM REP=DATA,PARM=NODE                                       USS01400
         USSPARM REP=LOGMODE,PARM=P2                                      USS01410
         USSPARM REP=LOGMODE,PARM=MODE                                    USS01420
         USSPARM REP=LOGMODE,PARM=LOGMODE                                 USS01430
*                                                                         USS01440
AUSNET2  USSCMD  CMD=AUSNET,REP=LOGON,FORMAT=BAL                          USS01450
         USSPARM PARM=APPLID,REP=APPLID,DEFAULT=NRASAM2                   USS01460
         USSPARM REP=DATA,PARM=P1                                         USS01470
         USSPARM REP=DATA,PARM=DATA                                       USS01480
         USSPARM REP=DATA,PARM=NODE                                       USS01490
         USSPARM REP=LOGMODE,PARM=P2                                      USS01500
         USSPARM REP=LOGMODE,PARM=MODE                                    USS01510
         USSPARM REP=LOGMODE,PARM=LOGMODE                                 USS01520
*                                                                         USS01530
CCDN     USSCMD  CMD=CCDN,REP=LOGON,FORMAT=BAL                            USS01540
         USSPARM PARM=APPLID,REP=APPLID,DEFAULT=CABZMENU                  USS01550
         USSPARM REP=DATA,PARM=P1                                         USS01560
         USSPARM REP=DATA,PARM=NODE                                       USS01570
         USSPARM REP=LOGMODE,PARM=P2                                      USS01580
         USSPARM REP=LOGMODE,PARM=MODE                                    USS01590
         USSPARM REP=LOGMODE,PARM=LOGMODE                                 USS01600
*                                                                         USS01610
AUSPUB2  USSCMD  CMD=AUSPUB2,REP=LOGON,FORMAT=BAL                         USS01620
         USSPARM PARM=APPLID,REP=APPLID,DEFAULT=TH1VSCS                   USS01630
         USSPARM REP=DATA,PARM=P1                                         USS01640
         USSPARM REP=DATA,PARM=NODE                                       USS01650
         USSPARM REP=LOGMODE,PARM=P2                                      USS01660
         USSPARM REP=LOGMODE,PARM=MODE                                    USS01670
         USSPARM REP=LOGMODE,PARM=LOGMODE                                 USS01680
*                                                                         USS01690
AUSVM1   USSCMD  CMD=AUSVM1,REP=LOGON,FORMAT=BAL                          USS01700
         USSPARM PARM=APPLID,REP=APPLID,DEFAULT=AUAVM1                    USS01710
         USSPARM REP=DATA,PARM=P1                                         USS01720
         USSPARM REP=DATA,PARM=NODE                                       USS01730
         USSPARM REP=LOGMODE,PARM=P2                                      USS01740
         USSPARM REP=LOGMODE,PARM=MODE                                    USS01750
         USSPARM REP=LOGMODE,PARM=LOGMODE                                 USS01760
```

Figure 92. USSHATS ASSEMBLE, Continued

```
*        DHAT menu and logo for SNA 3270 terminals                      USS01900
*        The buffer form is used, thus precluding any luname or         USS01910
*        message parameter substitution by VTAM                         USS01920
*                                                                       USS01930
         USSMSG  MSG=5,BUFFER=MSG10                                     USS01940
         USSMSG  MSG=10,BUFFER=MSG10                                    USS01950
*                                                                       USS01960
MSG10    DS    0F                  ALIGN ON FULL WORD                  USS01970
         DC    AL2(MSG10E-MSG10S)  LENGTH OF MSG10                     USS01980
MSG10S   EQU   *                                                       USS01990
         DC    X'15'                NL To line 2                       USS02000
         DC    CL12' '                                                 USS02010
         DC    C'DEVELOPMENT HOST ATTACH TEST - AUSTIN, TEXAS, USA'    USS02020
         DC    X'15'                NL To line 3                       USS02030
         DC    X'15'                NL To line 4                       USS02040
         DC    X'15'                NL To line 5                       USS02050
         DC    CL33' '                                                 USS02060
         DC    C'"""""    "   "   """"   """""""'                      USS02070
         DC    X'15'                NL To line 6                       USS02080
         DC    CL15' '                                                 USS02090
         DC    C'*****          "   "  "   "  "   "   "'              USS02100
         DC    X'15'                NL To line 7                       USS02110
         DC    CL15' '                                                 USS02120
         DC    C'*****         "   "  """""""  """""""   "'           USS02130
         DC    X'15'                NL To line 8                       USS02140
         DC    CL15' '                                                 USS02150
         DC    C'**********   "   "  "   "  "   "   "'               USS02160
         DC    X'15'                NL To line 9                       USS02170
         DC    CL10' '                                                 USS02180
         DC    C'**********  """""""  "   "  "   "   "'               USS02190
         DC    X'15'                NL To line 10                      USS02200
         DC    CL11' '                                                 USS02210
         DC    C'**********'                                           USS02220
         DC    X'15'                NL To line 11                      USS02230
         DC    CL13' '                                                 USS02240
         DC    C'**********    DHAT Information Line:   3-6300'        USS02250
         DC    X'15'                NL To line 12                      USS02260
         DC    CL14' '                                                 USS02270
         DC    C'**  ******                (Tie Line: 793-6300)'       USS02280
         DC    X'15'                NL To line 13                      USS02290
         DC    CL19' '                                                 USS02300
         DC    C'****'                                                 USS02310
         DC    X'15'                NL To line 14                      USS02320
         DC    CL20' '                                                 USS02330
         DC    C'***'                                                  USS02340
         DC    X'15'                NL To line 15                      USS02350
         DC    X'15'                NL To line 16                      USS02360
         DC    X'15'                NL To line 17                      USS02370
         DC    CL13' '                                                 USS02380
         DC    C'AUSHAT1'                                              USS02390
         DC    X'15'                NL To line 18                      USS02400
         DC    X'15'                NL To line 19                      USS02410
         DC    X'15'                NL To line 20                      USS02420
         DC    CL13' '                                                 USS02430
         DC    C'Valid access codes: Hat1  Ausvm1  Ausnet'             USS02440
         DC    X'15'                NL To line 21                      USS02450
         DC    X'15'                NL To line 22                      USS02460
         DC    C'Enter access code:  '                                 USS02470
               C'Please stand by.                              ')      USS02720
                                                             MORE...
```

Figure 93. USSHATS ASSEMBLE, Continued

```
MSG1OE   EQU   *                                                      USS02480
* VTAM   USS message 0 not supplied with VTAM                         USS02680
         USSMSG   MSG=0,TEXT=(X'4015',                               *USS02690
               C'% Command from node @@LUNAME is in progress',       *USS02700
               X'4015',                                              *USS02710
*                                                                     USS02730
* VTAM   USSMSG   MSG=1,TEXT=' INVALID COMMAND SYNTAX'               USS02740
         USSMSG   MSG=1,TEXT=(X'4015',                               *USS02750
               C' Invalid command syntax for terminal @@LUNAME       *USS02760
                 (USSMSG1)',                                         *USS02770
               X'4015',                                              *USS02780
               C' Press CLEAR then ENTER key for application menu')  USS02790
*                                                                     USS02800
* VTAM   USSMSG   MSG=2,TEXT='% COMMAND UNRECOGNIZED'                 USS02810
         USSMSG   MSG=2,TEXT=(X'4015',                               *USS02820
               C' Command "%" is not defined for @@LUNAME    (USSMSG2)',*USS02830
               X'4015',                                              *USS02840
               C' Press CLEAR then ENTER key for application menu')  USS02850
*                                                                     USS02860
* VTAM   USSMSG   MSG=3,TEXT='% PARAMETER UNRECOGNIZED'               USS02870
         USSMSG   MSG=3,TEXT=(X'4015',                               *USS02880
               C' Parameter "%" is not defined for command entered',*USS02890
               C'     (USSMSG3)',                                    *USS02900
               X'4015',                                              *USS02910
               C' Press CLEAR then ENTER key for application menu')  USS02920
*                                                                     USS02930
* VTAM   USSMSG   MSG=4,TEXT='% PARAMETER INVALID'                    USS02940
         USSMSG   MSG=4,TEXT=(X'4015',C' Data for ',                 *USS02950
               C' Parameter "%" is invalid or missing.     (USSMSG4)',*USS02960
               X'4015',                                              *USS02970
               C' Press CLEAR then ENTER key for application menu')  USS02980
*                                                                     USS02990
* VTAM   USSMSG   MSG=5,TEXT=' UNSUPPORTED FUNCTION'                  USS03100
*        USSMSG   MSG=5,TEXT=(X'4015',                                USS03110
*              C' @@LUNAME is active to VTAM.         (USSMSG5) ',   USS03120
*              X'4015',                                              USS03130
*              C' Enter application access code or logon command.')  USS03140
*                                                                     USS03150
* VTAM   USSMSG   MSG=6,TEXT=' SEQUENCE ERROR'                        USS03160
         USSMSG   MSG=6,TEXT=(X'15',                                 *USS03170
               C' Sequence error for @@LUNAME   (USSMSG6)',          *USS03180
               X'1515',                                              *USS03190
               C' You are attempting to log off from a terminal that is *USS03200
               not in session,',                                     *USS03210
               X'15',                                                *USS03220
               C' or log on from a terminal already in session.',    *USS03230
               X'1515',                                              *USS03240
               C' Press CLEAR then ENTER key for application menu')  USS03250
*                                                                     USS03260
                                                          MORE...
```

Figure 94. USSHATS ASSEMBLE, Continued

```
* VTAM   USSMSG   MSG=7,TEXT='%(1) UNABLE TO ESTABLISH SESSION - %(2) F USS03270
* VTAM         AILED WITH SENSE %(3)'                      /* @R495104*/ USS03280
         USSMSG  MSG=7,TEXT=(X'4015',                                   *USS03290
               C'%(1) Unable to establish session.',                    *USS03300
               X'4015',                                                 *USS03310
               C'%(2) Failed with sense %(3)         (USSMSG7)',        *USS03320
               X'4015',                                                 *USS03330
               C' Press CLEAR then ENTER key for application menu')      USS03340
*                                                                        USS03350
* VTAM   USSMSG   MSG=8,TEXT=' INSUFFICIENT STORAGE'                     USS03360
         USSMSG   MSG=8,TEXT=(X'4015',                                  *USS03370
               C' VTAM  has a temporary storage shortage - PLEASE ',    *USS03380
               C' Wait 5 minutes and retry',                            *USS03390
               X'4015',                                                 *USS03400
               C' Press CLEAR then ENTER key for application menu')      USS03410
*                                                                        USS03420
* VTAM   USSMSG   MSG=9,TEXT=' MAGNETIC CARD DATA ERROR'                 USS03430
         USSMSG   MSG=9,TEXT=' Magnetic card data error (USSMSG9)'      USS03440
* VTAM   USSMSG   MSG=11,TEXT='% SESSIONS ENDED'                        USS03610
         USSMSG   MSG=11,TEXT=(X'4015',                                 *USS03620
               C'% Session complete        (USSMSG11)',               *USS03630
               X'4015',                                                 *USS03640
               C' Press CLEAR then ENTER key for application menu')      USS03650
*                                                                        USS03660
* VTAM   USSMSG   MSG=12,TEXT=' REQUIRED PARAMETER OMITTED'             USS03670
         USSMSG   MSG=12,TEXT=(X'4015',                                 *USS03680
               C'% Required parameter omitted     (USSMSG12)',         *USS03690
               X'4015',                                                 *USS03700
               C' Press CLEAR then ENTER key for application menu')      USS03710
*                                                                        USS03720
*                                                                        USS03730
* VTAM   USSMSG   MSG=13,TEXT=' IBMECHO % '                             USS03740
         USSMSG   MSG=13,TEXT=(C' IBMecho to @@LUNAME %',               *USS03750
               X'4015',                                                 *USS03760
               C' Press CLEAR then ENTER key for application menu')      USS03770
*                                                                        USS03780
STDTRANS DC      X'000102030440060708090A0B0C0D0E0F'                    USS03790
         DC      X'101112131415161718191A1B1C1D1E1F'                    USS03800
         DC      X'202122232425262728292A2B2C2D2E2F'                    USS03810
         DC      X'303132333435363738393A3B3C3D3E3F'                    USS03820
         DC      X'404142434445464748494A4B4C4D4E4F'                    USS03830
         DC      X'505152535455565758595A5B5C5D5E5F'                    USS03840
         DC      X'606162636465666768696A6B6C6D6E6F'                    USS03850
         DC      X'707172737475767778797A7B7C7D7E7F'                    USS03860
         DC      X'80C1C2C3C4C5C6C7C8C98A8B8C8D8E8F'                    USS03870
         DC      X'90D1D2D3D4D5D6D7D8D99A9B9C9D9E9F'                    USS03880
         DC      X'A0A1E2E3E4E5E6E7E8E9AAABACADAEAF'                    USS03890
         DC      X'B0B1B2B3B4B5B6B7B8B9BABBBCBDBEBF'                    USS03900
         DC      X'C0C1C2C3C4C5C6C7C8C9CACBCCCDCECF'                    USS03910
         DC      X'D0D1D2D3D4D5D6D7D8D9DADBDCDDDEDF'                    USS03920
         DC      X'E0E1E2E3E4E5E6E7E8E9EAEBECEDEEEF'                    USS03930
         DC      X'F0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF'                    USS03940
END      USSEND                                                         USS03950
         END                                                            USS03960
```

Figure 95. USSHATS ASSEMBLE File

C.1.4 MTAWD ASSEMBLE

This is the MTAWD ASSEMBLE file.

```
D3270L2Q MODEENT LOGMODE=D3270L2Q, (24 X 80 , 24 X 80) LARGE PU        X
               FMPROF=X'03',                                            X
               TSPROF=X'03',                                            X
               PRIPROT=X'B1',                                           X
               SECPROT=X'90',                                           X
               COMPROT=X'3080',                                         X
               RUSIZES=X'8989',                                         X
               PSERVIC=X'028000000000185018507F00'
************************************************************************
D3270M0Q MODEENT LOGMODE=D3270M0Q,                                      X
               TYPE=X'01',                                              X
               FMPROF=X'03',                                            X
               TSPROF=X'03',                                            X
               PRIPROT=X'B1',                                           X
               SECPROT=X'90',                                           X
               COMPROT=X'3080',                                         X
               SSNDPAC=X'00',                                           X
               SRCVPAC=X'00',                                           X
               RUSIZES=X'87C7',                                         X
               PSNDPAC=X'00',                                           X
               PSERVIC=X'028000000000000000000300'
```

Figure 96. MTAWD ASSEMBLE File

C.1.5 RSCS DIRECT

This is the RSCS DIRECT file.

```
USER RSCS T1MEOUT 32M 48M BG                                    07171642
* RSCS OWNER:<System>,<607835>                                   07171642
 OPTION ACCT MAXCONN 16 SVMSTAT LANG AMENG SETORIG              07171642
 ACCOUNT 60L                                                    07171642
 MACH ESA                                                       07171642
 IPL GCS PARM AUTOLOG                                           07171642
 NAMESAVE GCS NETVSG00                                          07171642
 CONSOLE 01F 3215                                               07171642
 SPOOL 00C 2540 READER A                                        07171642
 SPOOL 00D 2540 PUNCH  A                                        07171642
 SPOOL 00E 1403 A                                               07171642
 DEDICATE 300 300                                               07171642
 DEDICATE 31C 31C                                               07171642
 DEDICATE 305 305                                               07171642
 DEDICATE 314 314                                               07171642
 DEDICATE 32C 32C                                               07171642
 LINK MAINT 190 190 RR                                          07171642
 LINK MAINT 193 193 RR                                          07171642
 LINK MAINT 19E 19E RR                                          07171642
 LINK P684096E 401 191 RR                                       07171642
* MDISK 0192 3380 4 5 VM4SP1 RR                                 07171642
```

Figure 97. RSCS DIRECT

C.1.6 RSCS CONFIG

This is the RSCS CONFIG file.

```
*************************************************************************
* 4/10/90  JNG   DELETED ALL SNA3270 PRT DEFINITIONS.  CAN BE DEFINED  *
*                VIA MERLIN                                            *
*11/23/93  James Boykin                                                *
*          - Changed Aushat2 to SNANJE                                  *
*          - Changed default ROUTE to AUSHAT2                           *
*          - Added ROUTE for AUSVM1 to AUSHAT2                          *
*12/07/93  James Boykin                                                *
*          - Routed AUSVHAT1 via AUSHAT2
*          - Deleted AUSVHAT1 link
*03/10/94  JAMES GARDNER                                               *
*          - RENAME AUSVHAT4 TO AUSHAT1
*09/06/94  JAMES GARDNER                                               *
*          - ROUTE HP903A & B TO AUSPUB2
*10/04/94  JAMES GARDNER                                               *
*          - ADDED DEST STATMENTS FOR PSF PRINTERS
*10/06/94  JAMES GARDNER                                               *
*          - CHANGED AUSHAT1 TO AUSPUB2 TO CTC LINK
*10/06/94  James Boykin                                                *
*          - Added commented out link definition for HP903A and HP903B
*11/28/94  James Gardner                                               *
*          - changed auspub2 to a notify link to send a note about
*            migration to aushat1
*04/06/95  JAMES BOYKIN                                                *
*          - added destination for vuser1pt
*          - deleted destination for vuser1pt
*08/03/95  HYUK KAHNG                                                  *
*          - ADDED LINK FOR RS/6K ESCON AND BLKMUX
                                                              MORE...
```

Figure 98. RSCS CONFIG File, Continued

```
*************************************************************************
*                                                                       *
* Following are the "rules" for defining an RSCS Configuration File:    *
*                                                                       *
*  (1)  The LOCAL statement MUST be the FIRST valid (non-commented,     *
*       non-blank) statement in the Configuration File.                 *
*                                                                       *
*  (2)  The OPFORM and CHANNELS statements, if present, MUST be         *
*       defined prior to defining ANY LINK entries.                     *
*                                                                       *
*  (3)  AUTH, PARM, and ROUTE statements can be intermixed within LINK  *
*       statements, but each MUST correspond to a previously defined    *
*       LINK.  AUTH statements which define userids which are           *
*       authorized for the entire RSCS system can be defined anywhere   *
*       within the Configuration File after the LOCAL statement.        *
*                                                                       *
*  (4)  PORT statements, if present, MUST follow the CHANNELS           *
*       statement, but can be defined ANYWHERE ELSE within the file.    *
*                                                                       *
*  (5)  TAGS, DUMP, MSGNOH, EXIT, and SAFCLASS statements can be        *
*       defined ANYWHERE within the Configuration File, after the LOCAL*
*       statement.                                                      *
*                                                                       *
*   Note:  This gives more flexibility in defining information about    *
*          LINKs, in that all statements pertaining to a given LINK     *
*          can be grouped with that LINK definition.                    *
*                                                                       *
*          Example:                                                     *
*                       LINK   TEST  NJE                                *
*                       ROUTE  TEST1 TEST                               *
*                       ROUTE  TEST2 TEST                               *
*                       PARM   TEST  BUFF=2000 STREAMS=2 TA=1           *
*                       AUTH   TEST  USER1 TEST1                        *
*                                                                       *
*    Warning:  Configuration File statements and their operands must    *
*              be in UPPER case.  This file must NOT contain sequence   *
*              numbers.                                                 *
*                                                                       *
*************************************************************************

*************************************************************************
*                   RSCS LOCAL NODEID SPECIFICATION                     *
*************************************************************************
*
*          LOCAL
*          NODEID
*          ------
 LOCAL     AUSESC

                                                              MORE...
```

Figure 99. RSCS CONFIG File, Continued

```
*************************************************************************
*                 RSCS OPERATOR FORM NAME SPECIFICATION                 *
*************************************************************************
*
*          OPERATOR
*          FORM NAME
*          ---------
 OPFORM    STANDARD

*************************************************************************
*                 RSCS CHANNEL RESERVATION SPECIFICATION                *
*************************************************************************
*
*          RESERVE THESE
*            CHANNELS
*          -------------
 CHANNELS  4

*************************************************************************
*               RSCS STORE AND FORWARD CLASS SPECIFICATION              *
*               FOR RECEIVING FILES                                     *
*************************************************************************
*
*          CLASS COMMENTS
*          ----- -------------------------------------------
 SAFCLASS  *     '*' MEANS USE THE CLASS OF THE RECEIVED FILE

*************************************************************************
*        RSCS LINK, ROUTE, PARM, AND AUTH SPECIFICATIONS                *
*************************************************************************
*
*              LINK     VIRT   SPOOL KEEP  QUEUE             LOGMODE  AUTO
*     LINKID   TYPE     ADDR * CLASS SLOTS TYPE  DP LUNAME   NAME     START
*     -------- -------- ---- - ----- ----- ----- -- -------- -------- -----
*LINK AUSVHAT1 NJE      305  * *     2     PRI   *  *        *        AST
*LINK AUSHAT4  NJE      021  * *     2     PRI   *  *        *        AST
 LINK DHATCHAN SNANJE   *    * *     16    PRI   *  TH1ECP   *        AST
 LINK DHATRISC SNANJE   *    * *     16    PRI   *  TH1BCP   *        AST

*
*     LINKID   PARM TEXT
*     -------- ------------------------------
*PARM AUSVHAT1 STREAMS=2 TA=1 TAPARM='TH=100'

*

 ROUTE    *   DHATCHAN
* NETWORK ROUTES - END
                                                          MORE...
```

Figure 100. RSCS CONFIG File, Continued

```
*************************************************************************
*             GIVE COMPLETE RSCS AUTHORIZATION TO OPERATOR              *
*************************************************************************
*          LINKID    USERID    NODEID     CP
*          --------  --------  --------   --
     AUTH     *      OPERATOR     *       CP
     AUTH     *      MAINT        *       CP
     AUTH     *      VMFTP1       *       CP
     AUTH     *      RICH         *       CP
     AUTH     *      MERLIN       *       NOCP
     AUTH     *      WENZLAFF     *       CP

*************************************************************************
*                  RSCS BISYNC DIAL PORT DEFINITIONS                    *
*************************************************************************
*
*      VIRTUAL DIAL OR
*      ADDRESS NODIAL
*      ------- -------
 PORT  080     NODIAL
 PORT  081     NODIAL
 PORT  082     DIAL

*************************************************************************
*             GIVE COMPLETE RSCS AUTHORIZATION TO OPERATOR              *
*             AND SPECIAL USERIDS.  THE SAMPLE AUTH STATEMENTS          *
*             SHOWN BELOW ARE REQUIRED IF YOU ARE USING IPF             *
*             TO CUSTOMIZE AND/OR OPERATE RSCS.                         *
*************************************************************************
*
*************************************************************************
*                   RSCS SUPERVISOR SPECIFICATIONS                      *
*************************************************************************
*
*                              COMMENTS
*                              ----------------------------------

 DUMP        VM    OPERATNS    DUMP TYPE AND USERID TO SEND IT TO

 MSGNOH                        SPECIFY NO HEADER (THE RSCS VIRTUAL
*                              MACHINE MUST BE PRIVILEGE CLASS B
*                              <OR EQUIVALENT USER-DEFINED CLASS>
*                              TO USE THIS)
*
*    THE SAMPLE DUMP AND MSGNOH STATEMENTS SHOWN ABOVE ARE REQUIRED
*    IF YOU ARE USING IPF TO CUSTOMIZE AND/OR OPERATE RSCS.
*
*************************************************************************
*                   EXIT ROUTINE SPECIFICATIONS                         *
*************************************************************************
```

Figure 101. RSCS CONFIG File

C.1.7 VSCS DIRECT

This is the VSCS DIRECT file.

```
USER VSCS T1MEOUT 24M 32M ABG                                          07171642
* VSCS OWNER:<SYSTEM>,<607835>                                          07171642
 OPTION MAXCONN 400 QUICKD DIAG98                                      07171642
 SHARE REL 800                                                         07171642
 IUCV *CCS P M 10                                                      07171642
 IUCV ANY P M 0                                                        07171642
 ACCOUNT 8000                                                          07171642
 IPL CMS PARM AUTOLOG                                                  07171642
 MACH XA                                                               07171642
NAMESAVE GCSXA VTAM NETVSG00                                           07171642
 CONSOLE 01F 3215 W OPERATOR                                           07171642
 SPOOL 00C 2540 READER A                                               07171642
 SPOOL 00D 2540 PUNCH  A                                               07171642
 SPOOL 00E 1403 A                                                      07171642
 LINK MAINT 190 190 RR                                                 07171642
 LINK MAINT 595 595 RR                                                 07171642
 LINK VTAMAINT 298 192 RR                                              07171642
*LINK VTAMAINT 39A 29A RR                                              07171642
 LINK VTAMAINT 59A 29A RR                                              07171642
MDISK 0191 3380 528 3 VM4SP3 MR                                        07171642
```

Figure 102. VSCS DIRECT

List of Abbreviations

APA	All Points Addressable
BLKMUX	Block Multiplexer
BMCA	Block Multiplexer Channel Adapter
CCS	Console Communication Services
CHPID	Channel Path ID
CLAW	Common Link Access to Workstation
CLIO	Client Input Output Application
CTC	Channel-to-Channel
DASD	Direct Access Storage Device
DLC	Data Link Control
ESA	Enterprise System Architecture
ESCD	ESCON Director
ESCON	Enterprise System Connection
EXD	Extended Distance Facility
FTP	Group Control System
GCS	File Transfer Protocol
GTF	Generalized Trace Facility
HCD	Hardware Configuration Definition
HCON	Host Connect Program
HIPPI	High Performance Parallel Interface
IBM	International Business Machines Corporation
IOCDS	Input Output Configuration Data Set
IOCP	Input Output Configuration Program
ITSO	International Technical Support Organization
IUCV	Inter User Communication Vehicle
IVP	Installation Verification Procedure
JCL	Job Control Language
LDSF	Logical Device Support Facility
LU	Logical Unit
NETID	Network Identification
NFS	Network File System
NSS	Named Saved System
PROFS	Professional Office System
PU	Physical Unit
RACF	Resource Access Control Facility
RSCS	Remote Spooling Communications Subsystem
RS6K	RS/6000
RTM	Real Time Monitor
SNA	System Network Architecture
TCP/IP	Transmission Control Protocol/Internet Protocol
UIM	Unit Information Model
USS	Unsupported System Services
VTAM	Virtual Telecommunications Access Method
VM	Virtual Machine
VMCF	Virtual Machine Communication Facility
VSCS	VTAM SNA Console Support
VTAM	Virtual Telecommunications Access Method
XID	Exchange Identifier

Index

Special Characters

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

ITSO Technical Bulletin Evaluation RED000

International Technical Support Organization
RISC/6000 to Mainframe
Using S/370 Channel Connections
January 1996

Publication No. SG24-4589-00

Your feedback is very important to help us maintain the quality of ITSO Bulletins. **Please fill out this questionnaire and return it using one of the following methods:**

- Mail it to the address on the back (postage paid in U.S. only)
- Give it to an IBM marketing representative for mailing
- Fax it to: Your International Access Code + 1 914 432 8246
- Send a note to REDBOOK@VNET.IBM.COM

Please rate on a scale of 1 to 5 the subjects below.
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction	____		
Organization of the book	____	Grammar/punctuation/spelling	____
Accuracy of the information	____	Ease of reading and understanding	____
Relevance of the information	____	Ease of finding information	____
Completeness of the information	____	Level of technical detail	____
Value of illustrations	____	Print quality	____

Please answer the following questions:

a) If you are an employee of IBM or its subsidiaries:

Do you provide billable services for 20% or more of your time? Yes____ No____

Are you in a Services Organization? Yes____ No____

b) Are you working in the USA? Yes____ No____

c) Was the Bulletin published in time for your needs? Yes____ No____

d) Did this Bulletin meet your needs? Yes____ No____

If no, please explain:

_____

_____

What other topics would you like to see in this Bulletin?

_____

_____

What other Technical Bulletins would you like to see published?

_____

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

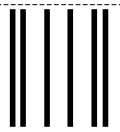
_____ _____
Name Address

_____ _____
Company or Organization

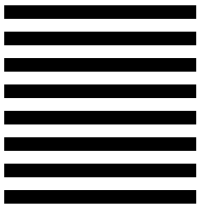
_____ _____
Phone No.

Cut or Fold
Along Line

Fold and Tape **Please do not staple** Fold and Tape



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM International Technical Support Organization
Department JN9B, Building 045
Internal Zip 2834
11400 BURNET ROAD
AUSTIN TX
USA 78758-3493

Fold and Tape **Please do not staple** Fold and Tape

Cut or Fold
Along Line

Printed in U.S.A.