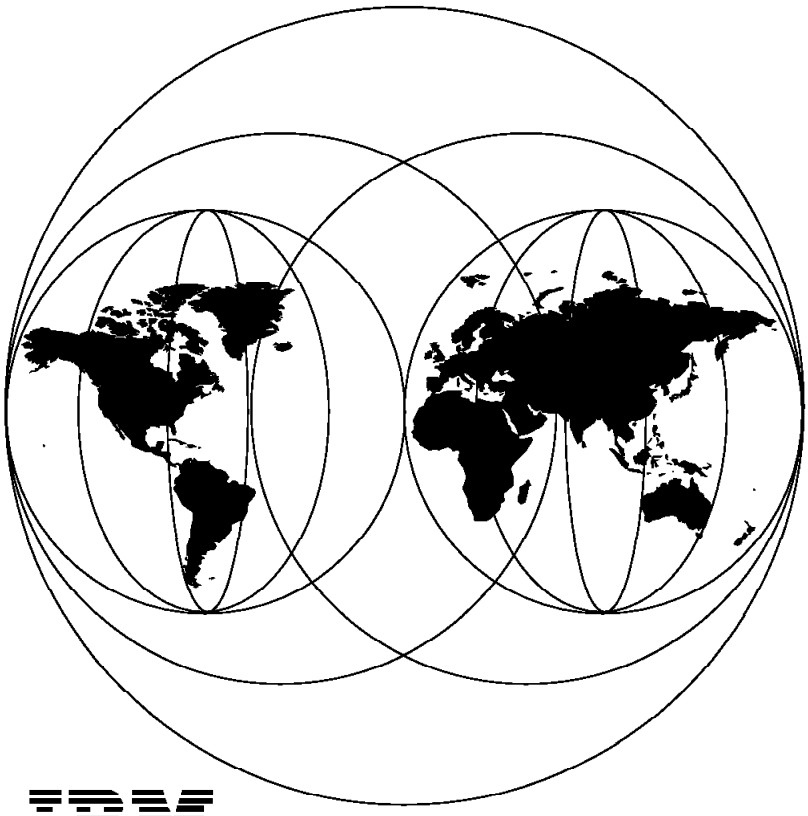
International Technical Support Organization

SG24-4619-00

OpenEdition MVS Communication Server Implementation Guide

January 1996



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4619-00

OpenEdition MVS Communication Server Implementation Guide

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 the OpenEdition Communications Server which is included in MVS/ESA System Product Version 5.2.2(5655-068 or 5655-069).

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. 541 Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

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

Abstract

This document is unique in its detailed coverage of the MVS/ESA SP 5.2.2 component OpenEdition Communication Server (known as the Outboard Communication Server), which will be called OCS through the rest of the document. This MVS/ESA component provides three ways to get to the OpenEdition MVS shell directly without having to use the TSO interface: directly with an rlogin command or, for ASCII terminals connected through a Communications Server, with either rlogin or a telnet command. It covers planning, installation, and operation of the product. Also, administration and diagnostic tasks are described.

This document was written for analysts, administrators, system support, and users who are planning to install OCS or looking for more information on how it functions.

(123 pages)

Contents

Figures

Tables

Special Notices

This document is unique in its detailed coverage of the MVS/ESA SP 5.2.2 component OpenEdition Communication Server (known as the Outboard Communication Server), which will be called OCS through the rest of the document. It covers planning, installation, and operation of the product. Also, administration and diagnostic tasks are described.

This publication is intended to help analysts, administrators, system support, and users to assist in planning to install OCS or who are looking for more information on how it functions. The information in this publication is not intended as the specification of any programming interfaces that are provided by the OpenEdition Communication Server (OCS) product. See the PUBLICATIONS section of the IBM Programming Announcement for the OpenEdition Communication Server 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.

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:

Advanced Peer-to-Peer Networking
AIX
AIX/6000
AIXwindows
CBIPO
DFSMS
DFSMS/MVS
DFSMSdfp
Enterprise System/9000
Enterprise Systems Architecture/390
ES/9000
ESA/390
IBM
MVS/DFP
MVS/ESA
OpenEdition
OpenEdition Communications Server
Operating System/2
OS/2
OCS
RACF
RISC System/6000
S/390
System/390
VTAM

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.

Other trademarks are trademarks of their respective companies.

Preface

This document is intended to assist with planning, installation, and operation of OCS. A chapter is included to detail OCS scenarios that a OCS user might experience. Also included, are three appendices that will provide additional supplemental information to ease the installation and configuration of OCS in your environment.

This document was written for analysts, administrators, system support, and users who are planning to install OCS or who are looking for more information on how it functions.

How This Document is Organized

The document is organized as follows:

- Chapter 1, “Introduction to OCS” on page 1 – States the benefits of OCS and the purpose for this project. It also includes OCS minimum system requirements and our Lab′s hardware and software environment used for this project.
- Chapter 2, “Description of OCS” on page 5 – Provides a high-level description of ‘What is OCS?’ and ‘How does it work?’
- Chapter 3, “Planning for OCS” on page 13 – Provides detailed planning for the installation of the major components of OCS.
- Chapter 4, “Installation of OCS” on page 27 – Explains how to install and configure OCS on both MVS/ESA OpenEdition and the RISC System/6000.
- Chapter 5, “OCS Checklist” on page 83 – This chapter presents a checklist for installing and customizing OCS for OpenEdition/MVS and the RISC System/6000 Environment.
- Chapter 6, “Tuning OCS” on page 87 – Explains how to tune the OCS configuration for each OCS node.
- Chapter 7, “OCS Error Recovery Scenarios” on page 91 – Documents working scenarios and describes some of the commands to control and display the different elements associated with OCS.
- Chapter 8, “OCS Diagnostics” on page 101 – Describes the techniques for obtaining information on both the OpenEdition/MVS host and the RISC System/6000 server machine to help diagnose OCS problems.

In addition, the following appendices are provided:

- Appendix A, “OpenEdition/MVS and OCS Code Page Translation” on page 103 – Describes the character encoding considerations that are specific to the OpenEdition/MVS MVS environment.
- Appendix B, “CLAW Tuning” on page 107 – Describes the elements for tuning the Common Link Access to Workstation (CLAW) interface for the CLAW-attached OCS Server.
- Appendix C, “OCS Experiences” on page 109 – Provides some helpful information and hints that can help you solve some of the problems that you may encounter while installing and using OCS.

Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this document. Documents with an LY prefix are available to IBM-licensed customers only.

- *OpenEdition MVS Communications Server Guide*, SC23-3883-00
- *MVS/ESA Diagnosis: Procedures*, LY28-1844
- *IBM MVS/ESA Initialization and Tuning Reference*, SC28-1452
- *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02
- *MVS/ESA OpenEdition MVS Command Reference*, SC23-3014-02
- *IBM TCP/IP for MVS: Customization and Administration Guide*, SC31-7134-00
- *IBM TCP/IP Performance Tuning Guide*, SC31-7188-00
- *ES/9000, ES/3090 IOCP User s Guide*, GC38-0097
- *MVS/ESA Interactive Problem Control System (IPCS) Commands*, GC28-1491
- *Block Multiplexer Channel Adapter: Users Guide and Service Information*, SC23-2427-02
- *System Management Guide: Communications and Networks*, SC23-2526-02
- *AIX Version 4.1 Problem Solving Guide and Reference*, SC23-2606

International Technical Support Organization Publications

- *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*, SC24-4529-00

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. Visa and MasterCard are accepted. Outside the USA, customers should contact their local IBM office. For guidance on ordering, send 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

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:

Fernando Alfonso, Jr.
International Technical Support Organization, Poughkeepsie Center

The authors of this document are:

Tom King
IBM Washington Systems Center, Gaithersburg

Jim Skorey
IBM Integrated Systems Solution Corporation, San Jose

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

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

Jan Baisden
International Technical Support Organization, Poughkeepsie Center

John Case
S390 Open Interface Development, Poughkeepsie

Rich Conway
International Technical Support Organization, Poughkeepsie Center

Chapter 1. Introduction to OCS

This is a supplement to the book MVS/ESA OpenEdition MVS Communications Server Guide. This project's objective was to assist analysts, system administrators, and support personnel to understand the installation, administration, and usage of the OCS product.

1.1 Benefits of OCS

OCS is a new terminal attachment mechanism that runs on the MVS host and AIX RISC/6000 server. It provides direct login to the OpenEdition MVS shell on MVS for terminals via:

telnet virtual terminal protocol
rlogin virtual terminal protocol
AIX serial-attached ASCII terminals

Better performance is provided to large configurations of interactive users connected to OpenEdition MVS via OCS than to terminals logged in by telnet or rlogin without OCS.

Common Link Access to Workstations (CLAW) protocol is supported by OCS, which improves system performance by allowing channel rate data transfer.

OCS reduces the number of interrupts to the OpenEdition MVS system by collecting input from multiple terminals in a packet before sending it to the host, and by providing the terminal line discipline processing. Line discipline processing performs primitive terminal character processing such as echoing input characters back to the terminal. Also, in line mode a whole line of data is built at the OCS server and forwarded to the host. Without line mode each character will interrupt the host.

Users of OCS can log into the host as if they were logging into a local system. OCS makes logging in easier.

1.2 Purpose of This Project

This project was defined to investigate and document the tasks involved in planning, installation, and configuration for OCS. The configuration used was smaller than that used by a normal customer. The following are the major items covered in this redbook:

- Document how to plan for the implementation in a customer environment
- Perform the installation of the program product and document how the installation was completed on both the AIX server and MVS/ESA mainframe
- Describe the operation of OCS and including examples of use
- Document the symptoms of the problems that occurred, along with the actions taken to solve the problems

1.3 Minimum System Requirements

RISC System/6000 uniprocessor (UP) configuration.

For channel connection:

Block multiplexer channel adapters for RISC System/6000 (feature #2755).
Block multiplexer channel adapters cables (feature #2757)
Block multiplexer channel interface assemble (feature #2758)
Channel cables (bus and tag) (cable group #0185)

For LAN connection:

Use a LAN attached RISC System/6000 as the LAN gateway. You need at least one network interface (LAN adapter).

Host system requirements:

OpenEdition MVS Version 5 Release 2.2 or later with OCS support.
TCP/IP for MVS Version 3 Release 1.

RISC System/6000

AIX Version 4 Release 1.3 or later.
For channel attached configure, you need the block multiplexer channel interface device driver licensed product 5697-037. This device driver corresponds to the CLAW device driver on the OpenEdition MVS system.

1.4 ITSO Lab Environment

The lab was located at the International Technical Support Center (ITSC), Poughkeepsie, New York. The environment, see Figure 1 on page 4, was configured to a minimum and most likely would not present itself as the “normal”customer environment. Minimizing the environment may also insure a successful startup of a minimum OCS configuration for testing and evaluation prior to rolling the OCS server code to other servers in your network.

The following sections will provide the hardware and software configurations and also a block diagram of the components, TCPIP, and CLAW connections that will be referenced throughout this document.

1.4.1 Hardware Environment

The following is the specific hardware that was part of the OCS configuration:

IBM 9021 Model 982 (A-side) of a LPAR partition called LABSYS

Two - IBM RISC/6000 Model 570 uniprocessor:

LAN network interface adaptor (token-ring)

Two serial ports for attachment of one IBM 3151 and one IBM 3161 asynchronous ASCII terminal

RISC71 has 256M of memory, two 2GB hard drives, a 1GB drive and was configured as our OCS LAN-attached server.

RISC73 has 128M of memory, three 2GB hard drives, and was configured as our OCS CLAW-attached server.

CLAW support required the block multiplexer (BMX) channel adapters for RISC System/6000 (feature #2755), BMX channel adapter cables (feature #2757), BMX channel interface assembly (feature #2758) and channel cables (bus and tag) from cable group #0185

There were various other sundry type terminals for remote rlogin and telnet connections via OCS Virtual Terminal Support. See 2.4, “OCS Components” on page 9 for further description of OCS Virtual Terminal Support.

1.4.2 Software Environment

The following is the specific software that was part of the OCS configuration.

The MVS host has the following software installed and configured:

MVS/ESA SP 5.2.2

CBIPO level 95A

This includes the following FMIDs:

FMIDs	Description
jbb5522, jbb52n0	Basic Contol Program (BPC) SP 5.2.2.
hom1130, jom13n0	OpenEdition MVS System Services
hot1130, jot13n0	OpenEdition MVS Application Services
hsu1130, jsu13n0	OpenEdition MVS Shell and Utilities
hdx1130	OpenEdition MVS Debugger
hmw1510	Language Environment/370

TCP/IP Version 3 Release 1

The RISC/6000 OCS server has the following software installed and configured:

AIX/6000 Version 4 Release 1.3

1.5 OCS Environment Diagram

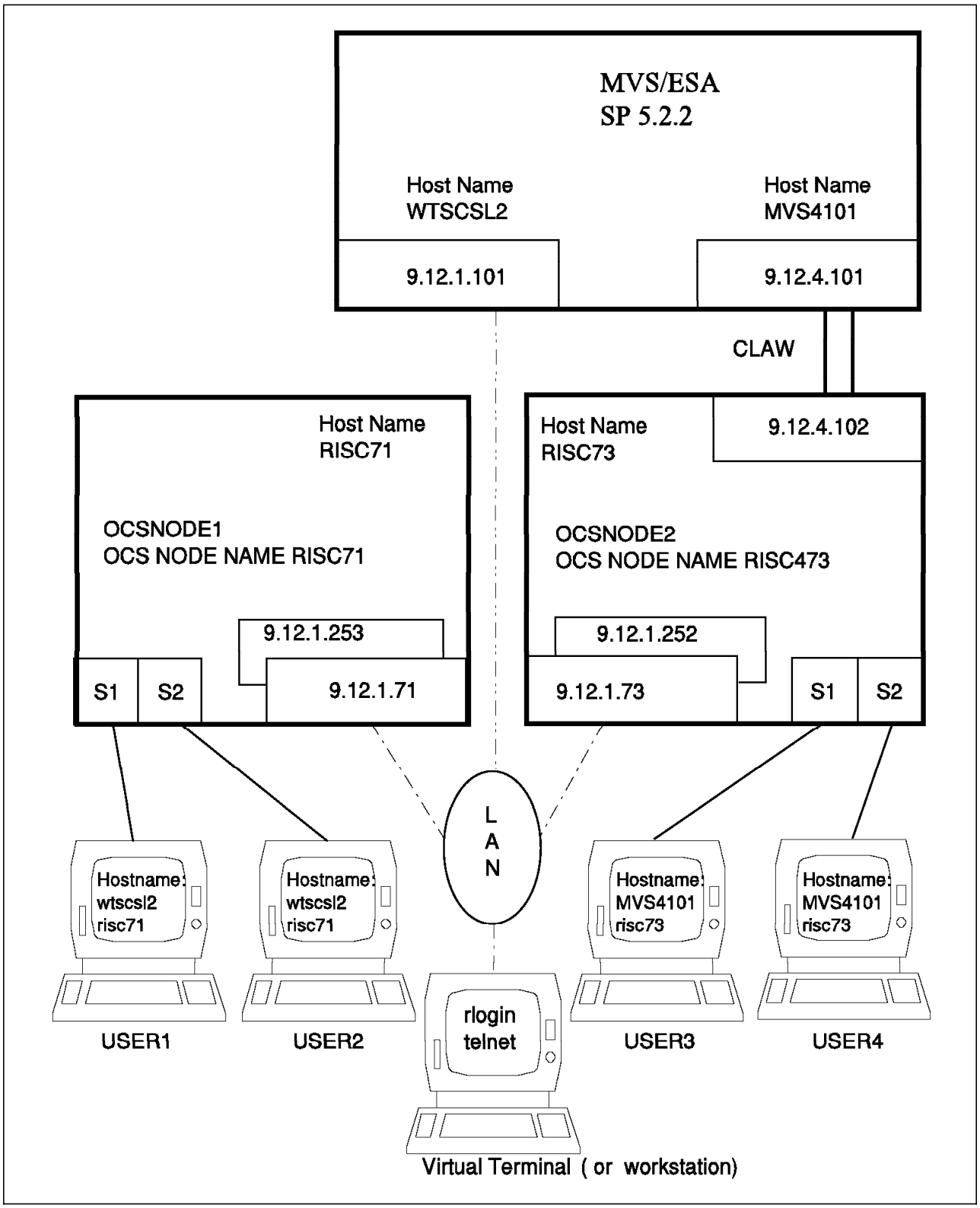


Figure 1. OCS Hardware Environment with TCP/IP Connectivity

Chapter 2. Description of OCS

This chapter describes the OpenEdition/MVS Communications Server (known as the Outboard Communications Server).

This chapter contains the following topics:

- What is the Outboard Communication Server (OCS)?
- Attachment options for OCS
- How does NON-OCS work?
- How OCS Works?
- OCS Components

2.1 What Is the Outboard Communications Server?

The Outboard Communication Server (OCS) is a RISC System/6000 connected to one or more OpenEdition/MVS systems, either by a channel or a local area network (LAN).

With this connection, terminals on asynchronous ports of the RISC System/6000 can operate as if they were connected directly to the OpenEdition/MVS system. Serial terminal users can log directly into the OpenEdition/MVS system instead of using remote login procedures. At the same time, the RISC System/6000 can continue to function as a standard TCP/IP host or gateway node.

Users can `telnet` or `rlogin` to the OpenEdition/MVS system through the OCS. The `telnet` and `rlogin` virtual terminal protocols are offloaded to the OCS, increasing the efficiency of the OpenEdition/MVS system.

OCS support is supplied in three parts:

- The Communications Server code, which ultimately resides and executes on the RISC System/6000 using AIX V4.1.3 as the base operating system
- Support code in the base OpenEdition/MVS system
- Support code in the base AIX V4.1.3 system running on the RISC System/6000

2.1.1 Attachment Options

An OCS system is the combination of one or more OpenEdition/MVS hosts and an attached OCS.

Each OpenEdition/MVS host can support multiple OCS server systems; each OCS server system can support multiple hosts. Figure 1 on page 4 illustrates the two ways (LAN and channel) that OCS systems can be attached to the OpenEdition/MVS host system. In Figure 1 on page 4:

1. OCS Server 1 (RISC71) is connected to the OpenEdition/MVS host by means of a LAN. Serially attached terminals (¢USER1¢ and ¢USER2¢) access the OpenEdition/MVS host via OCS server 1.
2. OCS Server 2 (RISC73) is connected to the OpenEdition/MVS hosts by means of Common Link Access to Workstations (CLAW) interface. Serially attached terminals (¢USER3¢ and ¢USER4¢) access the OpenEdition/MVS host via OCS Server 2.

3. The workstation accesses the OpenEdition/MVS host via telnet or rlogin using either OCS Server 1 or OCS Server 2.

2.2 How Does Non-OCS Work?

In the non-OCS environment, end users are gaining access to host applications on OpenEdition MVS by rlogin from remote workstations. This operation can be performed directly by the direct attached workstation node or from a complex networking attached node. Let's look at two examples, first the token-ring attached terminal and then, the serial attached terminal.

Figure 2 on page 7 shows the connectivity of both examples without OCS support. The token-ring terminal, USER1, is connected into the host via a backbone Local Area Network (LAN). The LAN is connected to the OpenEdition/MVS host via a channel attached 3172. User 1 can rlogin into OpenEdition/MVS to gain access to OpenEdition/MVS applications.

User 2 is a serially-attached terminal to a RISC System/6000. The RISC System/6000 can either be LAN-attached or channel-attached via a CLAW, or both. In Figure 2 on page 7, both type of connections are configured for User 2. To gain access to the OpenEdition MVS applications, User 2 must perform two steps. First, he logs in to the local AIX system. Second, he uses rlogin to gain access to OpenEdition MVS.

In both of these scenarios for ¢User 1¢ and ¢User 2¢, the terminals will operate in character or "Raw" mode. The OpenEdition/MVS host will perform primitive line discipline processing such as echoing input characters back to the terminal.

This process will create an OpenEdition/MVS interrupt for each keystroke entered at either ¢User 1¢ or ¢User 2¢. In a very interactive system that includes numerous workstation accesses, this will provide a significant overhead to your OpenEdition/MVS system. This is where OCS will help.

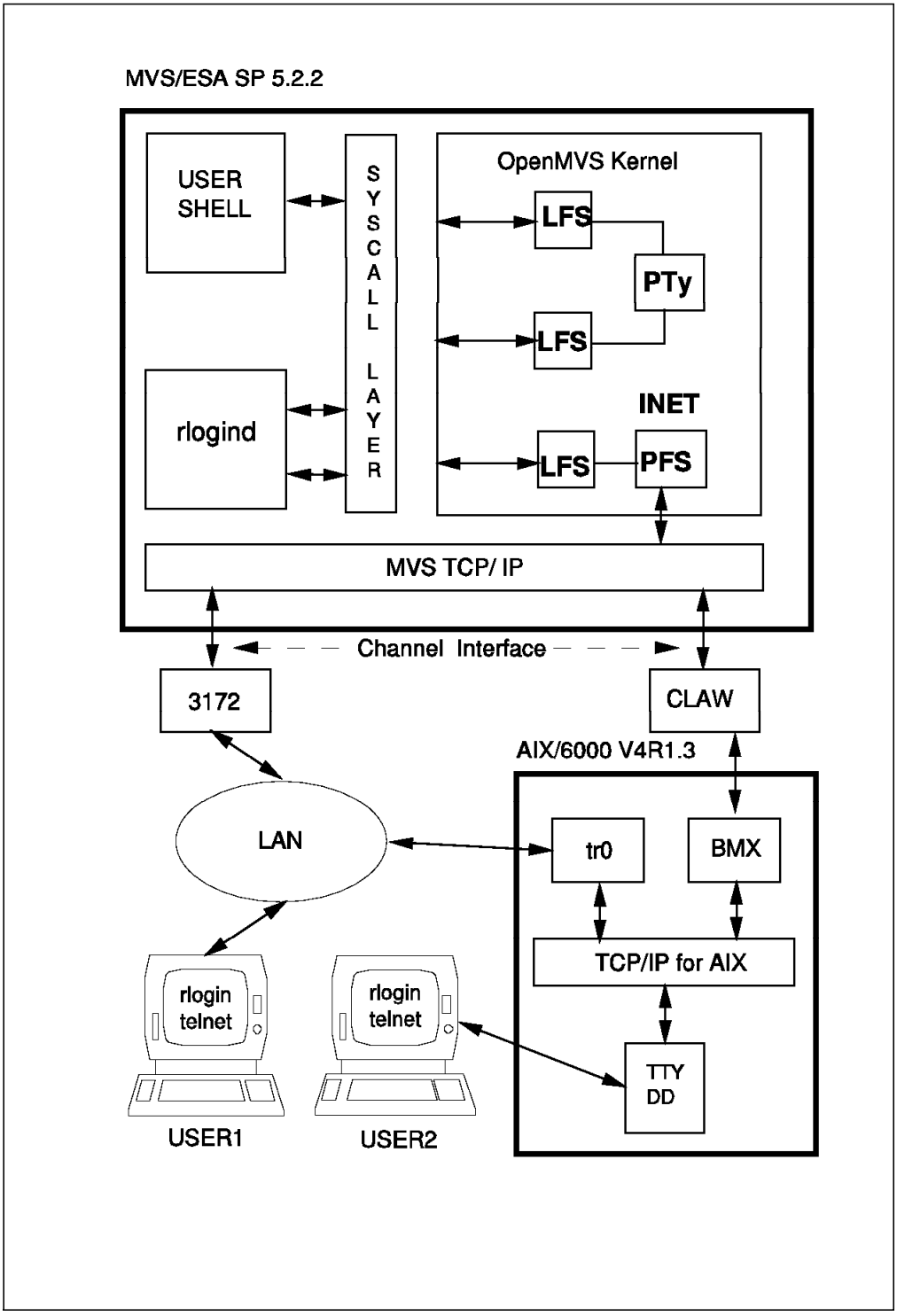


Figure 2. Non-OCS Direct Login Support

2.3 How OCS Works

OCS offloads the terminal line discipline processing from the OpenEdition/MVS host to the OCS server system. For remote access to the OpenEdition/MVS host system using rlogin or telnet virtual terminal protocols, the rlogin and telnet server function is also offloaded to the OCS server system. Finally, the OCS server will multiplex or collect multiple terminal inputs into packets prior to sending them to the OpenEdition/MVS host, thus providing further savings by utilizing more efficient TCP/IP packet transfers from multiple workstations to your OpenEdition/MVS host machine. The net result is that fewer I/O interrupts occur at the host machine and primitive terminal processing is offloaded to the OCS server system.

In Figure 3 on page 9, the ¢Serial Terminal¢ is serially connected to the RISC System/6000 serial port and will use OCS services to directly login to the OpenEdition/MVS host system without requiring the terminal user to previously login to the local AIX system. The terminal connection to the OpenEdition/MVS host can be achieved by using either the LAN connection or via the CLAW channel connection. In our environment, see Figure 1 on page 4, we did not include the LAN definitions for USER3 and USER4 to connect from OCS server 2 (RISC73) to OpenEdition/MVS. This could be included for alternate host support, if so desired.

Remote access to the OpenEdition/MVS host system via an OCS server is achieved by specifying an OCS ″virtual host address″ on the telnet or rlogin command. An OCS ″virtual host address″ is an alias address that you have defined for the LAN connection on your OCS server machine, and have configured for OCS use, In Figure 1 on page 4, 9.12.1.253 is the assigned alias address for OCSNODE1 (RISC71) and is configured as an OCS virtual host address for the OpenEdition/MVS host; an rlogin or telnet connections using this virtual host address will cause OCSNODE1 to establish a user connection with the OpenEdition/MVS host via the LAN interface. Likewise, 9.12.1.252 is assigned alias address for OCSNODE2 (RISC73) and is configured as an OCS virtual host address for the OpenEdition/MVS host; an rlogin or telnet connections using this virtual host address will cause OCSNODE2 to establish a user connection with the OpenEdition/MVS host via the CLAW interface.

Proper steps must be taken to insure that the routing and name resolution is correct for the network. This is described further in 4.5.2, “Assigning Internet Addresses” on page 51.

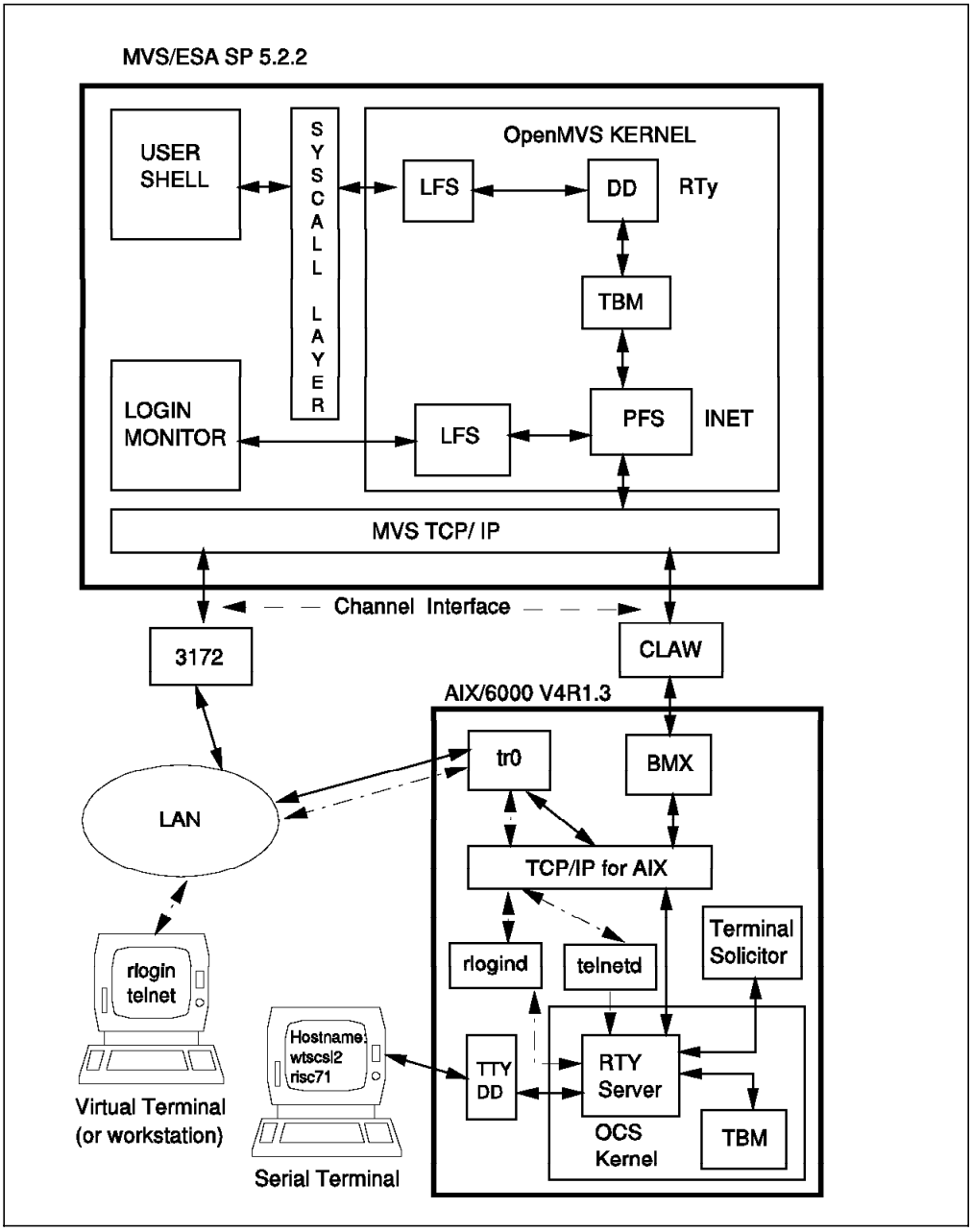


Figure 3. OCS Direct Login Support

2.4 OCS Components

This section describes the components of OCS on the OpenEdition/MVS system and the RISC System/6000.

OCS support consists of new or enhanced components on AIX Version 4 Release 1.3 and OpenEdition/MVS. The following is a brief description of the new OCS components and the existing OpenEdition/MVS components that are depicted in Figure 3.

The following are new OCS components:

OCS Serial Line Support – takes advantage of existing terminal support in AIX 4.1.3. It includes a remote terminal (DD RTY) device driver on the OpenEdition/MVS host system and an rty server (RTY Server) on AIX 4.1.3, the Terminal Buffer Manager, the terminal solicitor, and the login monitor.

Rty Device Driver (RTY) – communicates with the rty server via the remote tty protocol (RTP). The rty device driver presents a standard tty system call interface to a shell or application. Using the RTP, it sends system calls and outbound data to the rty server for processing.

Remote-tty Server (TTY DD) – drives serial lines on the RISC System/6000, using a standard tty interface, to present the system calls to existing terminal support in AIX 4.1.3. It returns status and incoming terminal data to the rty device driver.

The rty server is defined in the Object Data Manager (ODM) database like an AIX 4.1.3 device driver.

Terminal Buffer Manager (TBM) – in AIX 4.1.3, collects input from several terminals in a packet before sending it to the host.

Terminal Buffer Manager (TBM) – in the OpenEdition/MVS host system collects output to several terminals in a packet before sending it to OCS.

Terminal Solicitor (ts) – When an OCS terminal is turned on, the terminal solicitor checks the ODM database for the systems the terminal is permitted to log into. Then it prompts the user to select one of those systems. If the user chooses the AIX 4.1.3 system on the OCS machine, the terminal solicitor spawns a `ts_getty` process to begin login locally. If the user chooses an OpenEdition/MVS host system, the terminal solicitor sends a message to the login monitor on the destination OpenEdition/MVS host system to begin login on that host system.

Login Monitor (lm) – is an OpenEdition/MVS daemon that listens for login requests on the AF_INET socket port that corresponds to the 'lm' service. When the login monitor starts to run, it creates an rty device table containing terminal names. The login monitor maps the OCS terminal name in the terminal solicitor's message to the OpenEdition/MVS name for the corresponding rty device, then spawns a child process to open the file and log into the OpenEdition/MVS system.

Virtual Terminal Support – is done when `rlogind` or `telnetd` sends a message to the login monitor indicating that a remote user is logging into the OpenEdition/MVS host system via the OCS.

telnetd – starts a login on either the RISC System/6000 or the OpenEdition/MVS system. When `telnetd` receives a login request, it compares the requested address with the virtual host addresses in the ODM database. If the addresses match, `telnetd` requests the login monitor to start a login on the OpenEdition/MVS host system. If they do not match, it assumes that the address is on the RISC System/6000 and starts the login there.

rlogind – starts a login on either the RISC System/6000 or the OpenEdition/MVS system. When `rlogind` receives a login request, it compares the requested address with the virtual host addresses in the ODM database. If the addresses match, `rlogind` requests the login monitor to start a login on OpenEdition/MVS. If they do not match, it assumes that the address is on the RISC System/6000 and starts the login there.

Note

Releases of AIX on the RISC System/6000 that support OCS include enhanced versions of `telnetd` and `rlogind`, which implement the `telnet` and `telnetd` virtual terminal protocol.

The following are OpenEdition/MVS components:

LFS – (Logical File System) is a component of the OE/MVS file system support that routes virtual file system requests (VFS requests) to the target physical file system (PFS).

PFS – (Physical File System) is a physical file system that controls access to data.

CSFS – (Character-special file system) is an internal PFS that provides access to OE/MVS character-special files.

CSF – (character special file) is a:

1. Special file that provides access to an input or output device. The character interface is used for devices that do not use block I/O.
2. A file that refers to a device. One specific type of character special file is a terminal device file. Other character special files have no structure defined by POSIX.1, and their use is unspecified by POSIX.1.

INET (INET - Address Family) – In OpenEdition MVS, there are two socket families supported--UNIX Domain Sockets, known as local sockets, which are part of the UNIX Address Family (AF_UNIX); and Internet Protocol Sockets, which are part of the Internet Address Family (AF_INET).

Chapter 3. Planning for OCS

3.1 Planning for MVS/ESA OpenEdition/MVS

To use OCS you must first have installed MVS/ESA 5.2.2 OpenEdition. Also, OpenEdition MVS Shell and Utilities feature must be installed. You can also optionally install the OpenEdition MVS Debugger feature, if so desired.

3.2 MVS Setup

Perform the following installation steps.

3.2.1 Install MVS/ESA 5.2.2 BCP (jbb5522 jbb52n2)

Install all parts of the MVS operating system at the required levels and updated with maintenance. The operating system consists of an MVS/ESA system and associated IBM licensed programs, plus all other program products at the right level. The OpenEdition MVS component requires the same hardware as other MVS/ESA components and associated licensed programs. If software and hardware components are not at the required levels, upgrade them to the correct levels.

3.2.2 Install OpenEdition MVS System Services (hom1130 jom13n0)

System Modification Program Extended (SMP/E) will be used to install the OpenEdition MVS component. OpenEdition MVS Shell and Utilities, the OpenEdition MVS Debugger, and the OpenEdition MVS Application Services cannot be installed at this time. You must wait until MVS/ESA 5.2.2 has been IPLed and is active to continue the installation of the above mentioned programs.

Because of time limitations, we IPLed at this time. The recommended action is to review the program directory for additional considerations. One area of concern is the existence of a HFS data sets; if they are present you should back them up before proceeding.

3.2.3 Setup DFSMS Environment for OpenEdition MVS

DFSMS/MVS manages the hierarchical file system (HFS) data sets that contain the file system. Detailed step-by-step procedures for installation and customization of DFSMS can be found in *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*, SC24-4529-00. HFS will contain the installed libraries for OpenEdition MVS products (Shell and Utilities, Debugger, and Application Services).

If you have not previously installed DFSMS/MVS, you will now have to install and configure it. The minimal DFSMS/MVS installation will require one dedicated physical volume of DASD.

Activate SMS by doing the following:

1. Create or update IGDSMSxx parmlib member. Figure 4 on page 14 shows the member IGDSMSxx that we used.

```
SMS ACDS(SMS.ACDS1.ACDS)
    COMMDS(SMS.COMMDS1.COMMDS)
    INTERVAL(15)
    DINTERVAL(150)
    REVERIFY(NO)
    ACSDEFAULTS(NO)
    TRACE(ON)
    SIZE(128K)
    TYPE(ERROR)
    JOBNAME(*)
    ASID(*)
    SELECT(ALL)
 /* DESELECT(...) NOT SPECIFIED                          */
 /*                                                      */
 /* LIB: SYS1.PARMLIB(IGDSMS02)                          */
 /* GDE: CBIPO MVS CUSTOMIZATION                         */
 /* DOC: THIS MEMBER SPECIFIES SMS INITIALIZATION INFORMATION   */
 /*      ALL ENTRIES ARE USING DEFAULTS                  */
```

Figure 4. SYS1.PARMLIB Member IGDSMSxx

2. Update SYS1.PARMLIB member IEFSSNxx to contain an entry for SMS. Figure 5 shows our IEFSNN00 member.

```
SMS,IGDSSIIN,'ID=00,PROMPT=DISPLAY' SMS
JES2,,,PRIMARY JES 2 SUBSYSTEM
RACF,IRRSSI00,# RACF ADDRESS SPACE
TNF,MVPTSSI
VMCF,MVPXSSI,WTSCSL2
```

Figure 5. SYS1.PARMLIB Member IEFSSN00

3. Prepare DASD volumes for SMS storage groups. Device Support Facility (ICKDSF) INIT command is used to assign DASD volume serial numbers and allocate VTOC. The STORAGEGROUP(STGR) keyword of the INIT command is used to make a DASD volume available for SMS managed data sets.
4. Allocate a SMS control data set and create a SMS configuration. Refer to *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*, SC24-4529-00 for a step-by-step configuration procedure.
5. If SMS has not been previously activated in the system, an IPL must be done to activate SYS1.PARMLIB members IEFSSNxx and IGDSMSxx. If SMS is active and changes have just been made to IGDSMSxx, then issue the SET SMS=xx command, where xx defines the IGDSMSxx parmlib member.

3.2.4 IPL MVS/ESA 5.2.2

IPL the prepared MVS/ESA 5.2.2 system and activate SMS.

3.2.5 MVS/ESA OpenEdition Planning

The next sections address the different elements need to implement OpenEdition/MVS.

3.2.5.1 Customize APPC/MVS for OpenEdition MVS

OpenEdition MVS uses address spaces provided by APPC/MVS. Even if APPC/MVS was used before, you will need to customize it for OpenEdition MVS. The following must be done:

1. Create an APPC/MVS transaction program (TP) profile, which will allow forked address spaces.
2. Add a LUADD statement for OpenEdition MVS to the APPCPMxx parmlib member. On the LUADD statement used for OpenEdition MVS, omit the SCHED(ASCH) and the TPLEVEL(SYSTEM) parameters because they are DEFAULT. OpenEdition MVS requires the system base LU to be associated with ASCH-scheduler. It is using LU=OWN, which means a conversation that uses a single LU as both local LU and partner . Therefore the LUADD statement that has the TPLEVEL(SYSTEM) should be the BASE and have SCHED(ASCH) defined. If there are multiple LUADD statements available, make sure no one is defined as a system BASE LU and NOSCHED. In case of multiple LUADD statements, the system will take the last one.
3. The ACBNAME should have been defined and activated through RACF.

```
CLASSADD CLASSNAME(OPENMVS) MIN(12) MAX(12) RESPGOAL(1)
```

Figure 6. CLASSADD Statement for Member ASCHPMxx

4. Refresh or Start APPC.

   If you are already running APPC on your system, you will need to refresh it to pick up the changes for OpenEdition MVS. From the operators console issue the following:

- **SET APPC=xx**

  (where xx is the suffix of your APPCPMxx member)

- **SET ASCH=xx**

  (where xx is the suffix of your ASCHPMxx member)

Figure 7. Refresh APPC with OE Changes

If you are not currently running APPC, you will need to start it. Make sure you have an APPC and an ASCH PROC in your procedure library, and then start them:

- **START APPC,SUB=MSTR,APPC=xx**

  (where xx is suffix of APPCPMxx member)

- **START ASCH,SUB=MSTR,ASCH=xx**

  (where xx is suffix of ASCHPMxx member)

Figure 8. Starting APPC

The following message will appear:

```
ATB052E LOGICAL UNIT APPCOMVS FOR TRANSACTION SCHEDULER ASCH 169
NOT ACTIVATED IN THE APPC CONFIGURATION. REASON CODE = 5A.
```

A display command shows that the local LU now has status pending.

```
D APPC,LU,A
ATB101I  10.04.31  APPC DISPLAY 177
  ACTIVE LU¢S    OUTBOUND LU¢S    PENDING LU¢S    TERMINATING LU¢S
     00000          00000          00001             00000
  SIDEINFO=*NONE*
LLUN=APPCOMVS          SCHED=ASCH          BASE=YES
   STATUS=PENDING      PARTNERS=00000      TPLEVEL=SYSTEM
   TPDATA=SYS1.OMVS.APPCTP
```

A status of pending is normal because OpenEdition/MVS doesn't do any APPC outbound requests.

3.2.5.2 RACF Setup for OpenEdition/MVS

On UNIX systems each user needs an account that is made up of a username, and a password. UNIX uses the /etc/passwd file to keep track of username and an encrypted password for every user on the system. Internally, the UNIX operating system uses a numeric ID to refer to a user, so in addition to username and password, the /etc/passwd file also contains a numeric ID called the user identifier or UID. UNIX operating systems differ, but generally these UIDs are unsigned 16 bit numbers, ranging from 0 to 65535. The UID is the actual number that the operating system uses to identify the user. Usernames are provided as a convenience, as an easy way for us to remember our sign on to the UNIX system. If two users are assigned the same UID, UNIX views them as the same user, even if they have different usernames and passwords. Two users with the same UID can freely read and write over each other's files and can destroy each other's processes. Assigning the same UID to multiple users is generally not recommended.

UNIX systems also have the concept of groups where you would group together many users who need to access a set of common files, directories, or devices. Like usernames and UIDs, groups have both group names and group identification numbers (GIDs). Each user belongs to a primary group that is also stored in the /etc/passwd file.

Along with username, encrypted password, UID, and GID, the /etc/passwd file also contains the user's full name, the user's home directory (the directory where a user would generally store their files) and the file name of the shell program that is executed when the user initially logs in.

With OpenEdition/MVS the concept of user accounts are the same, but the method of storing this account information is different. RACF, when used with OpenEdition/MVS, integrates the OpenEdition/MVS account information with the existing MVS account and system information to provide a central secure data base in which to store all your security information.

As with UNIX, each OpenEdition/MVS user needs a UID and GID. These UIDs and GIDs are used by OpenEdition/MVS to control the files and processes that a user may use. OpenEdition MVS security functions are implemented in RACF, partially as modifications to existing RACF functions, and partially as new RACF

functions. The security functions provided include user validation, file access checking, and privileged user checking.

OpenEdition MVS users are defined with RACF commands. When a job starts or a user logs on, the user ID and password are verified by existing MVS and RACF functions. When an address space requests an OpenEdition MVS function for the first time, RACF:

1. Verifies that the user is defined as an OpenEdition MVS user
2. Verifies that the user's current connect group is defined as an OpenEdition MVS group
3. Initializes the control blocks needed for subsequent security checks

An OMVS segment has been added to the RACF USER profile to store the UID and an OMVS segment has been added to the RACF GROUP profile to store the GID. The main idea of defining users or groups to the OpenEdition/MVS environment is to provide them with OpenEdition/MVS User Identification (UID) numbers and Group Identification (GID) numbers. See the *MVS/ESA OpenEdition MVS User s Guide*, SC23-3013 for more information on file system permission bits.

You can control access for the following users:

1. The user who is the owner of a file or directory, whose UID matches the UID for the file.
2. The user who is a member of the group whose GID matches the GID for the file.
3. Others: any other user accessing files, whose UID does not match the UID for the file; or who is not a member of the group whose GID matches. This is granting universal access to a file.

Therefore, it is recommended that you assign a *unique UID* to each user ID on your system and specify a *unique GID* for each group.

The RACF database contains, among other things, user profiles and group profiles. Within these user and group profiles is a new segment called the OMVS segment. The OMVS segment contains the UID and GID definitions for each user and group respectively. Your user profile may contain OpenEdition MVS information about you, in the OMVS segment.

The details RACF lists from the OMVS segment of the user profile are:

- The user identifier (UID)
- The initial directory pathname (HOME)
- The program pathname (PROGRAM)

To see the above information issue the `lu userid omvs` command; userid is the RACF id of the requested information. Figure 9 on page 18 shows the results of the `lu userid omvs` command.

```
OMVS INFORMATION
----------------
UID= 0000000000
HOME= /
PROGRAM= /bin/sh
READY
```

Figure 9. Display of RACF OMVS Segment

3.2.5.3 Allocate HFS Root filesystem for OpenEdition MVS

Allocate the hierarchical file system (HFS) data set that will be the root file system.

The following figure shows a sample job that defines the HFS root data set. The space parameter should either specify a nonzero value for the directory parameter or specify DSORGR=PO to create a partitioned data set. An HFS data set is limited to one physical volume. Before deciding on the DASD space parameter, review *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02.

```
//STEP1   EXEC PGM=IEFBR14
//MKFS    DD DSNAME=OMVS.ROOT,
//        SPACE=(CYL,(100,1,1)),DCB=(DSORG=PO),
//        DSNTYPE=HFS,
//        DISP=(NEW,KEEP,DELETE),
//        STORCLAS=STANDARD
```

Figure 10. Job to Allocate HFS Root Data Set

3.2.5.4 Customize the OpenEdition MVS Parmlib Member

Create a BPXPRMxx parmlib member in SYS1.PARMLIB with the options you desire for OpenEdition/MVS, based on the way your installation expects to use the OpenEdition/MVS environment. The following figure shows the BPXPRMxx that was used in our installation.

```
MAXPROCSYS (1139)
MAXPROCUSER(1139)
MAXUIDS(200)
MAXFILEPROC(12500)
MAXPTYS(256)
MAXRTYS(256)
CTRACE(CTIBPX00)
STEPLIBLIST('/SYSTEM/STEPLIB')
FILESYSTYPE TYPE(UDS)  ENTRYPOINT(BPXTUINT)
FILESYSTYPE TYPE(INET)  ENTRYPOINT(BPXTIINT)
FILESYSTYPE TYPE(HFS)
         ENTRYPOINT(GFUAINIT)
         PARM(' ')
ROOT FILESYSTEM('OMVS.ROOT')
    TYPE(HFS)
    MODE(RDWR)
MAXTHREADTASKS(1000)
NETWORK DOMAINNAME(AF_UNIX) DOMAINNUMBER(1) MAXSOCKETS(2000)
      TYPE(UDS)
NETWORK DOMAINNAME(AF_INET) DOMAINNUMBER(2) MAXSOCKETS(10000)
      TYPE(INET)
```

Figure 11. BPXPRMxx Member of SYS1.PARMLIB

To fully understand all the above parameters, refer to *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02

3.2.5.5 Start OpenEdition/MVS (OMVS)

Start the OpenEditon MVS component with the command in Figure 12. The OMVS=xx operand specifies the BPXPRMxx parmlib member that is in Figure 11.

```
START OMVS,SUB=MSTR,OMVS=00
```

Figure 12. Command to Start OpenEdition MVS

OMVS can be started under the master subsystem or under JES. We decided to start OpenEdition/MVS under the master subsystem in order to bring up OpenEdition MVS concurrently with JES during system initialization.

Figure 13 on page 20 shows the resulting message received at the console after the start was issued.

```
S OMVS,SUB=MSTR,OMVS=00
IEF196I IEF695I START OMVS     WITH JOBNAME OMVS     IS ASSIGNED TO
IEF196I USER OMVSKERN, GROUP OMVSGRP
IEF695I START OMVS     WITH JOBNAME OMVS     IS ASSIGNED TO USER
OMVSKERN, GROUP OMVSGRP
IEF403I OMVS - STARTED - TIME=10.11.38
 BPXF013I FILE SYSTEM OMVS.ROOT 829
 WAS SUCCESSFULLY MOUNTED.
 BPXF203I DOMAIN AF_UNIX WAS SUCCESSFULLY ACTIVATED.
 BPXF203I DOMAIN AF_INET WAS SUCCESSFULLY ACTIVATED.
 BPXF024I (OMVSKERN) Oct  9 16:41:54 inetd 2424837 : login/tcp: 509
 socket: EDC5112I Resource temporarily unavailable.
 BPXI004I OMVS INITIALIZATION COMPLETE
 IEA631I OPERATOR SKOREY NOW INACTIVE, SYSTEM=IPO1, LU=SLINTM11
 BPXF024I (OMVSKERN) Oct  9 16:42:54 inetd 2424837 : login/tcp: 512
 socket: EDC5112I Resource temporarily unavailable.
 EZY2140I OpenEdition-TCP/IP connection established
```

Figure 13. MVS Console Message After Start OMVS Issued

In Figure 13 the message BPXF024I (resource temporarily unavailable) occurs. This occurs due to not establishing connectivity with TCP/IP immediately. If you continue to wait until TCP/IP is completely up and running, you will receive the EZY2140I message connection established.

3.2.6 Install OpenEdition MVS Features

The following sections describe the additional features that were installed for OpenEdition/MVS.

3.2.6.1 OpenEdition MVS Application Services (hot1130 jot13n0)

Use SMP/E to install the OpenEdition MVS Application Services. No unusual problems were encountered during this install.

References

See the following:

- *Program Directory for MVS/ESA System Product - JES2 5.2.2* for details of installing OpenEdition MVS Application Services
- *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*

3.2.6.2 OpenEdition MVS Shell and Utilities (hsu1130 jsu13n0)

Before installing OpenEdition MVS Shell and Utilities, you should consider partitioning the file system. This involves file organization, which means deciding which files are going to be on which physical DASD. The placement of files within the system hierarchy will decide the number of lookups for each file when opened.

Use SMP/E to install the OpenEdition/MVS MVS Shell and Utilities feature in the HFS file system.

References

See the following:

- *Program Directory for MVS/ESA System Product - JES2 5.2.2* for details of installing OpenEdition MVS Shell and Utilities
- *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*

3.2.6.3 OpenEdition/MVS Debugger (hdx1130)

Begin the installation of the debugger with an MVS/ESA system that has the OpenEdition MVS component and the OpenEdition MVS Shell and Utilities feature installed. Use SMP/E to install the OpenEdition MVS Debugger feature in the HFS file system.

References

See the following:

- *Program Directory for MVS/ESA System Product - JES2 5.2.2* for details of installing OpenEdition MVS Debugger
- *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*

3.2.6.4 Install Language Environment for MVS (hmwl510)

The Language Environment should be installed if users will write applications in high-level languages such as C/C++. SMP/E is used to install the Language Environment.

References

See the following:

- *Program Directory for MVS/ESA System Product - JES2 5.2.2* for details of installing The Language Environment
- *MVS/ESA SP 5.2.2 OpenEdition MVS Installation and Customization Starter Kit*

3.2.6.5 Customize Shell and Utilities

Before the Shell is invoked `/etc/inti.options` is read to initialize the value for various options. Refer to *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02 for detailed explanation of all the values. The initial values are obtained by copying the init.options file from the /samples directory to the /etc directory. The `cp /samples/init.options /etc/init.options` command can be used to accomplish the copy. Figure 14 on page 22 shows the options used to configure and run OCS.

```
 keep 10.
-a  120                                  timeout = 120 seconds
-t  1                                    terminate shell = yes
-sc /etc/rc                              shell script = /etc/rc
-e  TZ=EST5EDT                           TZ environment variable
*e  LANG=USA                             LANG environment variable
*e  NLSPATH=/usr/lib/nls/msg/%L/%N  NLSPATH environment variable
*sh /bin/sh                              shell = /bin/sh
*e  PATH=/bin                            PATH environment variable
*e  SHELL=/bin/sh                        SHELL environment variable
*e  LOGNAME=ROOT                         LOGNAME environment variable
```

Figure 14. inti.options File

The next customization should be to the /etc/rc file. The file contains customization commands for the Shell and Utilities feature. Figure 15 shows our /etc/rc file.

```
# Initialization shell script, pathname = /etc/rc

# Initial setup for OpenEdition MVS
export _BPX_JOBNAME='ETCRC'

# Setup utmpx file
>/etc/utmpx
chmod 644 /etc/utmpx

# Reset all slave tty files
chmod 666 /dev/tty*
chown 0 /dev/tty*

# Setup write, talk, mesg utilities
chgrp TTY   /bin/write
chgrp TTY   /bin/mesg
chgrp TTY   /bin/talk
chmod 2755  /bin/write
chmod 2755  /bin/mesg
chmod 2755  /bin/talk

# Invoke vi recovery
mkdir -m 777 /etc/recover
/usr/lib/exrecover

# Start the INET daemon for remote login activity
_BPX_JOBNAME='INETD' /usr/sbin/inetd  /etc/inetd.conf

LANG=En_US LC_ALL=En_US.IBM-037 /usr/sbin/ocsconfig -ac
/etc/lm.rc
echo /etc/rc script executed, date
```

Figure 15. /etc/rc File

The environmental variables and commands used by most Shell and Utilities users are placed in the /etc/profile file. Figure 16 on page 23 and Figure 17 on page 24 shows the /etc/profile as it exists in our installation.

```
if – – z "$STEPLIB" – && tty – s;
then
   echo " - - - - - - - - - - - - - - - - - - - - - - - - - - - - "
   echo " - Improve performance by preventing the propagation - "
   echo " - of TSO/E or ISPF STEPLIBs                         - "
   echo " - - - - - - - - - - - - - - - - - - - - - - - - - - - - "
   export STEPLIB=none
   exec sh -L
fi

# Set the time zone as appropriate.
TZ=EST5EDT

# This sets a default command path, including your current working
# directory (CWD).
PATH=/usr/sbin:/etc:/usr:/bin:$HOME:

# Sets the path for NLS files (message catalogs).
NLSPATH=/usr/lib/nls/msg/%L/%N

# Sets the language
# For Japanese: LANG=Ja_JP
export LC_ALL=En_US.IBM-037

LANG=En_US.IBM
# Sets the name of the system mail file and enables mail notification.
MAIL=/usr/mail/$LOGNAME

# Export the values so the system will have access to them.
export TZ PATH NLSPATH MAIL LANG

# Set the default file creation mask - reference umask in the OpenEdition
# Commands Reference
umask 022

# Set the LOGNAME variable readonly so it is not accidentally modified.
readonly LOGNAME

# = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
# Start of c89/cc customization section
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
# Version of programs and runtime library used by c89:
# = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
#
#  Compiler:
#  ---------
#  export _C89_CVERSION="0x13010000"
#  export  _CC_CVERSION="0x13010000"
#
#  Prelinker and runtime library:
#  ------------------------------
#  export _C89_PVERSION="0x11050000"
#  export  _CC_PVERSION="0x11050000"
```

Figure 16. /ETC/PROFILE file (Part 1 of 2)

```
# Program (member) names of programs called by c89:
# = = = = = = = = = = = = = = = = = = = = = = = = = = =
#
#  Compiler:
#  ---------
#  export _C89_CNAME="CBC310"
#  export  _CC_CNAME="CBC310"
#
#  Prelinker:
#  ----------
#  export _C89_PNAME="EDCPRLK"
#  export  _CC_PNAME="EDCPRLK"
#
#
# Message file (member) names used by programs called by c89:
# = = = = = = = = = = = = = = = = = = = = = = = = = = =
#
#  Compiler:
#  ---------
#  export _C89_CMSGS="EDCMSGE"
#  export  _CC_CMSGS="EDCMSGE"
#
#  Prelinker:
#  ----------
#  export _C89_PMSGS="EDCPMSGE"
#  export  _CC_PMSGS="EDCPMSGE"
#
#
# High-Level Qualifier "prefixes" for data sets used by c89:
# = = = = = = = = = = = = = = = = = = = = = = = = = = =
#
#  Compiler:
#  ---------
   export _C89_CLIB_PREFIX="CEE.V1R5M0"
#  export  _CC_CLIB_PREFIX="CBC.V3R1M0"
#
#  Prelinker and runtime library:
#  ------------------------------
    export _C89_PLIB_PREFIX="CEE.V1R5M0"
    export _CC_PLIB_PREFIX="CEE.V1R5M0"
#
#  MVS system:
#  -----------
#  export _C89_SLIB_PREFIX="SYS1"
#  export  _CC_SLIB_PREFIX="SYS1"
#
#
# Esoteric unit for data sets:
# = = = = = = = = = = = = = = = = = = = = = = = = = = =
#
#  Unit for (unnamed) work data sets:
#  ----------------------------------
#  export _C89_WORK_UNIT="SYSDA"
#  export  _CC_WORK_UNIT="SYSDA"
#
#
#
# End of c89 customization section
# = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

Figure 17. /ETC/PROFILE file (Part 2 of 2)

3.2.7 TCPIP 3.1 Plan

TCPIP 3.1 needs to be installed on the MVS/ESA 5.2.2 system. Contact your TCPIP administrator to verify installation and operation. For more information about MVS TCPIP support and OpenEdition MVS TCPIP support, refer to the following:

- *IBM TCP/IP for MVS: Customization and Administration Guide*, SC31-7134-00
- *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02

3.3 RISC System/6000 Planning

This section lists the hardware, software, and storage requirements for the OCS server running on the RISC System/6000.

3.3.1 Hardware Requirements

In addition to the hardware requirements for an OpenEdition/MVS system, you must have a RISC System/6000. OCS runs on any RISC System/6000 uniprocessor (UP) configuration.

OCS does not support the symmetric multiprocessing (SMP) hardware.

3.3.2 Channel Attachment

To link to the OpenEdition/MVS system by using a channel connection, you need the following:

- Block multiplexer (BMX) channel adapters for RISC System/6000 (feature #2755)
- Block multiplexer channel adapter cables (feature #2757)
- Block multiplexer channel interface assembly (feature #2758)
- Channel cables (bus and tag) of appropriate length (cable group #0185)

For instructions on installing and using the channel adapter, see *Block Multiplexer Channel Adapter: Users Guide and Service Information*, SC23-2427-02.

3.3.3 For LAN Attachment

To link to the OpenEdition/MVS system by using a LAN connection, or to use a channel-attached RISC System/6000 as a LAN gateway, you need at least one network interface (LAN adapter).

For instructions on installing and using the LAN adapter, see *System Management Guide: Communications and Networks*, SC23-2526-02.

3.3.4 AIX Software Requirements

To support OCS, the RISC System/6000 must run with the following software:

- AIX Version 4 Release 1.3 or a later release.
- For a channel-attached OCS, you need the block multiplexer channel interface device driver, which is an AIX Version 4.1 licensed product, 5697-037. This device driver corresponds to the CLAW device driver on the OpenEdition/MVS system.

- The AIX/ESA OCS component, which uses AIX Version 3, does not support OpenEdition/MVS host systems. Nor does the OpenEdition/MVS OCS component, which uses AIX Version 4, support AIX/ESA host systems.

3.3.5 Storage Planning

To install parts of OCS support on the RISC System/6000, you must have enough storage available in the file system. The estimated number of 512-byte storage blocks needed is:

/usr/lib/objrepos	16
/etc/drivers	800
/etc/methods	480
/usr/sbin	280
/usr/man/man8	8
/usr/man/cat8	8
/usr/lpp/msg/En_US	32
Subtotal	1624
/u/ocsmaint	4000
Total	5624

Note: The estimate for */u/ocsmaint* allows two versions of the install images (USR and SHARE packages) to exist simultaneously. The OCS USR and SHARE packages collectively require 2000 to 2100 storage blocks.

Chapter 4. Installation of OCS

This chapter explains how to install and configure OCS on both OpenEdition/MVS and the RISC System/6000.

4.1 MVS Installation for OCS

OCS is part of the OpenEdition/MVS MVS/ESA 5.2.2 system. The only thing necessary to active OCS is an OCS configuration file. We will address both creating a configuration file and the commands to maintain the file later in this chapter.

4.1.1 Maintenance Required

Before installing OpenEdition/MVS MVS, review the current Preventive Service Planning (PSP) information. Make sure that the APARs described in Appendix C, “OCS Experiences” on page 109 are applied.

4.1.2 Code Pages

A code page for a character set determines the graphic character produced for each hexadecimal code. The code page is determined by the programs and national languages being used.

The reason code pages become important in the installation of OCS is that in the editing of the OCS /etc/system file certain characters (brackets) may not translate properly. Therefore, before continuing with the installation of OCS, become familiar with code pages by reading Appendix A, “OpenEdition/MVS and OCS Code Page Translation” on page 103.

One of the most time consuming areas in the installation was understanding why the /etc/system file would not configure properly. This was due to brackets not being translated into the correct hexadecimal code. We identified the error by putting *hex on* in the `oedit` editor, and looking at the generated hex code.

Depending on the type of terminal and code pages used, you may have to use the convert parameter on the `OMVS`. Refer to *MVS/ESA OpenEdition MVS Command Reference* to determine which conversion table to use.

References

- *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02
- *MVS/ESA OpenEdition MVS Command Reference*, SC23-3014-02
- Appendix A, “OpenEdition/MVS and OCS Code Page Translation” on page 103

4.2 MVS Configuration for OCS

4.2.1 Defining Network Interfaces

The network interfaces (CLAW or LAN) must be define both to TCP/IP and OpenEdition/MVS.

4.2.2 Defining the Interface to IOCP and Performing the I/O Configuration

The definition of IOCP is covered in *OpenEdition MVS Communications Server Guide*, SC23-3883-00.

4.2.3 Defining the Interface to MVS TCP/IP

The CLAW and LAN connections must be defined to TCP/IP in the profile file. Figure 18 shows all the parameters that were added or changed in the TCP/IP profile file.

TINYDATABUFFERPOOLSIZE, as shown in Figure 18, must be defined with a sufficient number in the TCPIP profile data set. If not defined or the number is too small, OCS will not communicate properly with the AIX OCS. The error message for a missing definition or too small of a number will be seen in the TCPIP log.

For more information about configuring TCP/IP, see *IBM TCP/IP for MVS: Customization and Administration Guide,* SC31-7134-00.

```
TINYDATABUFFERPOOLSIZE      500
HOME
  9.12.1.101     TR
  9.12.4.101     CHA1
;
:
DEVICE LCS1   LCS          202
LINK TR    IBMTR    0 LCS1
:
DEVICE LCA1 CLAW 722 WTSCSL2 CHANGATE NONE 26 26 4096 4096
LINK CHA1 IP 0 LCA1
;
; ROUTE TO RISC0036 DIRECT OVER CHANNEL
  9          =       CHA1    4096      0.255.255.0 0.12.4.0
; 9.12.8      =       CHA1    4096      HOST
HOME
9          =       TR      2000      0.255.255.0 0.12.1.0
9       9.12.1.32  TR      2000      0.255.255.0 0.12.0.0
9       9.12.1.32  TR      2000      0.255.255.0 0.12.14.0
9       9.12.1.32  TR      2000      0.255.255.0 0.12.15.0
```

Figure 18. TCP/IP Parameters Added or Changed for OCS

4.2.4 Defining TCP/IP Hosts For OCS Nodes

The OCS nodes are the RISC/6000 systems that will run the OCS code. Each of these nodes must be defined to TCP/IP either by putting an entry in /etc/hosts or TCPIP.HOSTS.LOCAL files. In this project, we decided to us the /etc/hosts file, and Figure 19 shows the file used.

```
9.12.1.71  RISC71
9.12.4.102 RISC473
```

Figure 19. /etc/hosts File

The statement format in /etc/hosts file is : IP_address nodename alias ..

If you decide to use the TCPIP.HOSTS.LOCAL file, refer to *IBM TCP/IP for MVS: Customization and Administration Guide* as a guide.

4.2.5 Creating a Configuration File

Copy the file /samples/system to /etc/system. This is done by using the `cp /samples/system /etc/system` command.

Before Editing

Before editing the system file please read Appendix A, "OpenEdition/MVS and OCS Code Page Translation" on page 103. The hexadecimal representation of left and right brackets should be carefully watched.

All the nodes and associated terminals that are going to be part of OCS must be defined before starting the configuration. Figure 20 on page 30 shows our final OCS configuration. The explanation for all the parameters are in *OpenEdition MVS Communications Server Guide*. In the following section we expand and try to clarify the definitions of the parameters and their relationship to each other.

```
# = = = = = = = = = = = = =
# First stanza:  OCS object name = ocsnode1
#  = = = = = = = = = = = = =    -------------------------
ocsnode1:
    Device_Description   = "OCS1 - Channel-attach interface"
    Device_Type        = OCSRTY
    Ocs_Node_Name       = risc473
    Automatic          = yes
    MVS_Device_Char_Minor = "[0-3]"
    MVS_Device_Char_Files = "sty0[0-1]"   , \
                    "vty[2-4]"
    Ocs_Device_Char_Files = "ocstty[0-1]" , \
                    "pts/[0-1]"
#  = = = = = = = = = = = = = =
# Second stanza:  OCS object name = ocsnode2
#  = = = = = = = = = = = = =    -------------------------
ocsnode2:
    Device_Description   = "Ocsnode2 - Lan-attach interface"
    Device_Type        = OCSRTY
    Ocs_Node_Name       = risc71
    Automatic          = yes
    MVS_Device_Char_Minor = "[10-11]", "[20-21]"
    MVS_Device_Char_Files = "sty[10-11]"   , \
                    "vty[20-21]"
    Ocs_Device_Char_Files = "ocstty[0-1]" , \
                    "pts/[0-1]"
####### End of Configuration File ############################
```

Figure 20. OCS Configuration File System

4.2.5.1 Define sty and ocstty

The sty values used within MVS_Device_Char_Files definition refer to the number of serially direct attached terminals on each node. The range of numbers within the brackets must be unique for each node. No range can be duplicated with other nodes.

Stys have a direct correlation with the ocstty values in the Ocs_Device_Char_Files definition. The same number of ocsttys must be defined as stys. The range specified in both parameters do not have to be the same, but the exact numbers must agree. You can see this in Figure 20 in the definition of node 2, ocsnode2. The starting number for the range of ocsttys should be zero.

Note

A very important concept here is that the OCSTTY values are dynamically allocated at the AIX system. You must configure the OCSTTYs first at the AIX machine before you can configure the OpenEdition/MVS host machine.

4.2.5.2 Define vty and pts

The vty values in the MVS_Device_Char_Files define the number of virtual terminals that OCS will handle. Vtys have direct correlation with the pts parameter in the Ocs_Device_ Char_Files. These terminals are all the non-serial terminals, which will use rlogin or telnet to access OpenEdition/MVS MVS.

The number of vtys and ptss defined for each node should be the same and their range should be large enough to ensure that a login request from the OCS server is sucessful. The start of the range for ptss should start at zero.

4.2.5.3 Define MVS_Device_Char_Minor

This range of numbers specifies the actual number of stys and vtys defined in the MVS_Device_Char_Files parameter. This range does not have to start at zero, but if you start the range at 300 and your MAXRTYS in the BPXPRMxx member of SYS1.PARMLIB only defines 256, you will be over the maximum MAXRTYS.

4.2.6 Configuring OCS Nodes

Using the definitions above, update the /etc/system file to include all the AIX nodes and associated terminals for your configuration. Use the oedit editor to update the /etc/system, and remember the coding of left and right brackets may be a problem. Refer to Appendix A, “OpenEdition/MVS and OCS Code Page Translation” on page 103.

4.2.7 Assigning Ports to Network Daemons

Port numbers for lm (login monitor) and TBM (terminal buffer manager) must be assigned. These port numbers need to be put in TCPIP.ETC.SERVICES or /etc/services files. In our implementation we chose the TCPIP.ETC.SERVICES data set to update. Figure 21 shows parameters updated.

```
#
# OCS sockets....
#
lm          801/tcp       OCS login monitor port
tbm          802/tcp       ocs tbm port
```

Figure 21. OCS Sockets Definition

Different port numbers maybe chosen, but the same port numbers must be used in all the OpenEdition/MVSOCS and AIX OCS nodes.

4.2.8 Starting and Customizing the Login Monitor Daemon

In our installation we decided to start the lm automatically during OpenEditon MVS kernel initialization. This is done by placing the appropriate commands in the /etc/rc shell script. Figure 22 shows these commands.

```
/usr/sbin/ocsconfig -a
/etc/lm.rc
```

Figure 22. lm Startup Command in /etc/rc

The statement /etc/lm.rc executes the shell script in file /etc/lm.rc, whose contents are shown in Figure 23 on page 32.

```
export _BPX_JOBNAME=LM       # SET JOBNAME TO LM
LANG=En_US LC_ALL=C /usr/sbin/lm  2>/etc/lm.log  &
```

Figure 23. Shell Script lm.rc

4.2.9 Starting OCS

Now you can start OCS by either stopping and starting OMVS from the system console or from within the OpenEdition/MVS shell use the following commands.

- **ocsconfig** # Configure OCS support
- **/etc/lm.rc** # Start login monitor daemon

Issue the ocsconfig command in the OMVS shell or from the ishell panels. If everything was correct, a null is returned. To be sure that all nodes are configured and connected, issue the ocsconfig -q command, which queries all the OCS nodes. Figure 24 shows the response to the command.

```
-----------------
OCS query results
-----------------
Most recent ocsconfig configure/unconfigure command:
Command  : ocsconfig
UID      : 0
User name: SKOREY
Time     : Mon Oct 16 14:51:47 1995
OCS daemon state: Waiting for connections     Active nodes = 2
RTYs            : Configured RTYs = 9         Maximum RTYs = 256
OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 5         Open RTYs = 0
  State         : Not connected
OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 4         Open RTYs = 0
  State         : Not connected
```

Figure 24. Output from the ocsconfig -q Command

The output from the ocsconfig -q command shows the following information:

- Active nodes
- Maximum RTYs
- Configured RTYs
- OCS object name
- OCS node name
- TCP/IP address for the RISC/6000 node
- Configured OCS node RTYs
- State of the connection

- Number of open RTYs

The 'not connected state' will be returned because at this time the OCS for AIX has not been installed and configured. When the AIX OCS node is installed and configured, issue the ocsconfig command with the "-q" option and the state will be changed to connected.

4.3 Reconfiguring OCS Nodes

This section addresses the task of reconfiguring OCS nodes. This means adding and/or deleting terminals from a node, and also, the removal of a OCS node. Always backup your /etc/system file before doing any updating.

4.3.1 Adding and Deleting a Terminal

You can add or delete terminals to an OCS node configuration without disruption to the existing terminal sessions. In this section, we will describe the procedure to add an OCS virtual terminal (vty) to an existing configuration.

Figure 25 on page 34 shows /etc/system before any updating has been done. The first stanza is the one that will be updated; it now has four terminals defined.

```
# = = = = = = = = = = = = =
# First stanza:  OCS object name = ocsnode1
#  = = = = = = = = = = = = =    -------------------------

ocsnode1:
    Device_Description   =  "OCS1 - Channel-attach interface"
    Device_Type        = OCSRTY
    Ocs_Node_Name       = risc473
    Automatic          = yes

    MVS_Device_Char_Minor = "[0-3]"

    MVS_Device_Char_Files = "sty0[0-1]"    , \
                    "vty[2-3]"

    Ocs_Device_Char_Files = "ocstty[0-1]" , \
                    "pts/[0-1]"


#  = = = = = = = = = = = = = =
# Second stanza:  OCS object name = ocsnode2
#  = = = = = = = = = = = = = =    -------------------------

ocsnode2:
    Device_Description   =  "Ocsnode2 - Lan-attach interface"
    Device_Type        = OCSRTY
    Ocs_Node_Name       = risc71
    Automatic          = yes

    MVS_Device_Char_Minor = "[10-11]", "[20-21]"

    MVS_Device_Char_Files = "sty[10-11]"   , \
                    "vty[20-21]"

    Ocs_Device_Char_Files = "ocstty[0-1]" , \
                    "pts/[0-1]"


####### End of Configuration File ##########################
```

Figure 25. /etc/system Before Update

Figure 26 on page 35 shows the results of the `ocsconfig -q` command before adding the terminal.

```
-----------------
OCS query results
-----------------

Most recent ocsconfig configure/unconfigure command:

Command  : ocsconfig -c
UID      : 0
User name: SKOREY
Time     : Wed Oct 11 09:21:14 1995

OCS daemon state: Waiting for connections     Active nodes = 2
RTYs            : Configured RTYs = 8         Maximum RTYs = 256

OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 4         Open RTYs = 0
  State         : Connected

OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 4         Open RTYs = 0
  State         : Connected
```

Figure 26. Results of ocsconfig -q Command Before Adding a Terminal

- First increase the range of MVS_Device_Char_Minor to 0-4.
- Increase the range of MVS_Device_Char_Files vty to 2-4.
- Increase the range of OCS_Device_Char_Files pts to 0-2.
- Exit the editor. Figure 27 on page 36 shows the updated /etc/system file.

```
# ===========
# First stanza:   OCS object name = ocsnode1
# ===========    -------------------------

ocsnode1:
      Device_Description    = †OCS1 - Channel-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc473
      Automatic             = yes

      MVS_Device_Char_Minor = †[0-4]†

      MVS_Device_Char_Files = †sty0[0-1]†     , \
                              †vty[2-4]†

      Ocs_Device_Char_Files = †ocstty[0-1]†  ,  \
                              †pts/[0-2]†


# ============
# Second stanza:   OCS object name = ocsnode2
# ============    -------------------------

ocsnode2:
      Device_Description    = †Ocsnode2 - Lan-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc71
      Automatic             = yes

      MVS_Device_Char_Minor = †[10-11]†, †[20-21]†

      MVS_Device_Char_Files = †sty[10-11]†    , \
                              †vty[20-21]†

      Ocs_Device_Char_Files = †ocstty[0-1]† , \
                              †pts/[0-1]†


######## End of Configuration File ############################
```

Figure 27. /etc/system File After Update

- Enter the command ocsconfig -c to see the results execute. Then issue command ocsconfig -q to see the changes. Figure 28 on page 37 shows that there are now five configured RTYs under ocsnode1.

```
-----------------
OCS query results
-----------------

Most recent ocsconfig configure/unconfigure command:

Command  : ocsconfig
UID      : 0
User name: SKOREY
Time     : Thu Oct 12 14:18:56 1995

OCS daemon state: Waiting for connections     Active nodes = 2
RTYs            : Configured RTYs = 9         Maximum RTYs = 256

OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 5         Open RTYs = 0
  State         : Connected

OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 4         Open RTYs = 1
  State         : Connected
```

Figure 28. Results of the ocsconfig -q Command after Adding a Terminal

- Enter the command ps -ef. This will show the pid number for the lm daemon, which will be needed for the kill command. Figure 29 shows the output of the ps -ef command.

```
    UID        PID       PPID  C    STIME TTY       TIME CMD
 ALFONSO          1          0  -   Oct 09 ?        0:00
 ALFONSO      65538          1  -   Oct 09 ?       32:39
 ALFONSO      65539          1  -   Oct 09 ?        0:00 /usr/sbin/lm
 ALFONSO    2424837          1  -   Oct 09 ?        0:00 /usr/sbin/inetd /etc/
 ALFONSO    3538950          1  - 16:34:28 ?        0:00
 ALFONSO    4128775    8192008  - 16:35:42 ttyp0000  0:00 ps -ef
 ALFONSO    8192008    3538950  - 16:34:31 ttyp0000  0:02 sh -L
```

Figure 29. Output from the ps -ef Command

- The last thing to be done is to notify lm(login monitor) that the OCS configuration has been changed. This is done by issuing the kill -HUP 65539 command. The number 65539 is the pid number; refer to Figure 29.

Now the OCS node has been updated and lm has been informed of the changes, the next thing to do is update the AIX OCS configuration on the appropriate node.

4.3.2 Deleting a Node from the OCS Configuration

Figure 25 on page 34 shows the original /etc/system file that will be used. Deleting a node requires that a stanza be deleted from /etc/system. In our example stanza two will be deleted from the OCS configuration.

ATTENTION

There is one anomaly. The ocsconfig command uses the configuration information in the /etc/system file to determine what OCS nodes can be configured and unconfigured. If you configure an OCS node and subsequently remove the stanza from the /etc/system file that defined the OCS node, then you will not be able to unconfigure (or query) that OCS node.

This example uses the /etc/system file, as shown in Figure 25 on page 34. The following is the sequence of steps to follow to delete a OCS node.

- Backup the /etc/system file.
- Issue the ocsconfig -u ocsnode2 command. The ocsconfig -q command will show that ocsnode2 is unconfigured; refer to Figure 30.

```
-----------------
OCS query results
-----------------

Most recent ocsconfig configure/unconfigure command:

Command  : ocsconfig -u -- ocsnode2
UID      : 0
User name: SKOREY
Time     : Wed Oct 11 11:35:47 1995

OCS daemon state: Waiting for connections     Active nodes = 1
RTYs            : Configured RTYs = 5         Maximum RTYs = 256

OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 5         Open RTYs = 0
  State         : Connected

OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 0         Open RTYs = 0
  State         : Not configured
```

Figure 30. ocsconfig -q Showing ocsnode2 Unconfigured

- Update /etc/system by deleting stanza two.
- Issue the ocsconfig -c command to remove ocsnode2 from the configuration. The output of ocsconfig -q, seen in Figure 31 on page 39, shows that ocsnode2 is no longer part of the configuration.

```
-----------------
OCS query results
-----------------

Most recent ocsconfig configure/unconfigure command:

Command  : ocsconfig -c
UID      : 0
User name: SKOREY
Time     : Thu Oct 12 15:58:58 1995

OCS daemon state: Waiting for connections     Active nodes = 1
RTYs            : Configured RTYs = 5         Maximum RTYs = 256

OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 5         Open RTYs = 0
  State         : Connected
```

Figure 31. ocsconfig -q Showing Only One Node Configured

- Issue the ps -ef command to determine the pid number for lm. Refer to Figure 29 on page 37
- Issue the kill -HUP (pid #) to inform lm of the update configuration.

4.4 AIX/6000 Installation for OCS

This section explains how to install and configure OCS on the RISC System/6000. Before you install you should:

- Insure that you have the required hardware/software listed in 1.3, “Minimum System Requirements” on page 2
- Complete the steps for configuring MVS starting at 4.2, “MVS Configuration for OCS” on page 28

4.4.1 Installing OCS

You installed the OCS support for the AIX OCS host when you successfully installed the OpenEdition/MVS component. The OCS installp images are located in the following datasets:

- **SYS1.SFOMDATA(FOMC6000)** - This dataset contains the USR part of the OCS product. The USR part contains all of the OCS executable code and AIX Object Data Manager (ODM) entries required to configure and use OCS.
- **SYS1.SFOMDATA(FOMC6010)** - This dataset contains the SHARE part of the OCS product. The SHARE part contains the man (manual) pages for OCS.

To install OCS on the RISC System/6000, you must:

- Create a installp image directory on the OCS host to receive the image files. In the example shown in Figure 32 on page 40, we created via the mkdir command the ¢/u/ocsmaint¢ directory for this purpose.
- Transfer the OCS installp images to the RISC System/6000. One method is to use the TCP/IP File Transfer Protocol (FTP) protocol. You may also use Network File Server (NFS) and NFS-mount the installp images and point the Systems Management Interface Tool (SMIT) to them.

- Read the README file that is packaged with each installp image to see if any changes have been made to the installation procedures.
- When installing the images, install the USR part before installing the SHARE part.

4.4.2 Transferring the Installp Images

1. On the AIX system, enter the `cd` command to change to the /u/ocsmaint directory.
2. Then enter the `ftp` command to connect to the OpenEdition MVS host. In the example in Figure 32, we used the IP address of 9.12.1.101. This is the address of our OpenEdition MVS host on our MVS system ¢wtscsl2¢.

   You could also use the symbolic

   `host_name`

   as well, assuming that you either perform name resolution on a ¢Name Server¢ or by the */etc/hosts/* file on your AIX machine.
3. Enter your TSO user ID and password when prompted.
4. Enter the `binary` command at the `ftp` prompt.
5. Enter the `get` command for the following installp images:

```
get ¢SYS1.SFOMDATA(FOMC6000)¢ /u/ocsmaint/fomc6000
get ¢SYS1.SFOMDATA(FOMC6010)¢ /u/ocsmaint/fomc6010
```

Attention

It is important to use correct case sensitivity with AIX. Notice that the installp images on the OpenEdition MVS host are uppercase and the `get` command and the AIX directory and file names are in lowercase. Also note the single quotes around the OpenEdition MVS file names.

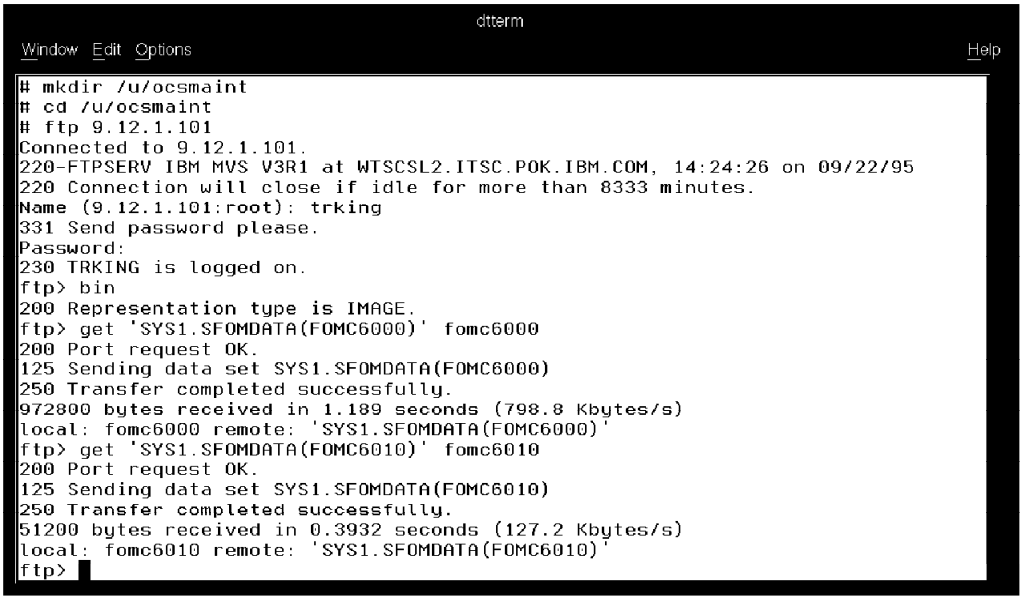


Figure 32. Sample FTP Dialog

4.4.3 Accessing the README File in the Install Images

To access the README file during the installp process do the following:

- Use the `-i` option with `installp` command

  ```
  installp -id /u/ocsmaint/fomc6000 all
  ```

4.4.4 Installing the OCS Installp Images on the RISC System/6000

There are two ways to install the OCS code on the RISC System/6000 OCS host. In our examples we decided on using SMIT, because you will use SMIT to configure OCS after installation.

If you chose the other way, which is the AIX command line method, issue the following commands to install the OCS code:

```
installp -qaXd /u/ocsmaint/fomc6000 all
```

After completion of the first command, then;

```
installp -qaXd /u/ocsmaint/fomc6010 all
```

4.4.5 Installing with SMIT

To install the OCS USR and SHARE images follow these steps.

1. Invoke the Systems Management Interface Tool (SMIT) by entering `smit` at the AIX command prompt. The ¢SMIT¢ main menu appears.

   Note

   The actual SMIT panels you see may vary from the panels shown here.

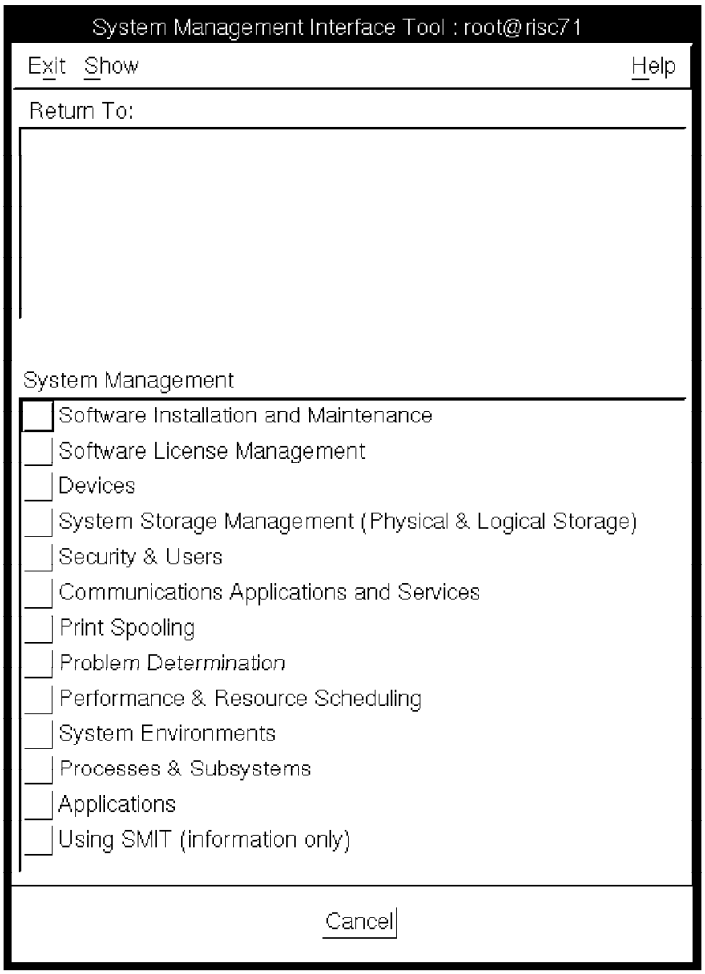


Figure 33. SMIT Main Menu Panel

2. Select **Software Installation and Maintenance**.

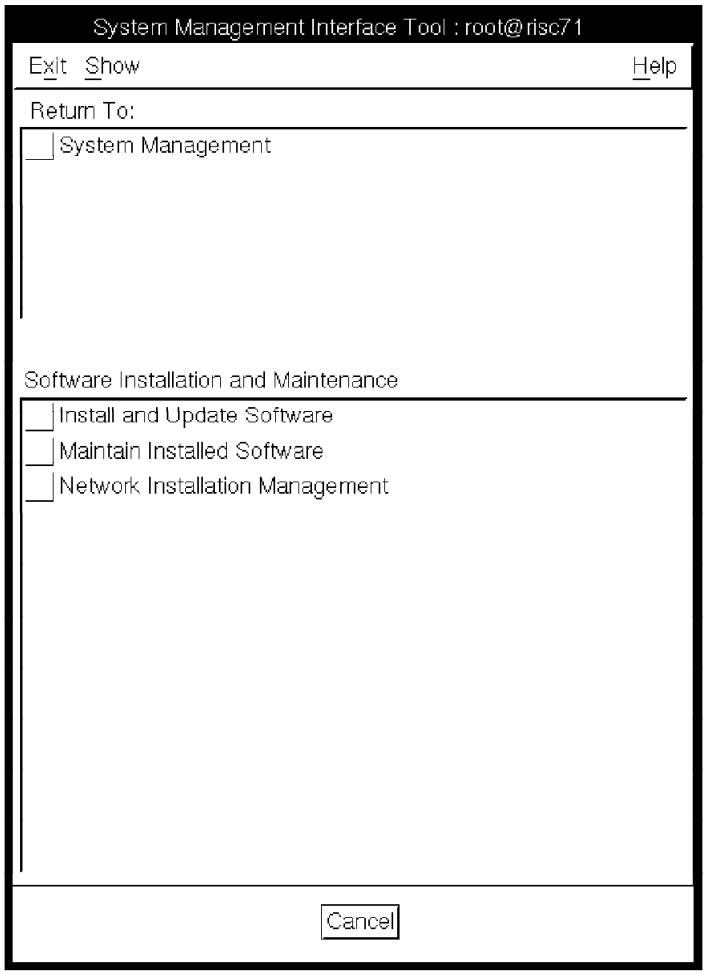


Figure 34. SMIT Software Installation and Maintenance Panel

3. Select **Install and Update Software**.

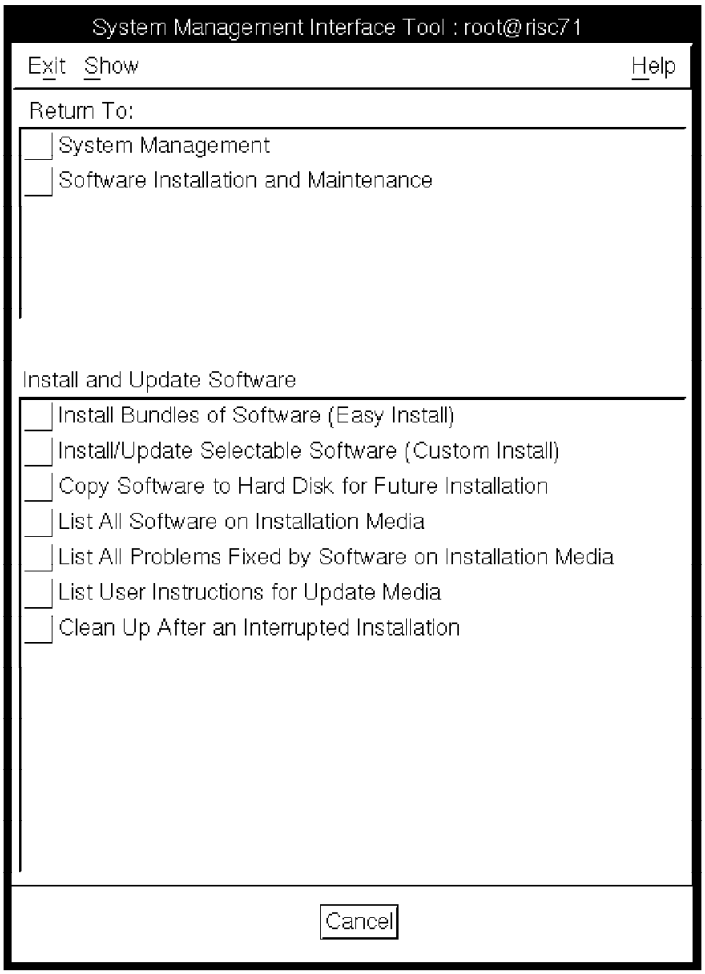


Figure 35. SMIT Install and Update Software Panel

4. Select **Install/Update Selectable Software (Custom Install)**.

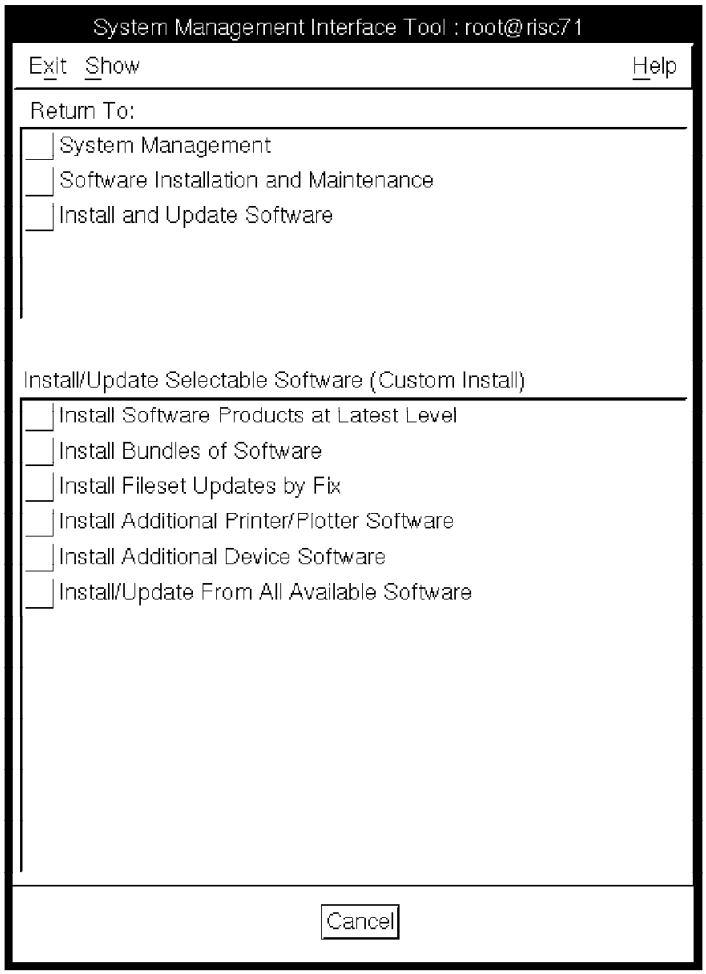


Figure 36. SMIT Install/Update Selectable Software (Custom Install) Panel

5. Select **Install Software Products at Latest Level**.

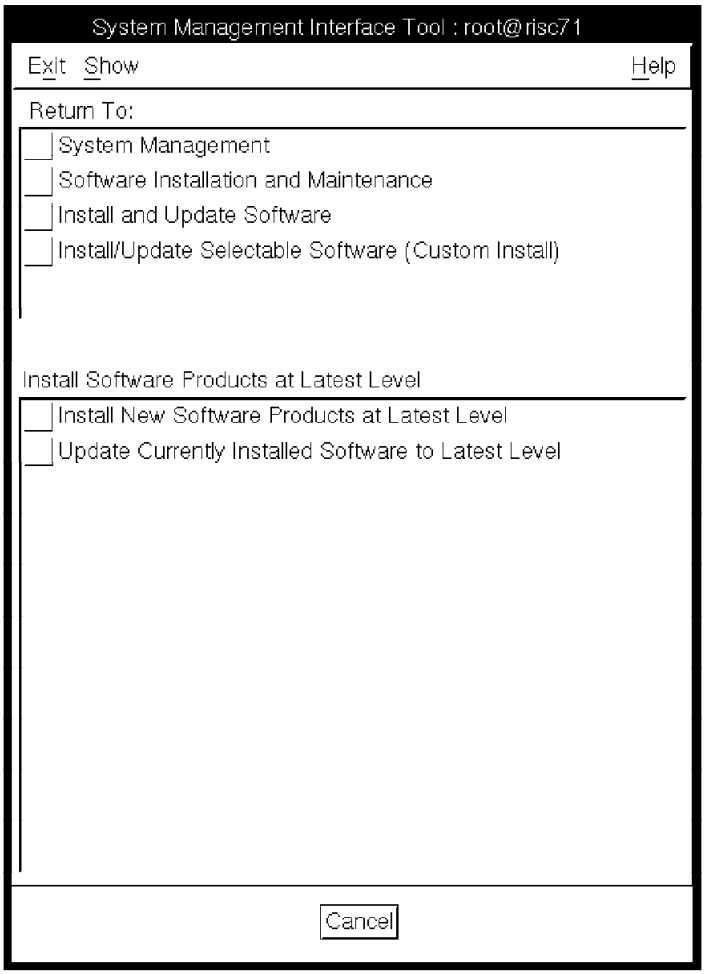


Figure 37. SMIT Install Software Products at the Latest Level Panel

6. Select **Install New Software Products at Latest Level**.

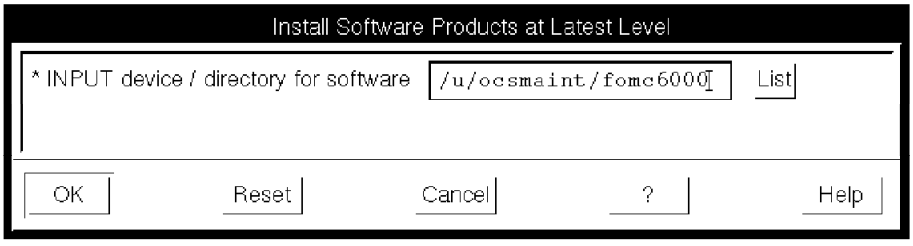


Figure 38. SMIT Install New Software Products at Latest Level Panel

7. Enter `/u/ocsmaint/fomc6000` in the input device field and select **OK**.

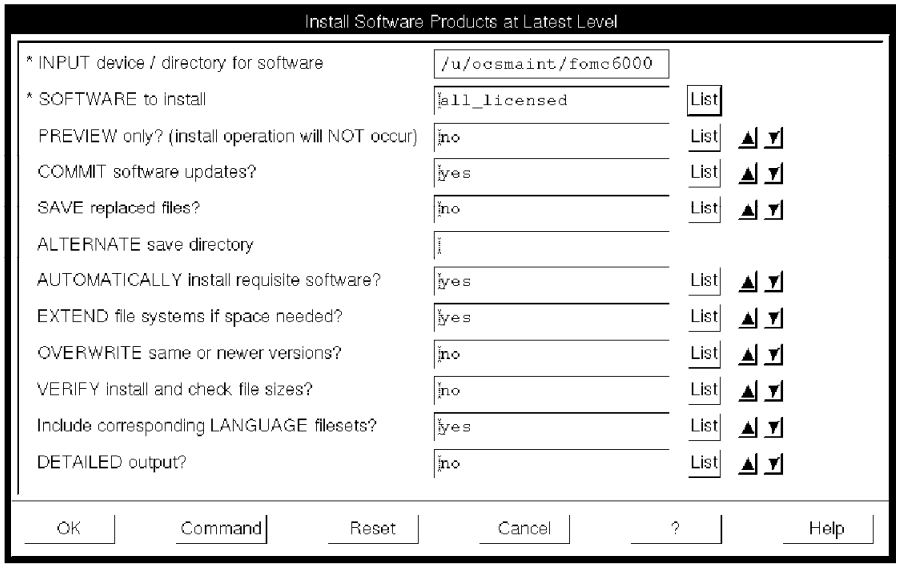


Figure 39. SMIT Install New Software Products at Latest Level Panel (Cont.)

8. Accept the defaults along with the input device for the installp images. Select **OK**.

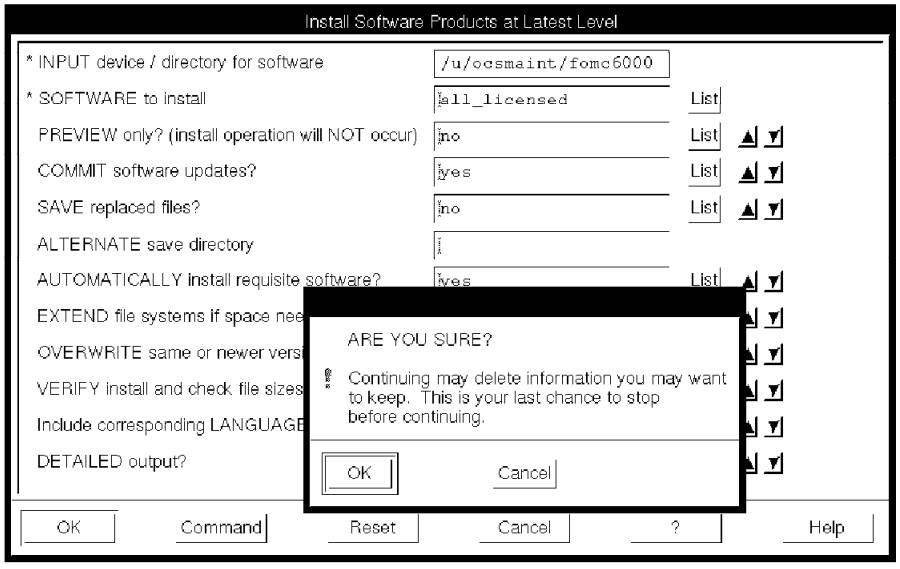


Figure 40. SMIT Install New Software Products at Latest Level Panel (Cont.)

9. Select **OK** to continue the install process.

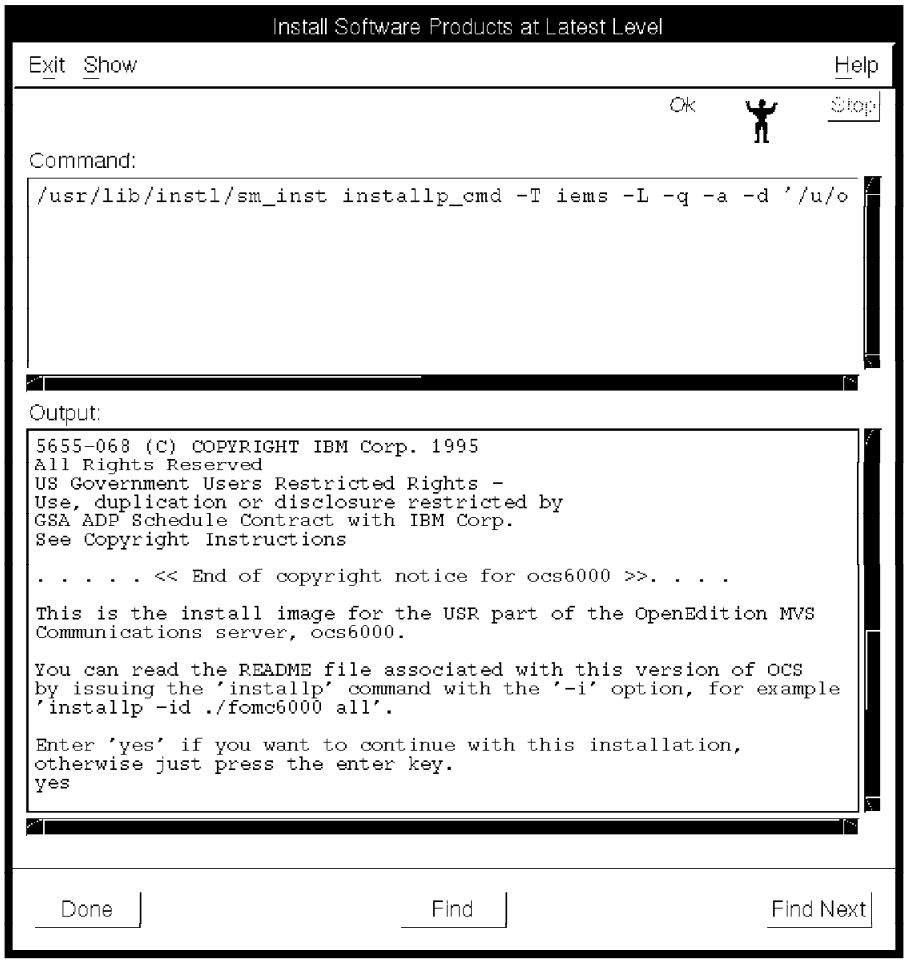


Figure 41. SMIT Install New Software Products at Latest Level Panel (Cont.)

10. Install process continues and will prompt you to read the README file.

    If you decide that you want to read the README file, enter no here and the install will be terminated. After you read the README information you would then restart this installation procedure.

    At this point we will enter, yes and continue with the installation of OCS.

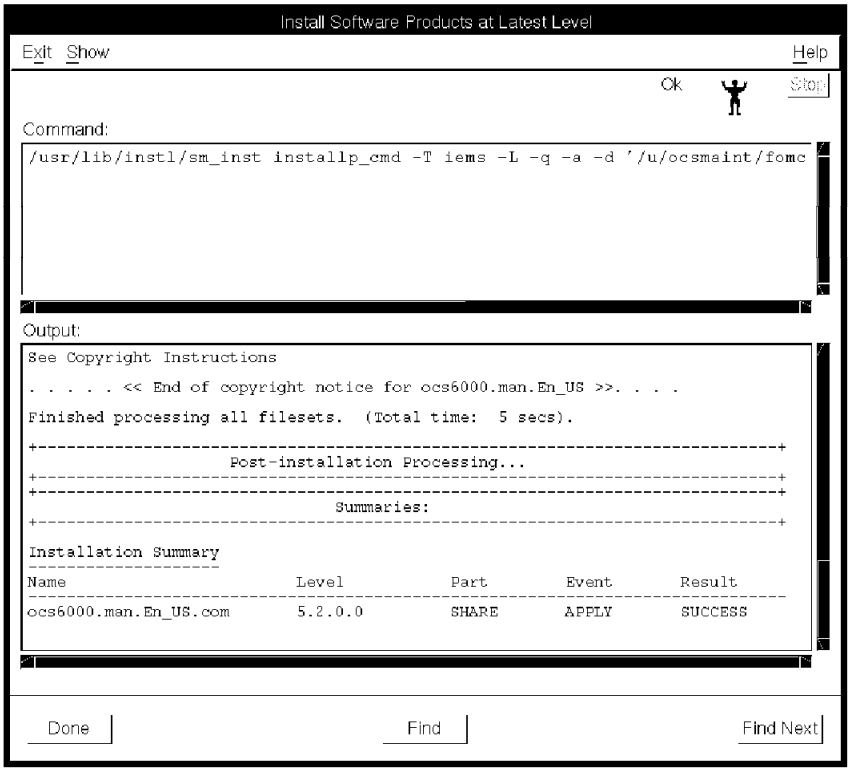


Figure 42. SMIT Install New Software Products at Latest Level Panel

11. Installation continues and ends successfully with the Ok status.
12. Select **Done** and return to Figure 39 on page 47. Then select **Cancel**, to return to Figure 37 on page 46 and continue with the installation for the SHARE part of OCS install.
13. Repeat steps 6 through 11 to install the SHARE part of OCS. Enter the input device field with `/u/ocsmaint/fomc6010`, as shown in Figure 38 on page 46 and continue until you see the panel shown in Figure 42, at which time the installation of the OCS code is completed. You may exit SMIT at this time.

4.4.6 Servicing OCS on the RISC System/6000

Updates for the OCS installp images are loaded on the OpenEdition MVS system using normal MVS service procedures. The update images contain cumulative fixes of all the proceeding fixes, so you only need to install the most recent update image to be at the most current level.

Dataset SYS1.SFOMDATA(FOMC600P) contains the most recent OCS update for the USR part, while the dataset SYS1.SFOMDATA(FOMC601P) contains the most recent OCS update for the SHARE part of OCS. Use the same steps, Figure 33 on page 42 through Figure 42, to install the updates as you did to install the base code.

4.5 AIX/6000 Configuration for OCS

This section explains how to configure AIX 4.1.3 for OCS. The following topics are covered:

- Defining network interfaces
- Assigning internet addresses
- Defining host nodes
- Defining terminals
- Assigning a port to network daemons
- Starting the Terminal Solicitor

4.5.1 Defining Network Interfaces on the RISC/6000

The AIX system administrator needs to define the connections to the OpenEdition MVS host and the OCS terminals. These definitions are controlled by the ODM and are defined via SMIT. Further configuration to the OCS host will rely on knowing the type of connections and their associated IP addresses.

For a channel-attached OCS, use SMIT to define the connection between the OpenEdition MVS host and the OCS server end. See *Block Multiplexer Channel Adapter: Users Guide and Service Information*, SC23-2427-02 for more information.

For LAN-attached OCS, again use SMIT to define the connection between the OpenEdition/MVS host and the OCS server end. See *System Management Guide: Communications and Networks*, SC23-2526-02 for more information.

Some of the values that you specify via SMIT are influenced by related values used on the OpenEdition/MVS host:

- **Transmit Buffer Size** - Must be less than or equal to the size of the read buffer used by the host.

  The MVS TCP/IP read buffer size is specified on the DEVICE statement for the CLAW connection in the PROFILE.TCPIP dataset. The default read buffer size is 4096.

- **Receive Buffer Size** - Must be less than or equal to the size of the write buffer used by the host.

  The MVS TCP/IP write buffer size is specified on the DEVICE statement for the CLAW connection in the PROFILE.TCPIP dataset. The default write buffer size is 4096.

- **CLAW Mode Host Name** - Must match the host named assigned on the OpenEdition/MVS host.

  This is the value of the host_claw_name parameter on the DEVICE statement for the CLAW connection in the PROFILE.TCPIP dataset. For example, in Figure 18 on page 28, the host_claw_name is MVS4101.

- **CLAW Mode Subchannel Set** - Must correspond to the value of the address parameter on the DEVICE statement for the CLAW connection in the PROFILE.TCPIP dataset. For example in, Figure 18 on page 28, the address is 722.

4.5.2 Assigning Internet Addresses

You must assign a primary internet address for each interface of the OCS network. Use either the `ifconfig` command or SMIT. SMIT will invoke the same `ifconfig` commands to assign the internet addresses.

To use SMIT:

1. Invoke the Systems Management Interface Tool (SMIT) by entering `smit` at the AIX command prompt. The SMIT main menu appears.

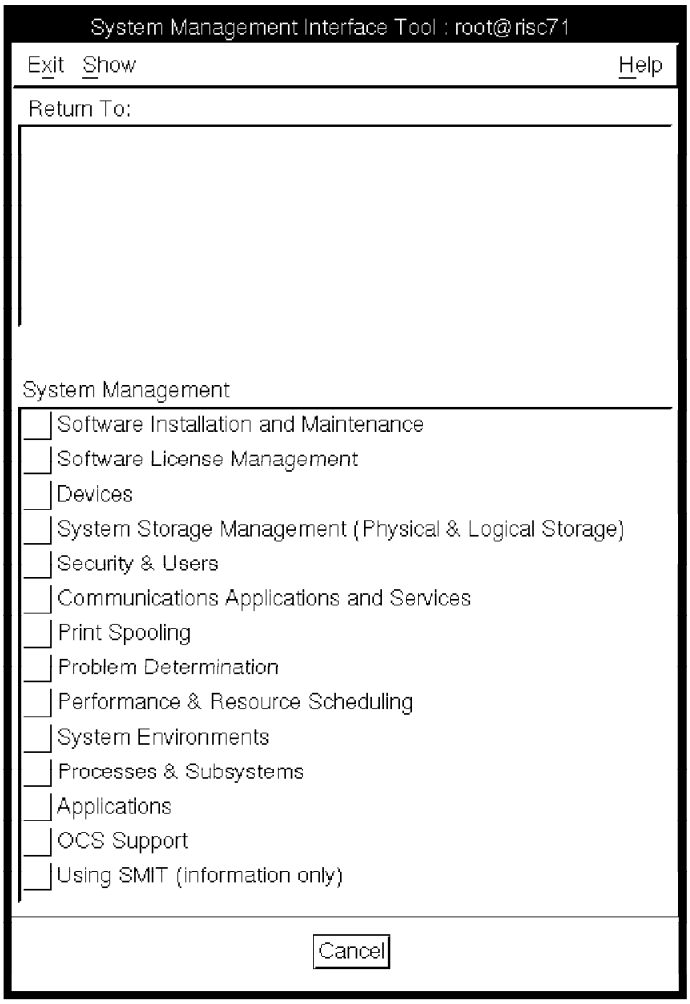


Figure 43. SMIT Main Menu Panel

2. Select **Communications Applications and Services**.

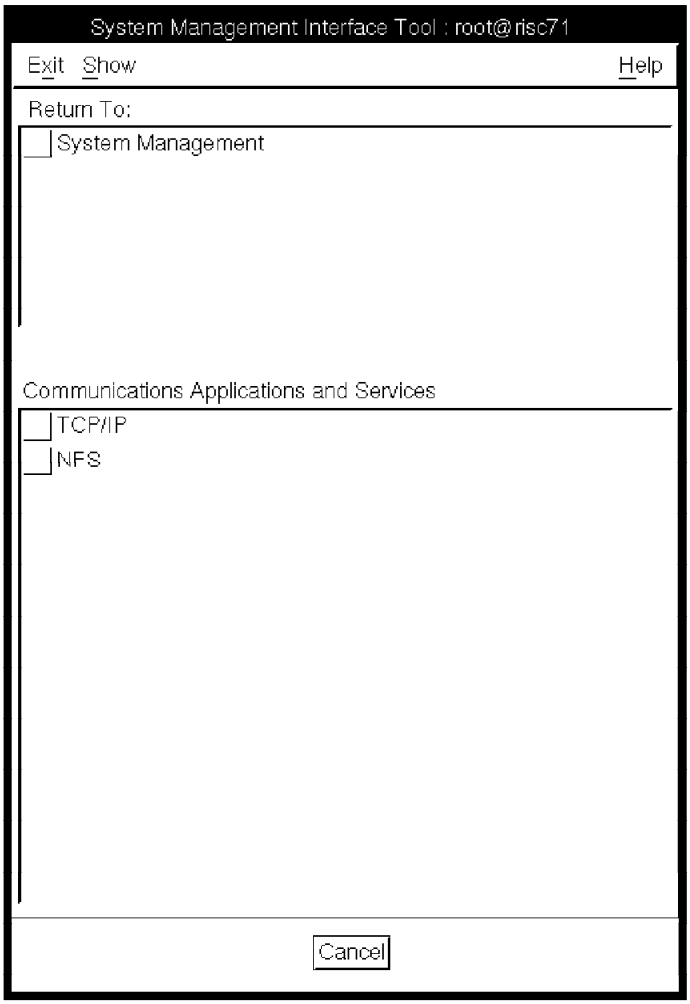


Figure 44. SMIT Communications and Services Panel

3. Select **TCP/IP**.

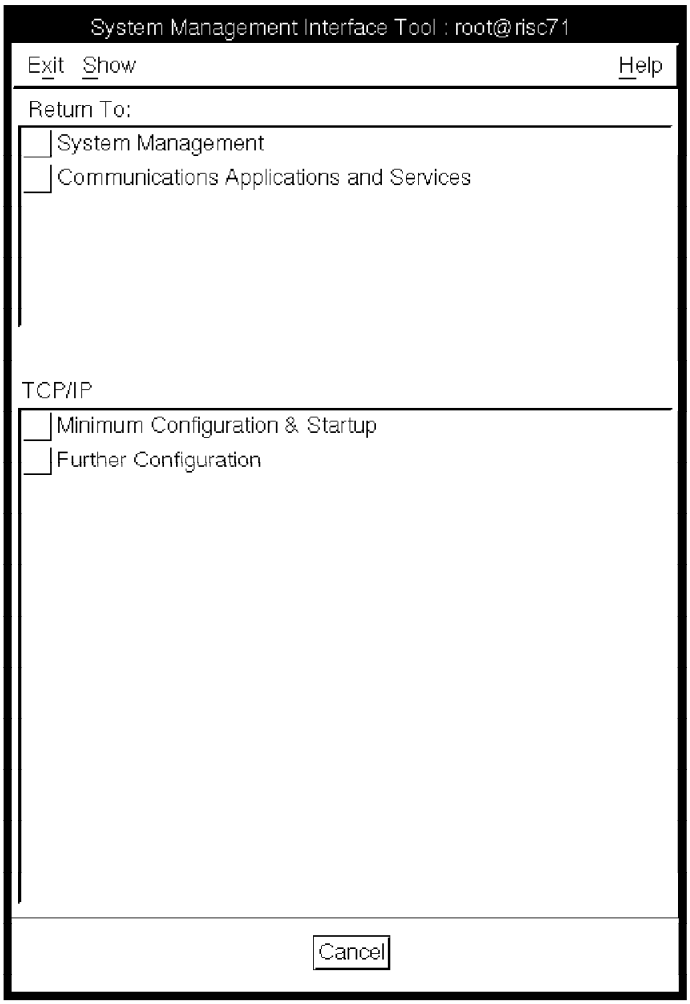


Figure 45. SMIT TCP/IP Panel

4. Select **Minimum Configuration & Startup**.

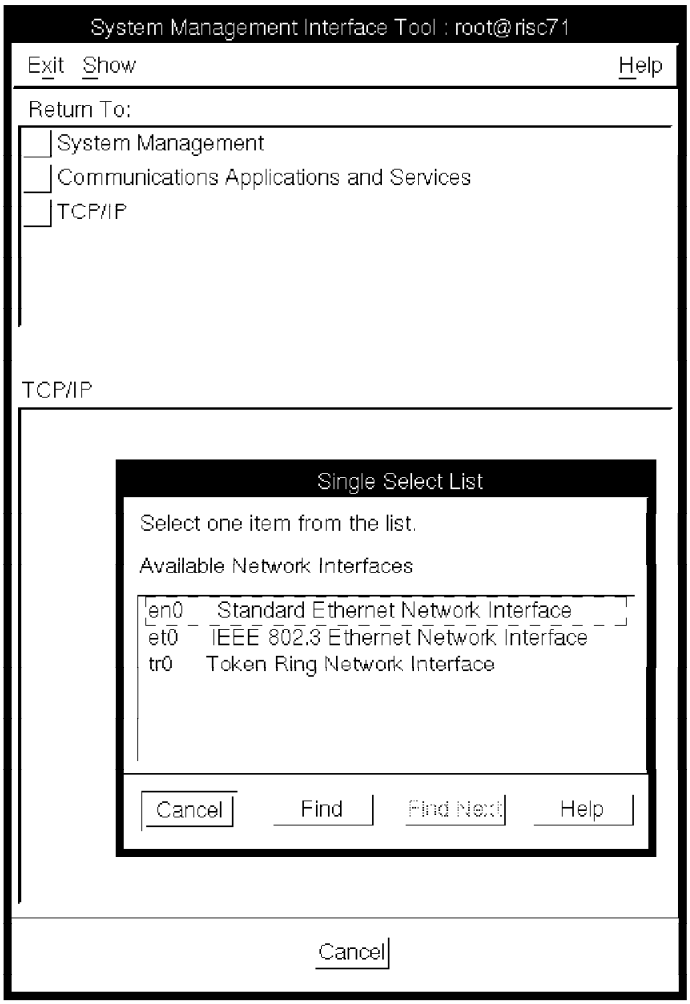


Figure 46. SMIT Single Selection List for Token-Ring Interface Panel

5. If configuring the token-ring interface, continue by selecting **tr0**. If configuring a CLAW interface, skip to step 10 after Figure 52 on page 58.

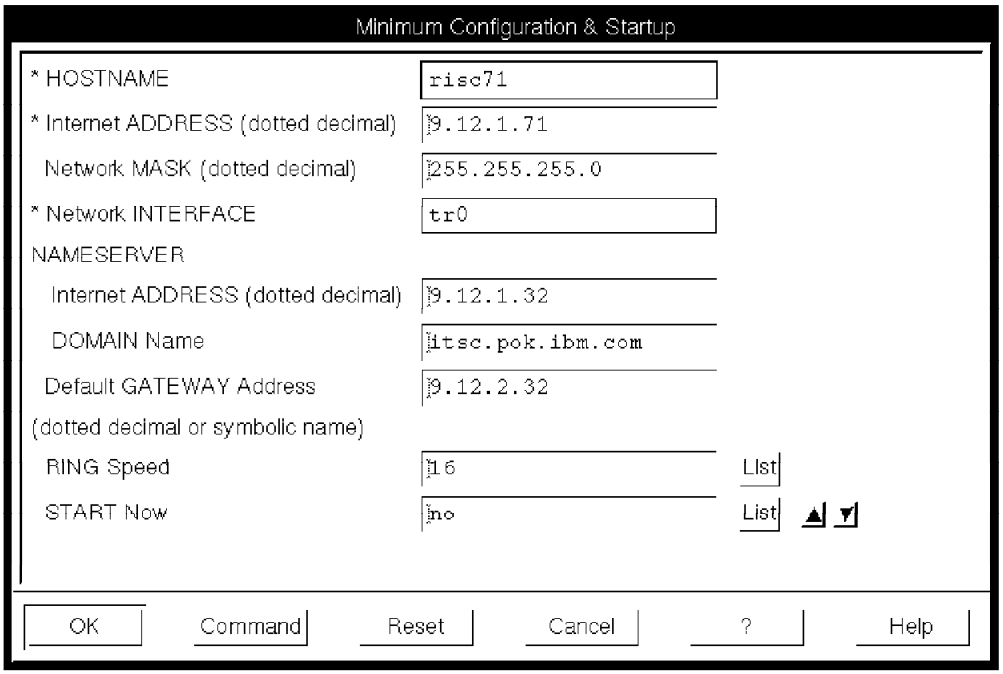


Figure 47. SMIT Minimum Configuration & Startup Panel for Token-Ring Interface

6. Fill in the information requested. For example in Figure 47, we coded `risc71` as our OCS HOSTNAME. The internet address for `risc71` is `9.12.1.71` and the network mask is `255.255.255.0`.

   We also provided the name server information here as well.

7. After completing the information, select **OK**.

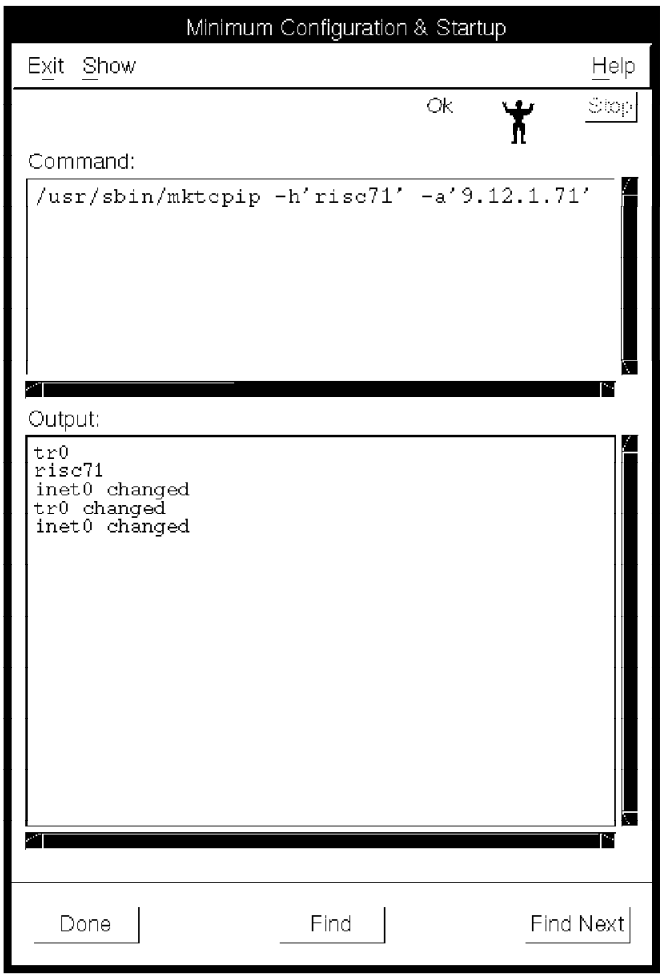


Figure 48. SMIT Completion for Token-Ring Interface Panel

8. This completes the assigning of the internet addresses for the LAN-attached OCS host.

   Select **Done** and return to Figure 47 on page 55. Then select **Cancel** to return to Figure 45 on page 53. Then select **Exit** to get out of SMIT and continue with the next step.

9. If you want to use the OCS `telnet` or `rlogin` virtual terminal protocols to establish sessions with the OpenEdition/MVS host, then you must define one or more virtual host addresses. You will need a unique alias IP address for each virtual host address that you want to define.

   Use the `ifconfig` command to assign alias addresses. To avoid routing table changes, assign alias addresses on the same network as the primary address. In the example, Figure 49 on page 57, we added the alias address of `9.12.1.253` for our LAN-attached OCS.

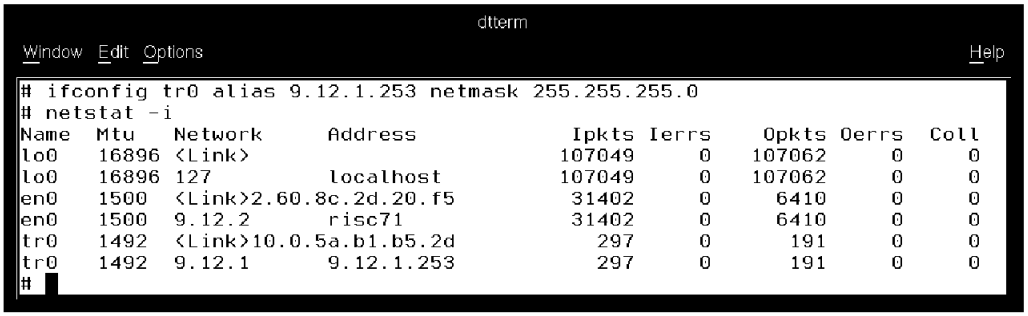


Figure 49. Sample ifconfig and netstat Commands on RISC71

You also need to add the ifconfig command to the /etc/rc.net file to maintain the alias addresses across subsequent AIX system initializations. We added the following to the bottom of the /etc/rc.net file:

```
/usr/sbin/ifconfig tr0 alias 9.12.1.253 netmask 255.255.255.0 >>$LOGFILE 2>&1
```

Attention

Add the ifconfig command at the bottom of the /etc/rc.net file. Do not attempt to use the commented out commands in section two. These commands are examples of the equivalent traditional commands used to perform the same function, but the commands in section two do ***not*** use the ODM database.

Adding the alias address to hostname RISC71 will allow virtual (non-serial attached) terminals to rlogin or telnet only to the OpenEdition/MVS host wtscsl2. You will need to add routing and name resolution information to RISC71 for virtual terminals to gain access to RISC73 (CLAW-attached OCS server). This was accomplished in the following two steps:

a. For routing capability:

   In Figure 50, we added the following network routing via the route add command and also issued the netstat -rn command to verify our net add. This resulted in the following display:

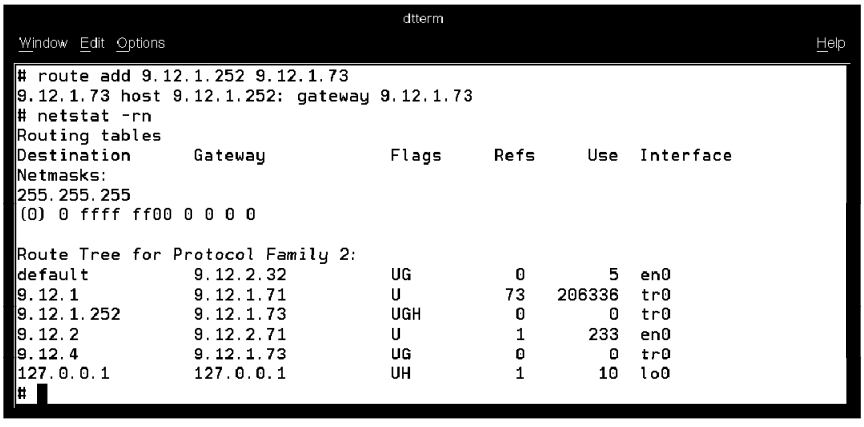


Figure 50. Route Add and Result from the netstat -rn Command on RISC71

b. For name resolution:

   In Figure 51 on page 58, we edited the /etc/hosts file on RISC71 to resolve the names for both the LAN and CLAW-attached

OpenEdition/MVS systems. The following figure shows the results of editing the /etc/hosts file:

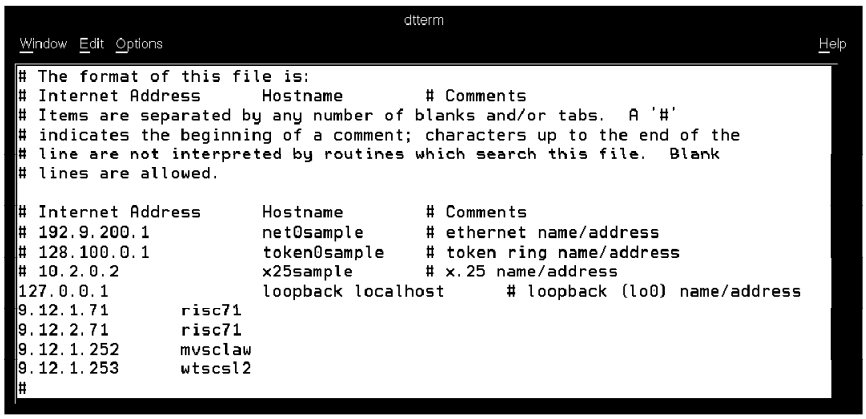


Figure 51. Cat Output of the /etc/host File on RISC71

If using a named server in your network, you will need to define the proper name on your name server. See *System Management Guide: Communications and Networks*, SC23-2526-02 for more information on name server.

This completes this section for the LAN-attached OCS server. Go to the next section, 4.5.3, “Defining Host Nodes on the RISC/6000” on page 62.

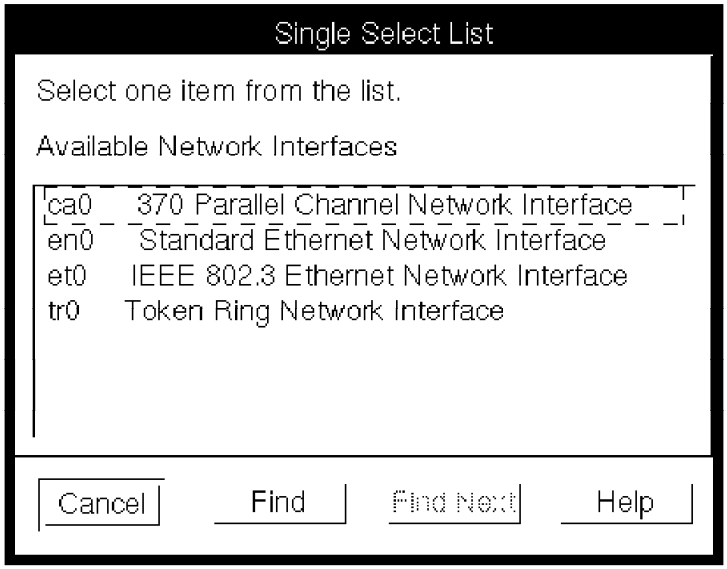


Figure 52. SMIT Single Selection List for CLAW Interface Panel

10. For the CLAW interface, select **ca0**.

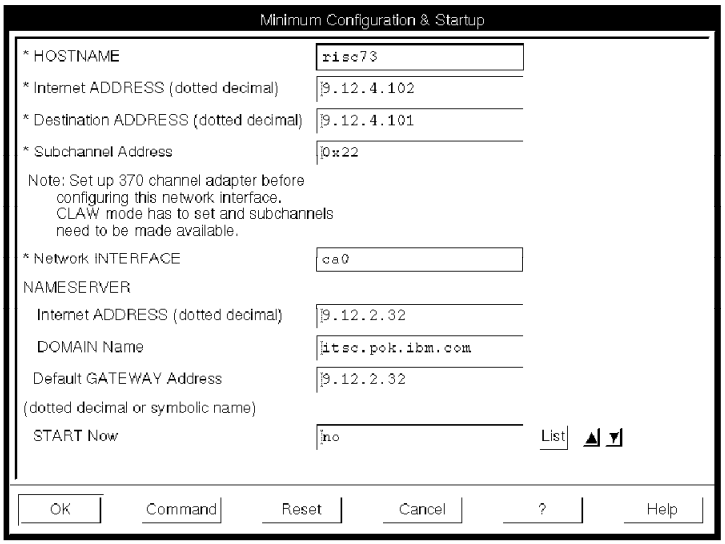


Figure 53. SMIT Minimum Configuration & Startup Panel for CLAW Interface Panel

11. Fill in the information requested. In Figure 53, we coded `risc73` as our CLAW-attached OCS HOSTNAME.

    Two internet addresses are required for `risc73`:

    - `9.12.4.102` is the internet address that is assigned to the AIX end of the CLAW.
    - `9.12.4.101` is the internet address that is assigned to the MVS channel end of the CLAW.

    `0x22` is the subchannel address that was assigned during the CLAW TCP/IP configuration in 4.5.1, “Defining Network Interfaces on the RISC/6000” on page 50, using the information from *Block Multiplexer Channel Adapter: Users Guide and Service Information*, SC23-2427-02.

    Again we also provided the name server information here as well.

    After completing the information, select **OK**.

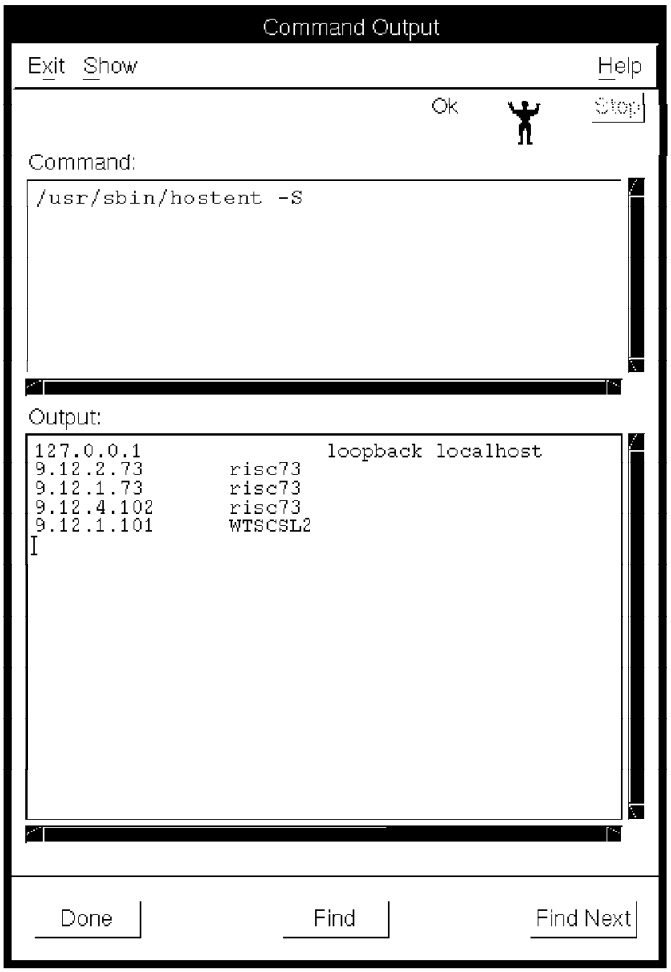


Figure 54. SMIT Completion for CLAW Interface Panel

12. This completes the assigning of the internet addresses for the CLAW-attached OCS host.

    Select **Done** and return to Figure 53 on page 59. Then select **Cancel** to return to Figure 45 on page 53. Then select **Exit** to get out of SMIT and continue with the next step.

13. If you want to use the OCS `telnet` or `rlogin` virtual terminal protocols to establish sessions with the OpenEdition/MVS host, then you must define one or more virtual host addresses. You will need a unique alias IP address for each virtual host address that you want to define.

    Do not assign an alias address to a channel interface. This means you can not use the `ca0` parameter on the `ifconfig` command. The end-user workstations are either token-ring or EtherNet attached, and the alias addresses are only associated with those network interfaces. On the CLAW-attached OCS server, we assigned the alias address of 9.12.1.252 to the tr0 interface. This was done the same as in step 9. See Figure 49 on page 57. We also added the following statement to the /etc/rc.net file:

    ```
    /usr/sbin/ifconfig tr0 alias 9.12.1.253 netmask 255.255.255.0 >>$LOGFILE 2>&1
    ```

    Adding the alias address to hostname RISC73 will allow virtual terminals to `rlogin` or `telnet` only to the OpenEdition/MVS host wtscsl2. You will need to add routing and name resolution information to RISC73 for virtual terminals

to gain access to RISC71 (LAN-attached OCS server). This was accomplished in the following two steps:

a. For routing capability:

   In Figure 55, we added the following network routing via the route add command and also issued the netstat -rn command to verify our net add. This resulted in the following display:

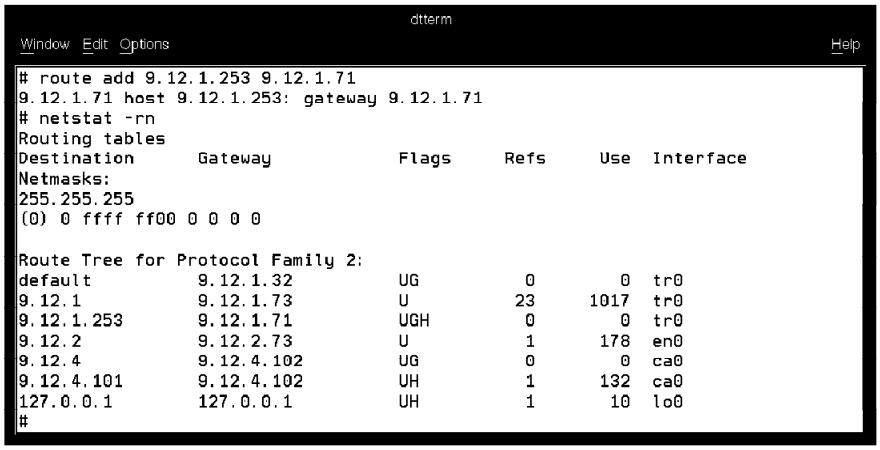


Figure 55. Route Add and Result from netstat -rn Command on RISC73

b. For name resolution:

   In Figure 56, we edited the /etc/hosts file on RISC73 to resolve the names for both the LAN and CLAW-attached OpenEdition/MVS systems. The following figure shows the results of editing the /etc/hosts file:

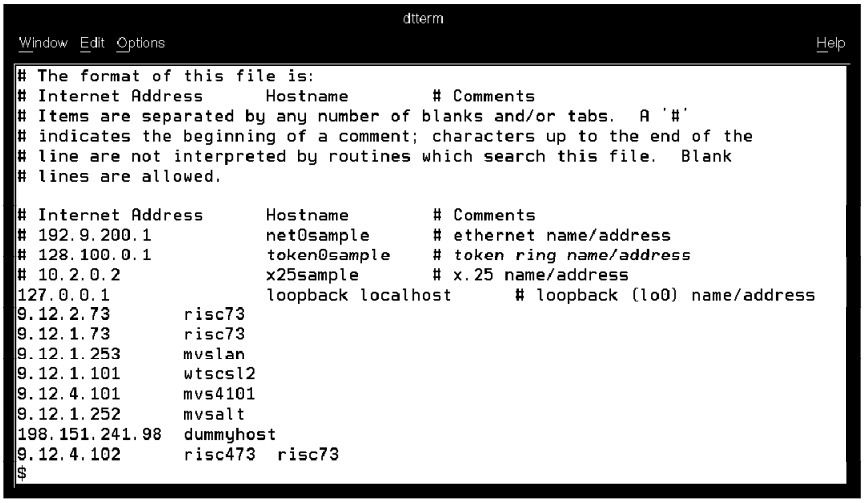


Figure 56. Cat Output of the /etc/host File on RISC73

If using a named server in your network, you will need to define the proper name on your name server. See *System Management Guide: Communications and Networks*, SC23-2526-02 for more information on name server.

This completes this section for the CLAW-attached OCS server. Continue on with the next section.

4.5.3 Defining Host Nodes on the RISC/6000

To define the OpenEdition/MVS host node to the AIX systems, use the following steps via SMIT to update the /etc/hosts file.

1. Repeat steps 1 through 3 from 4.5.2, “Assigning Internet Addresses” on page 51 and return here.
2. Select **Further Configuration**.

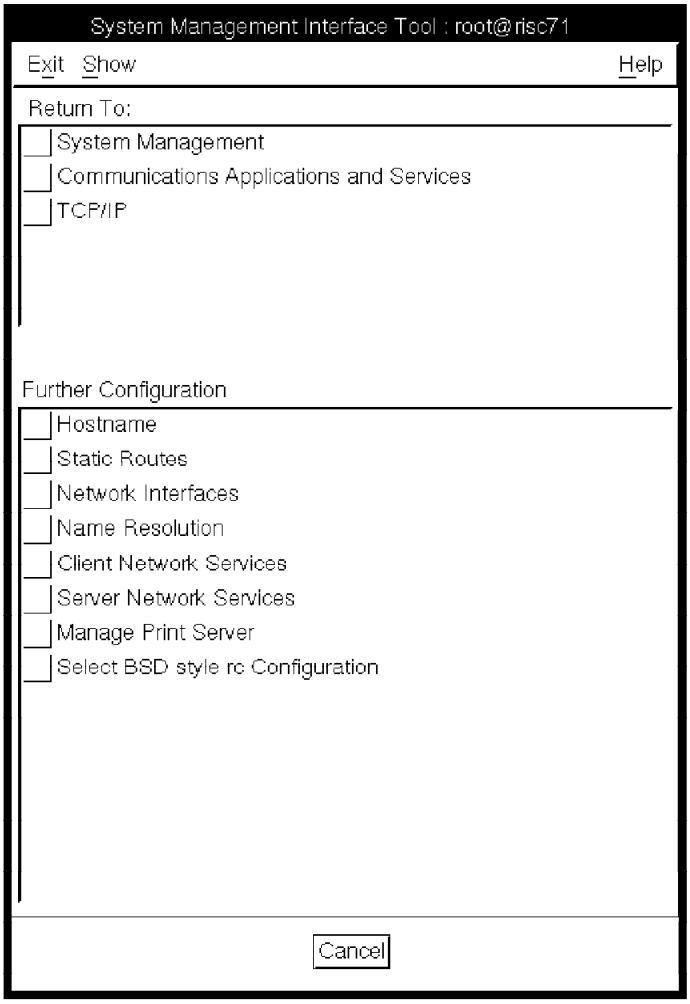


Figure 57. SMIT Further Configuration Panel

3. Select **Name Resolution**.

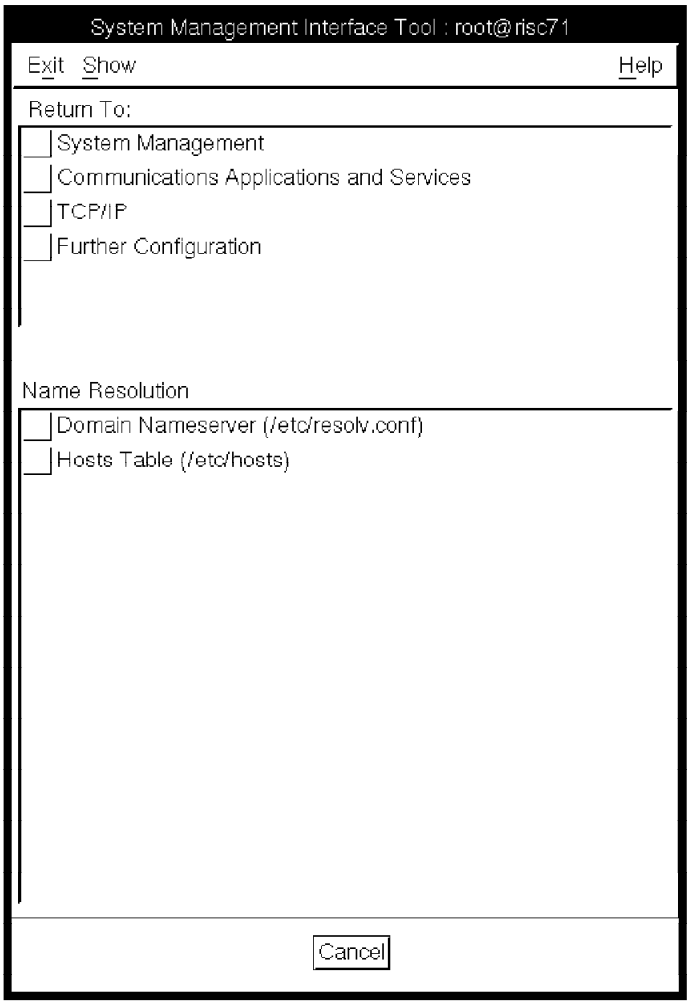


Figure 58. SMIT Name Resolution Panel

4. Select **Hosts Table (/etc/hosts)**.

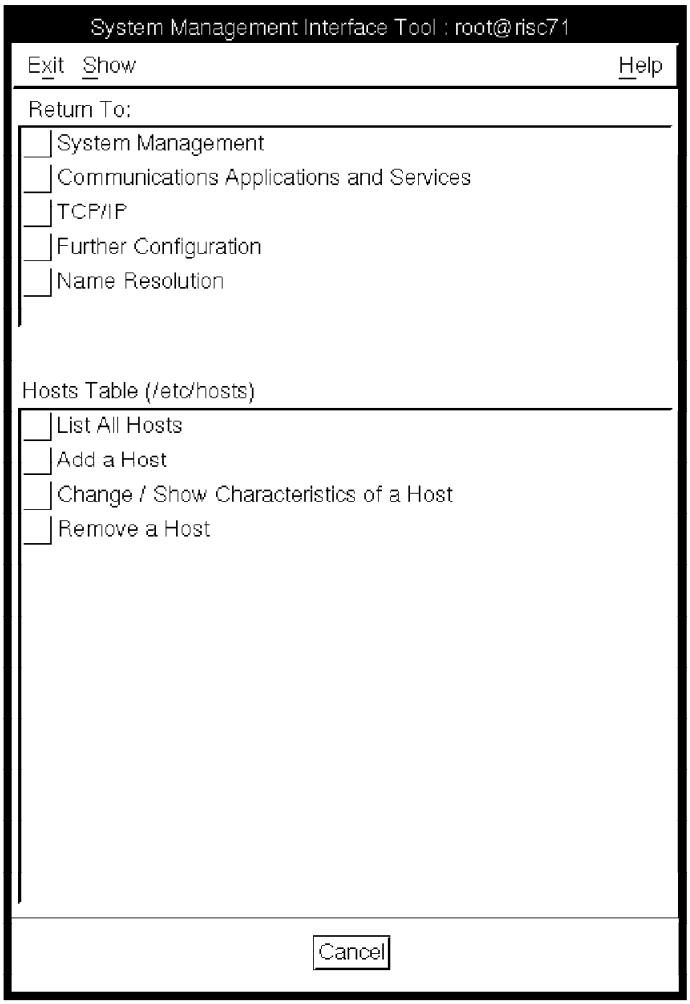


Figure 59. SMIT Host Table (/etc/host) Panel

5. Select **Add a Host**.

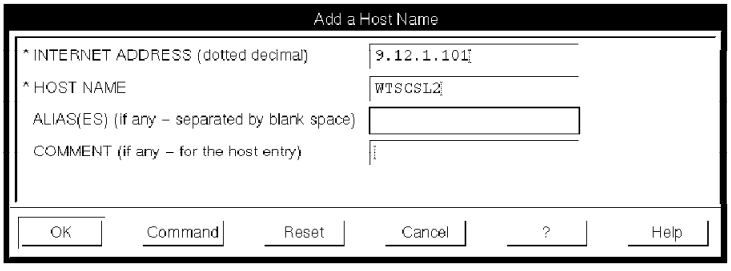


Figure 60. SMIT Add A Host Name Panel

6. Fill in the information requested. In Figure 60, we coded `9.12.1.101` as our Host INTERNET ADDRESS that corresponded to our HOST NAME of `wtscsl2`

Note

These steps were also done for the CLAW-attached OCS, but we coded `9.12.4.101` as our Host INTERNET ADDRESS that corresponded to our HOST NAME of `MVS4101`.

7. Select **OK** to assign the host node. Repeat if required for additional alternate host nodes.

Figure 61. Completion for SMIT Add A Host Name Panel

8. The status in Figure 61 shows the completion for adding the Host Node for `wtscsl2`. Select **Done** to return to Figure 60 on page 64. If no more host names are needed, select **Cancel** to return to Figure 59 on page 64, and select **Exit** to exit out of SMIT.

4.5.3.1 Adding Host Nodes to the OCS Configuration

This procedure will add the host node name to the OCS Configuration. To add the name, follow these steps via SMIT:

1. Invoke the Systems Management Interface Tool (SMIT) by entering `smit` at the AIX command prompt. The SMIT main menu appears.

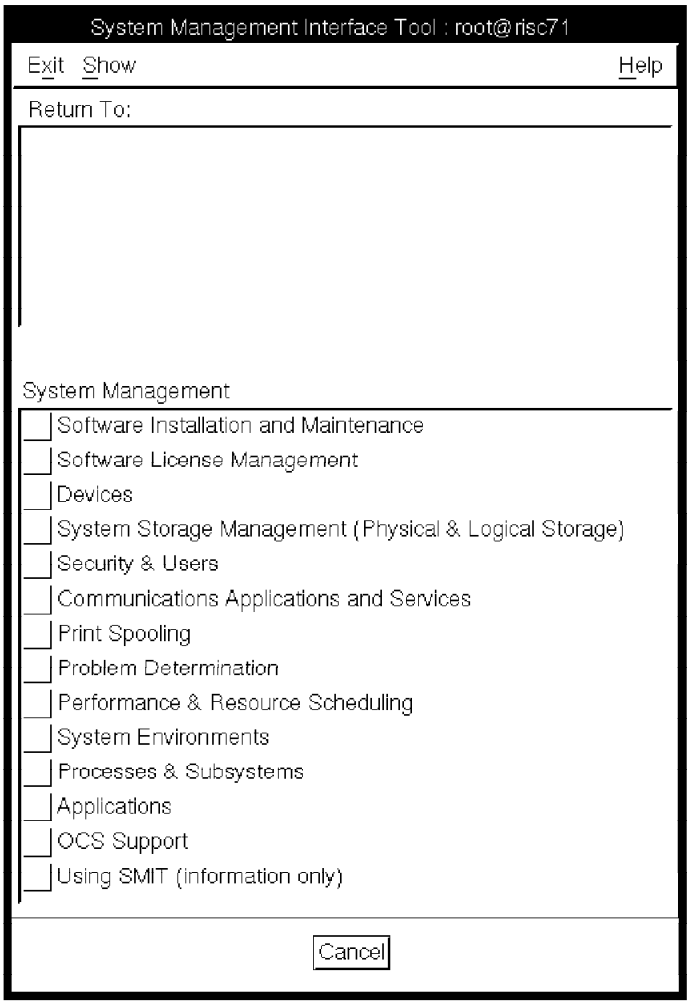


Figure 62. SMIT Main Menu Panel

2. Select **OCS Support**.

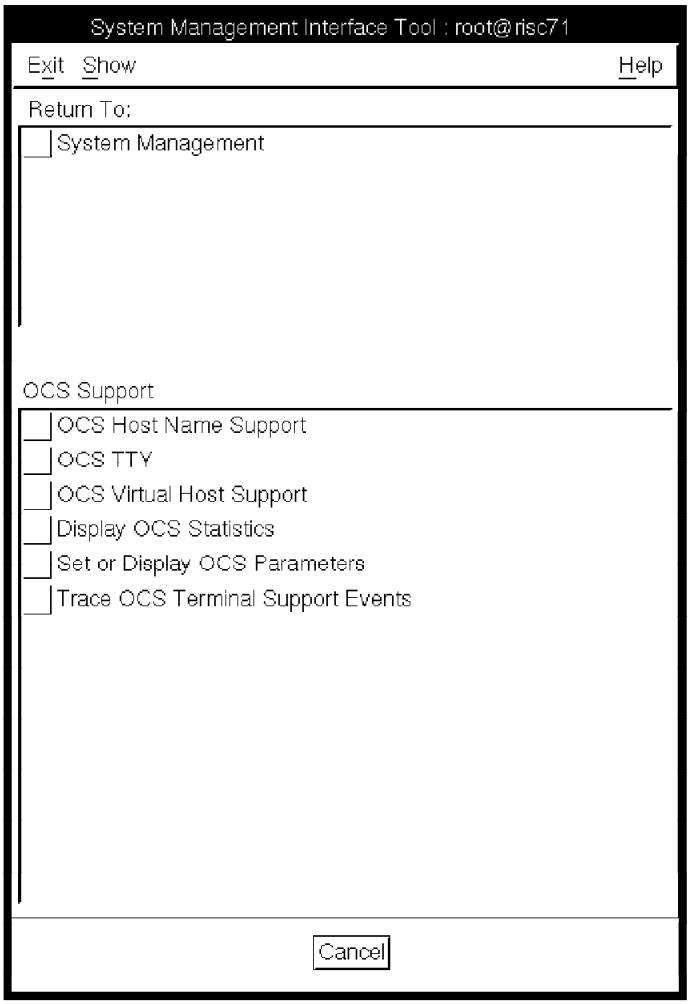


Figure 63. SMIT OCS Support Panel

3. Select **OCS Host Name Support**.

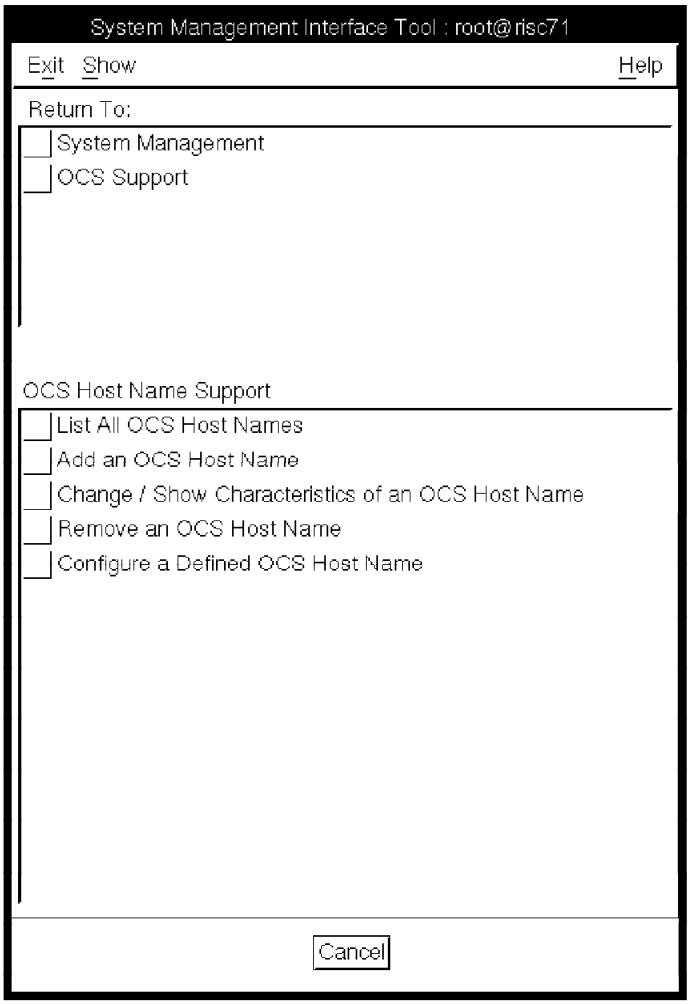


Figure 64. SMIT Host Name Support Panel

4. Select **Add an OCS Host Name**.

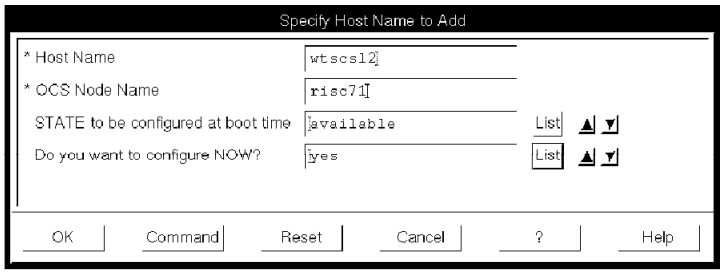


Figure 65. SMIT Add an OCS Host Name Panel

5. Fill in the information requested. In Figure 65, we coded the LAN-attached OCS Host Name as `RISC71` on the OpenEdition/MVS Host Name as `wtscsl2`.

> **Note**
>
> These steps were also done for the CLAW-attached OCS, but we coded `RISC73` as our CLAW-attached OCS Host Name and `mvs4101` as the OpenEdition/MVS Host Name.

State: is the state to be configured at boot time. If you want OCS support to be available at boot time, take the default, available. Otherwise, answer defined.

If you want OCS support to be available immediately, answer yes to Do you want to configure now?

Attention

To answer yes to Do you want to configure now?, the OpenEdition/MVS host has to be configured and ready for connections. Answering yes will result in SMIT issuing a mkdev command. This command will fail and result in the OCS server remaining in the ¢defined¢ state.

If you have not completed the OpenEdition/MVS OCS host configuration, then respond no to Do you want to configure now? and you will have to remember to perform a Configure a Defined OCS Host Name back on Figure 64 on page 68 when you are ready to activate the OCS configuration on the OCS Server.

6. Select **Cancel** to return to Figure 64 on page 68.
7. Select **Cancel** to return to Figure 63 on page 67 and continue with 4.5.3.2, "Adding Virtual Host Nodes to the OCS Configuration."

4.5.3.2 Adding Virtual Host Nodes to the OCS Configuration

If you are offloading virtual terminal support (i.e. rlogin or telnet) to the OCS, use these steps via SMIT to create a virtual host for each attached host node.

1. Select **OCS Virtual Host Support**.

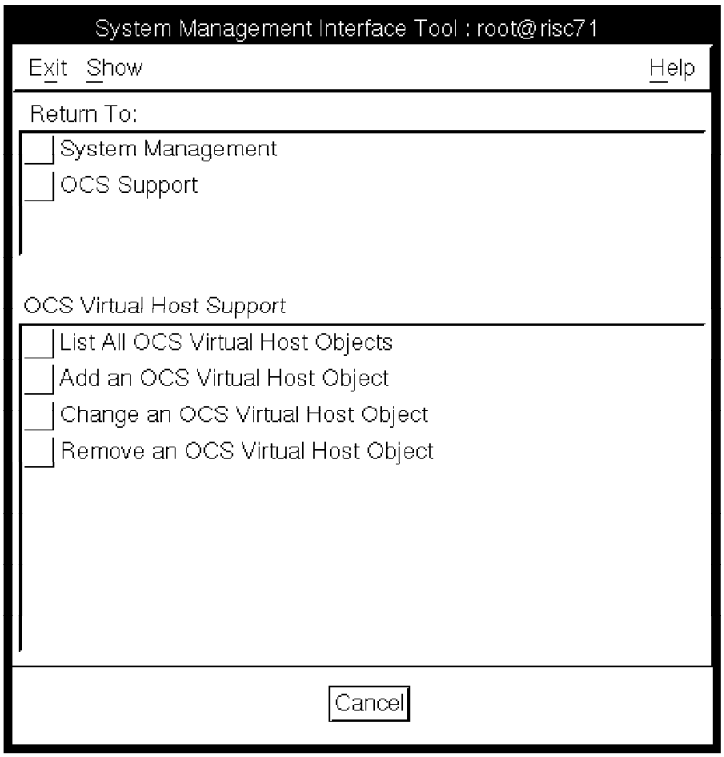


Figure 66. SMIT OCS Virtual Host Support Panel

2. Select **Add an OCS Virtual Host Object**.

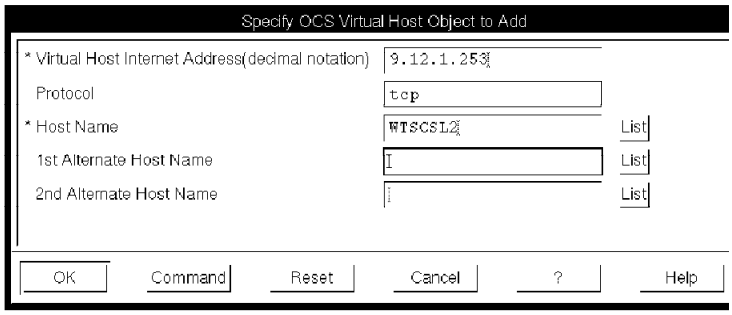


Figure 67. SMIT Add an OCS Virtual Host Object Panel

3. Fill in the information requested. In Figure 67, for the LAN-attached OCS Host we coded the Virtual Host Internet Address as `9.12.1.253`. The Host Name was selected via the pull-down list (F4) and is `wtscsl2`.

   Select **OK** to accept the Virtual Host definition.

   Note: These steps were also done for the CLAW-attached OCS, but we coded `9.12.1.252` as our the Virtual Host Internet Address and the Host Name was `mvs4101`.

   `1st Alternate Host Name`: and `2nd Alternate Host Name`: are the names of other interfaces of the OpenEdition/MVS host systems. If the virtual host connection to the primary host (`wtscsl2`) is not available, then `telnet` or `rlogin` will try to connect via the first alternate host. If this connection is not available, then the second alternate host will be tried next.

   Alternate host names can be either on the same real host or on a different host. Because RISC73 has both LAN and CLAW capability, you could assign `wtscsl2` as an alternate host name for `mvs4101` for the CLAW-attached OCS server. See the OCS hardware environment in Figure 1 on page 4. This would also require that you reconfigure OCS on both OpenEdition/MVS and RISC73 to allow OCS connections via the LAN on OCSNODE2. In our environment, we only configured OCSNODE2 (RISC73) as a CLAW-attached OCS server.

4. Exit SMIT and then continue with 4.5.4, “Defining Terminals on the RISC System/6000.”

4.5.4 Defining Terminals on the RISC System/6000

A terminal that is already defined (configured) as a tty cannot be designated as an OCS terminal unless you first change its state from available to defined (or undefined). To change the tty state, first make sure that the terminal is not in use. Most likely the terminal has a `getty` process assigned to it.

To remove the terminal:

1. Use `rmitab` to remove the tty entry from the /etc/inittab file. For example, to remove the device, issue:

   ```
   rmitab tty20
   ```

2. Then direct the `init` process to re-read the /etc/inittab file, which will also cause any active `getty` process to exit. For example, to re-read the /etc/inittab issue:

   ```
   telinit q
   ```

Now we are to remove the old tty terminals and define new OCS terminals to the OCS Server.

1. Invoke the Systems Management Interface Tool (SMIT) by entering `smit` at the AIX command prompt.

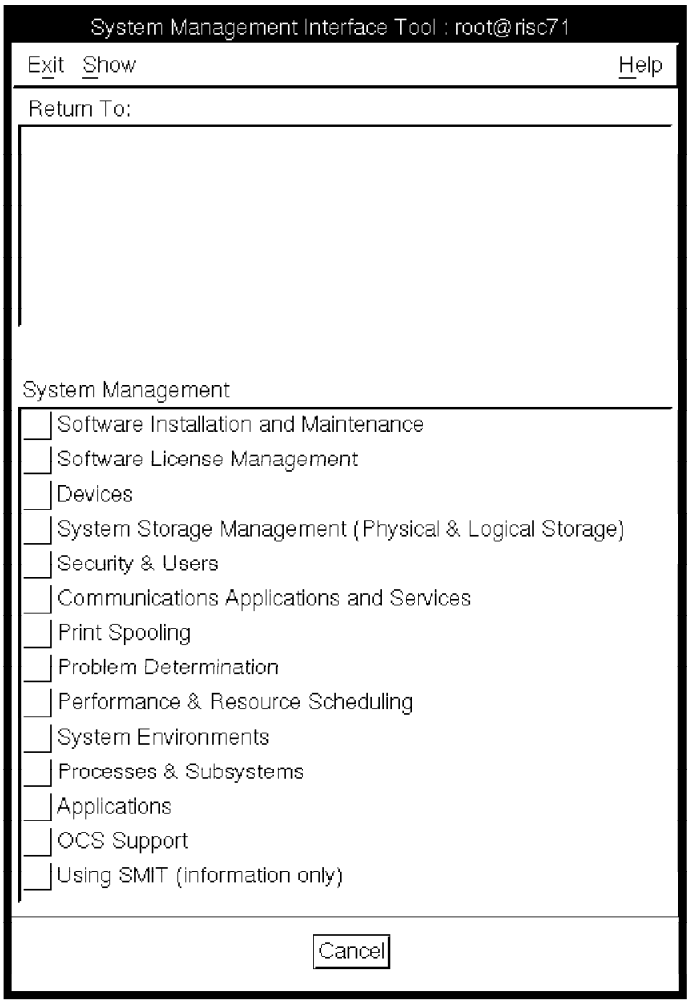


Figure 68. SMIT Main Menu Panel

2. Select **Devices**.

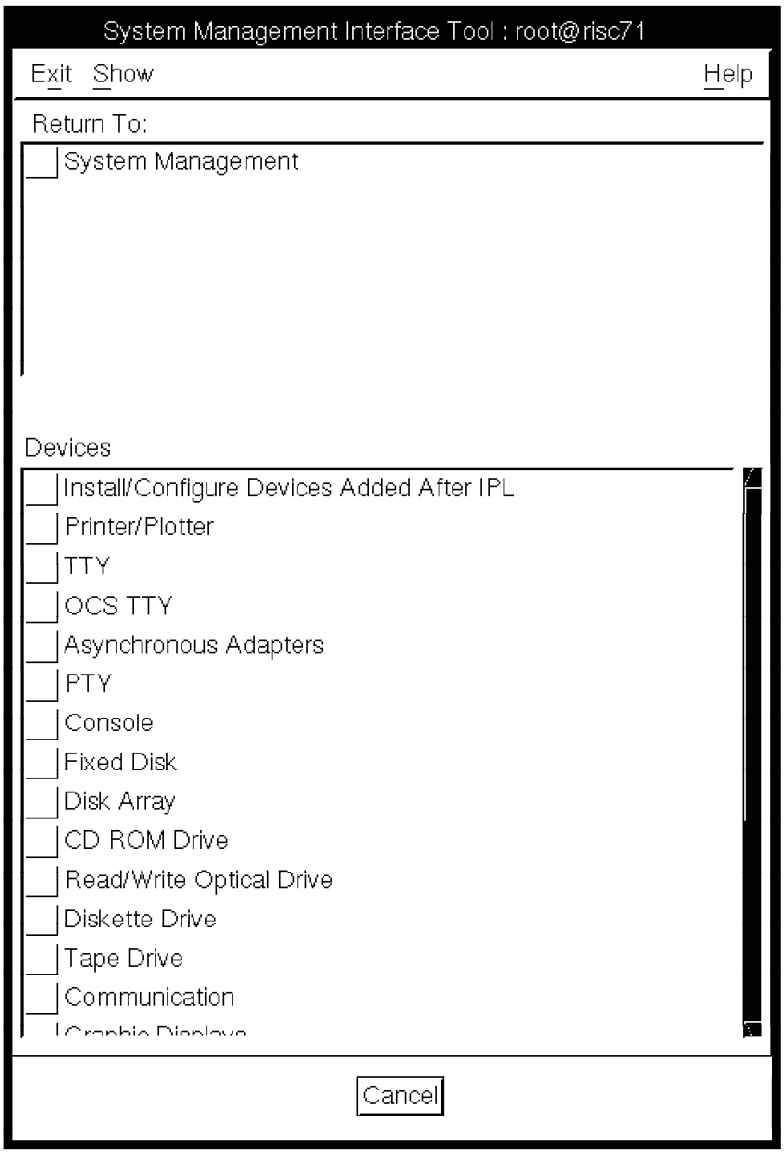


Figure 69. SMIT Devices Panel

3. Select **TTY**.

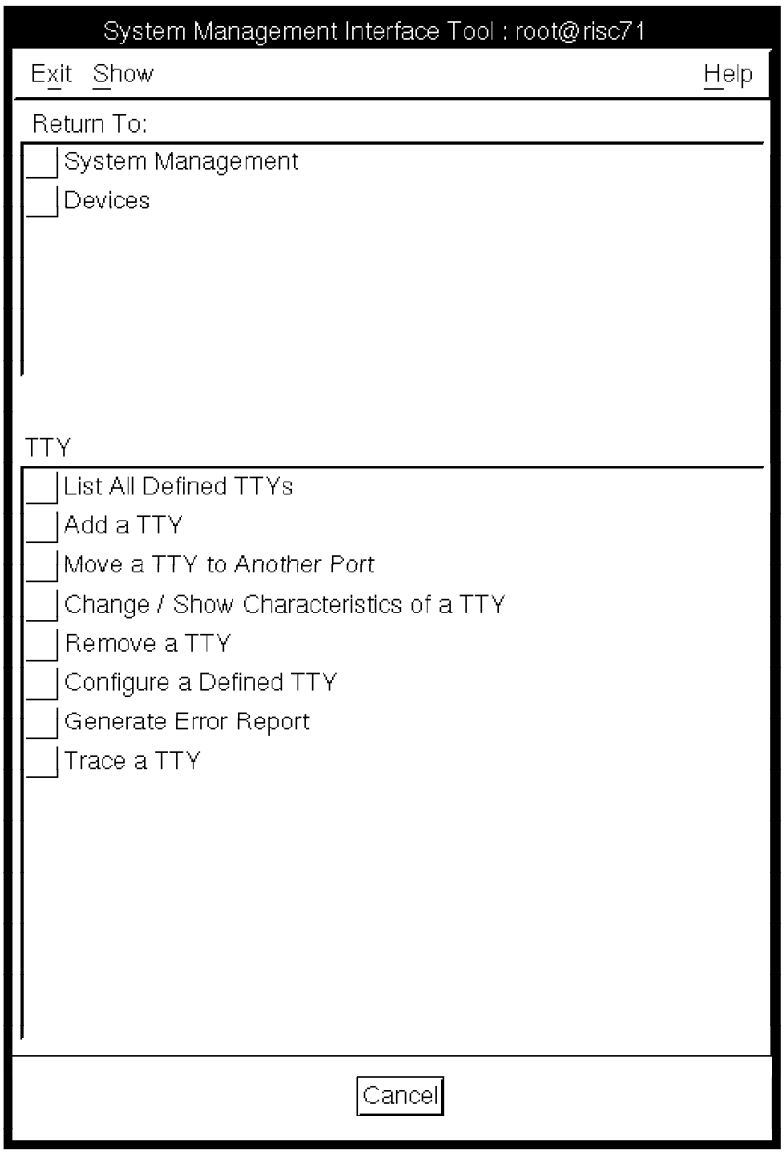


Figure 70. SMIT TTY Panel

4. Select **Remove a TTY**.

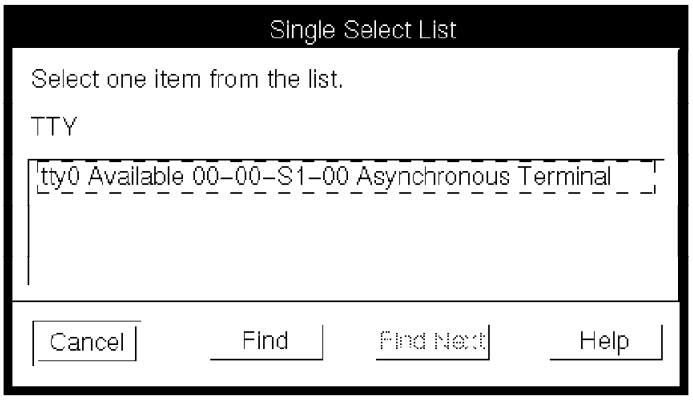


Figure 71. SMIT Remove a TTY Panel

5. Select an available tty from the selection list. In this example, we selected tty0.

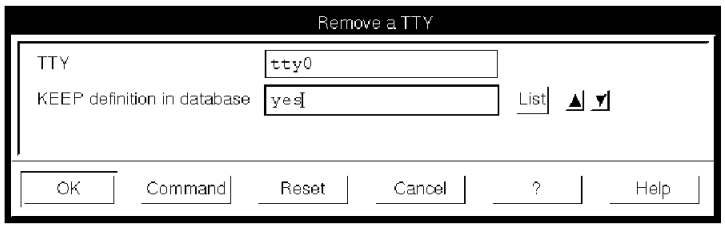


Figure 72. SMIT Remove a TTY Panel (Cont.)

6. Select **OK** to remove the tty0 terminal and take the default to leave the definition in the database.

Figure 73. SMIT Remove a TTY Panel (Cont.)

7. Last chance, select **OK**.

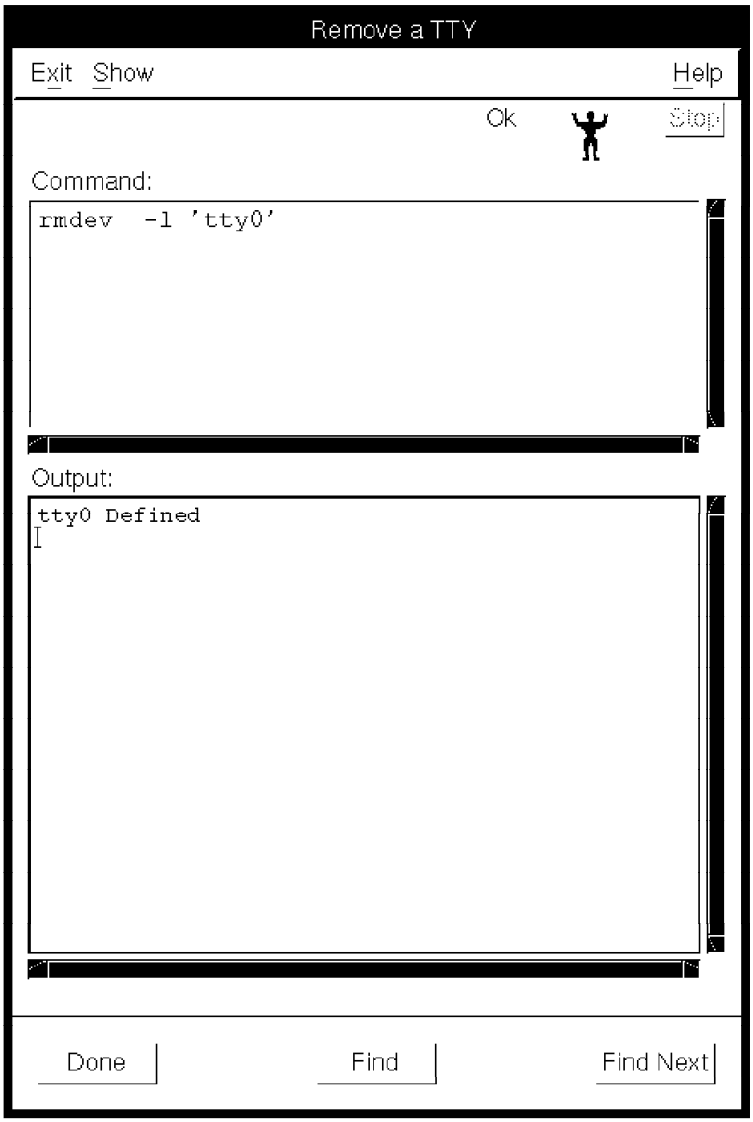


Figure 74. SMIT Completion for Remove a TTY Panel

8. The tty0 terminal has been changed from *available* to *defined*. Now we can go back to define the ocstty. Select **Done** and then **Cancel** three times to return to Figure 62 on page 66.
9. Select **OCS Support**.
10. Select **OCS TTY** and you should now be at Figure 75 on page 76.

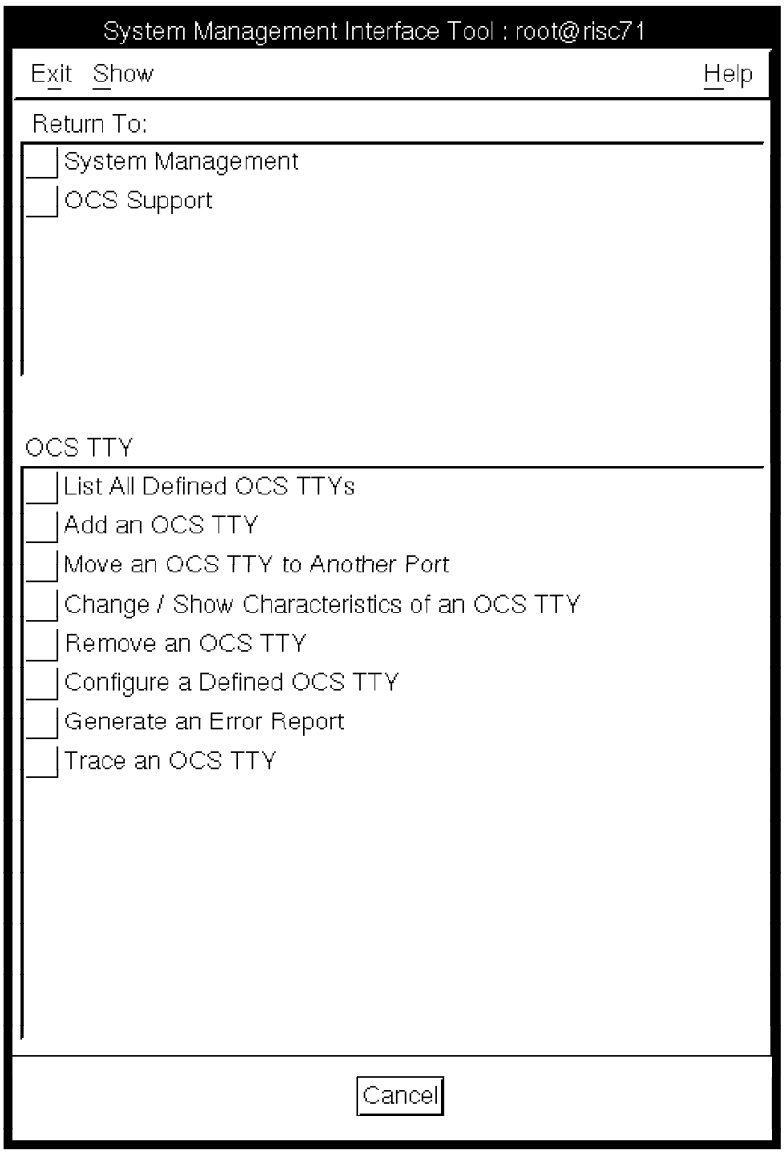


Figure 75. SMIT OCS TTY Panel

11. Select **Add an OCS TTY**.

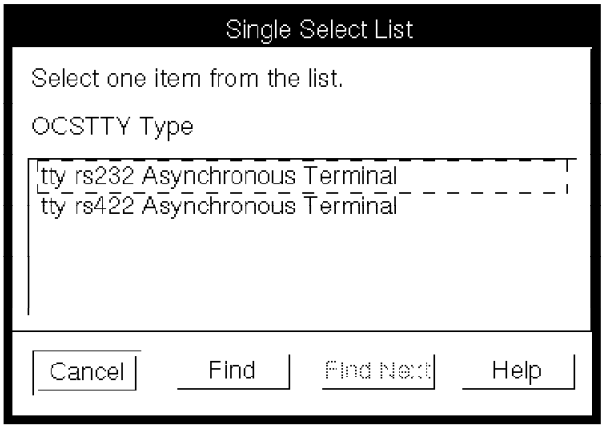


Figure 76. SMIT Add an OCS TTY Panel

12. Select your OCSTTY Type. In our configuration, we are using a rs232 Asynchronous Terminal Type.

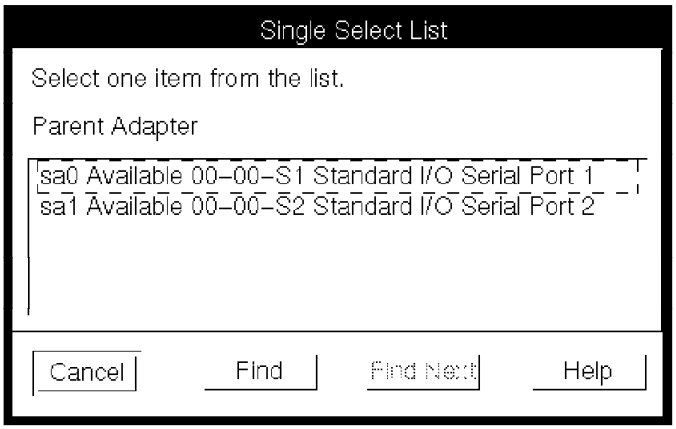


Figure 77. SMIT Select a Parent Adapter Panel

13. Select a Parent Adapter. In our configuration, we are using the sa0 parent adapter.

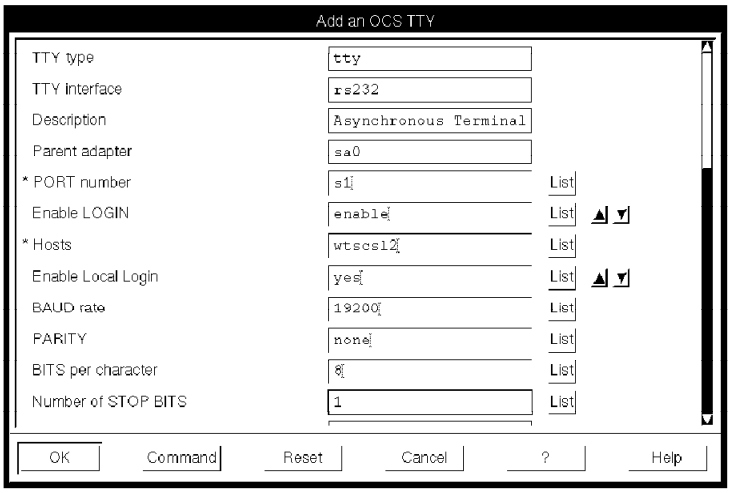


Figure 78. SMIT Add an OCS TTY Panel (Cont.)

14. Fill in the information requested. In Figure 78, the important fields that need changing are:

    a. Port number selected as s1.
    b. Enable Logon set as enable.
    c. Hosts set to the OpenEdition/MVS host. In our configuration this was wtscsl2 for the LAN-attached OCS (RISC71) and mvs4101 for the CLAW-attached OCS (RISC73).
    d. Set Enable Local Login to yes if you want the ocsttys to be allowed to login to the local RISC System/6000.
    e. Set the BAUD Rate to the configured BAUD Rate of the attaching terminal. In our IBM 3151 and IBM 3161 terminal configurations, this was set to 19200.
    f. Use the ¢Scroll Bar¢ to move further down in Figure 78, and set the TERMINAL Type (not shown) to the required type. In our configuration, see Figure 1 on page 4, this setting was set to ibm3161 for both the IBM 3151 and IBM 3161 asynchronous terminals.

Note: These settings must match or you will not communicate with the ocstty. See TTY Devices and Serial Communications, Chapter 4 of the *System Management Guide: Communications and Networks*, SC23-2526-02 for more information.

15. Select **OK** after completing the requested input.

Figure 79. SMIT Completion for Add an OCS TTY Panel

16. Select **Done** and this completes the definition for the ocstty0. If you have more terminals to define, go back to Figure 75 on page 76 and repeat to add additional terminals. When done, select **System Management** to return to the SMIT main menu and continue with 4.5.4.1, “Starting the Terminal Solicitor.”

4.5.4.1 Starting the Terminal Solicitor

After you defined your ocstty terminals and coded yes to Enable Login, you must start the terminal solicitor (ts). The steps are:

1. Add an entry to the /etc/inittab file to respawn the ts. The entry should look like this:

   ```
   startts:123:respawn:/usr/sbin/ts
   ```

2. Signal the `init` process to re-read the /etc/inittab file and start ts. The command is:

   `telinit q`

4.5.4.2 Defining OCS Pseudoterminals

To support virtual host logins you need to define a pool of pseudoterminals (pty) to the OCS Host.

At the SMIT Main Menu:

1. Select **Devices**.
2. Select **PTY**.

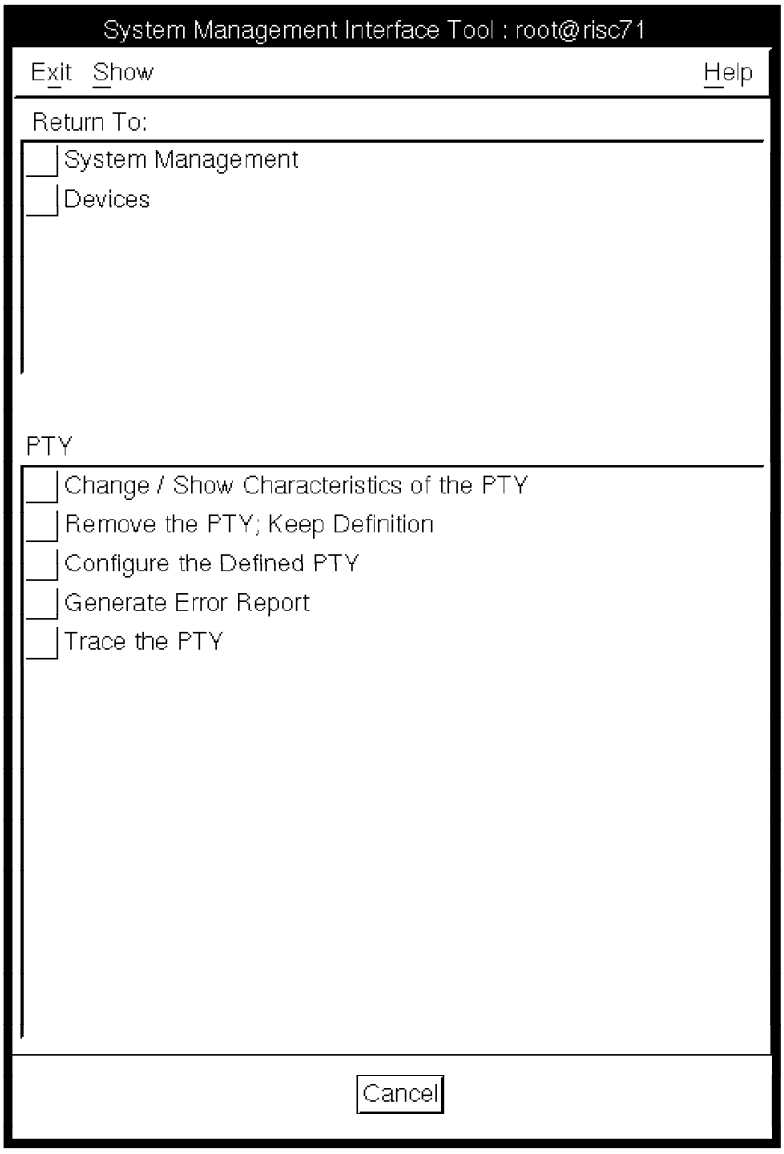


Figure 80. SMIT PTY Panel

3. Select **Change/Show Characteristics of the PTY**.

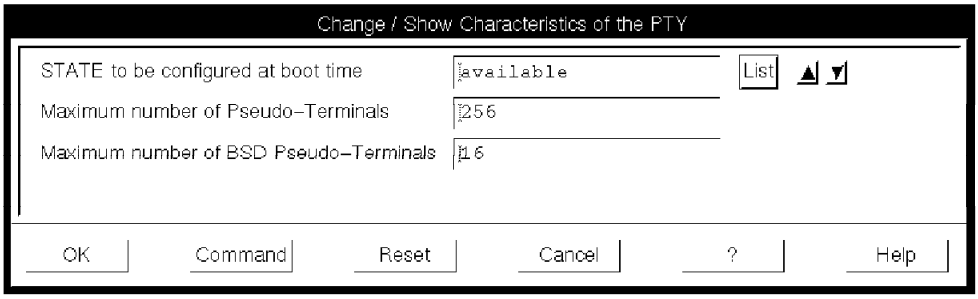


Figure 81. SMIT Change/Show Characteristics of the PTY Panel

4. Choose a **Maximum number of Pseudo-Terminals** value large enough to meet your configuration needs and take the default for **STATE to be configured at boot time** as **available**.
5. Select **Enter** to define the pty.

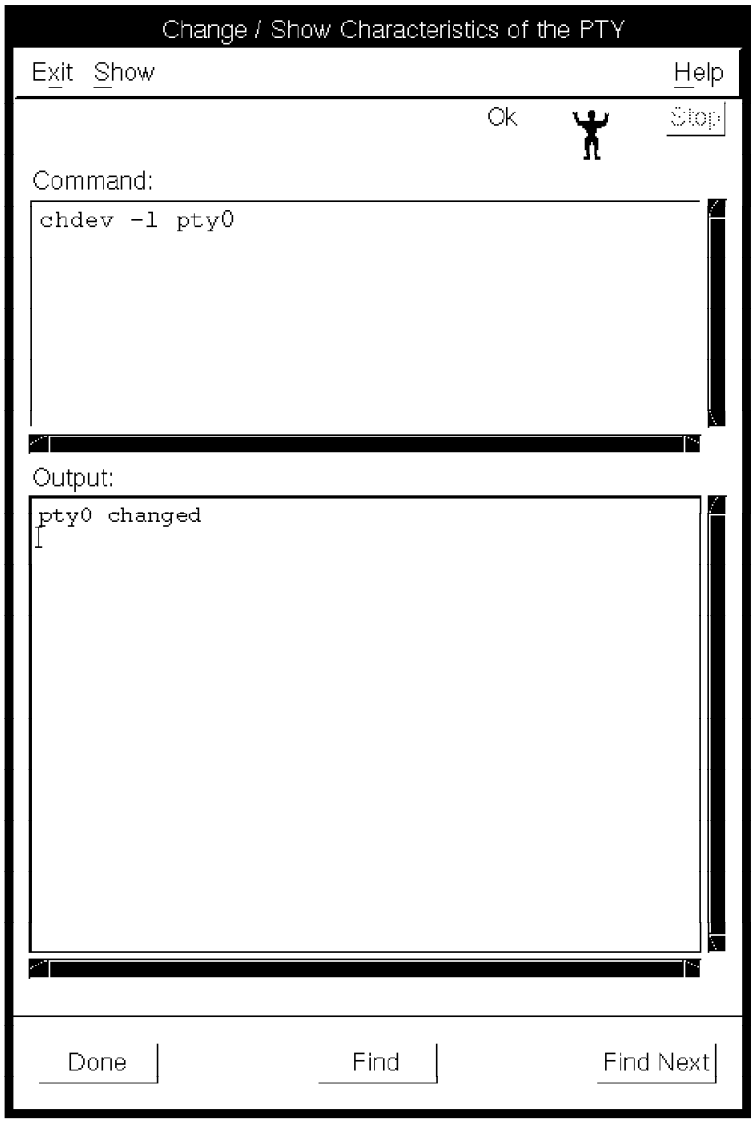


Figure 82. SMIT Completion for Change/Show Characteristics of the PTY Panel

6. This completes the OCS pty. Select **Done** and then select **System Management** to return to the SMIT main menu and continue with 4.5.5, “Assigning Ports to Network Daemons” on page 81.

4.5.5 Assigning Ports to Network Daemons

You must assign a TCP port for both the lm and TBM services. Use SMIT to add the ports. At the SMIT Main Menu:

1. Select **Communications Applications and Services**.
2. Select **TCP/IP**.
3. Select **Further Configuration**.
4. Select **Client Network services**.

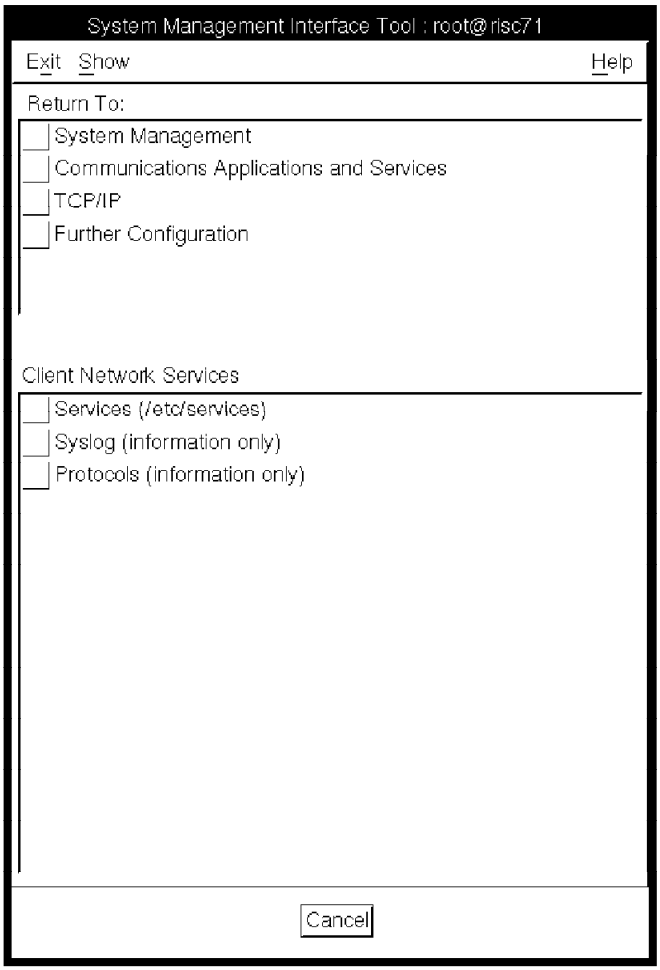


Figure 83. SMIT Client Network Services Panel

5. Select **Services (/etc/services)**.

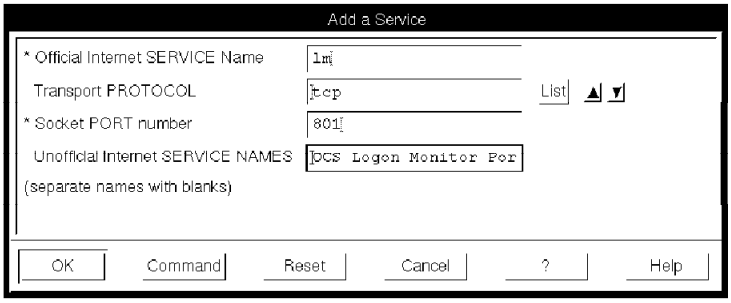


Figure 84. SMIT Add a Service (lm) Panel

6. Fill in the information requested. In the example shown in Figure 84, add to the following fields:
   a. Official Internet SERVICE Name is lm.
   b. Take the default (tcp) for the Transport PROTOCOL.
   c. Socket PORT Number is 801.
   d. Add LM Port in the Unofficial Internet SERVICE NAMES.
   e. Select **OK**.

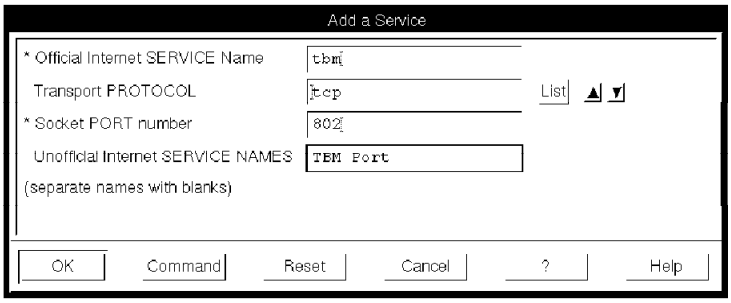


Figure 85. SMIT Add a Service (tbm) Panel

7. Repeat the same steps for the tbm:
   a. Official Internet SERVICE Name is tbm.
   b. Take the default (tcp) for the Transport PROTOCOL.
   c. Socket PORT Number is 802.
   d. Add TBM Port in the Unofficial Internet SERVICE NAMES.
   e. Select **OK**.
8. This completes the port assignments and also completes the configuration of AIX for OCS. Select **Exit** to exit out of SMIT.

Chapter 5. OCS Checklist

This chapter presents a checklist for installing and customizing OCS for OpenEdition/MVS and the RISC System/6000 Environment.

The checklist was split into two major tasks, One for OpenEdition/MVS and the other for RISC System/6000. The OpenEdition/MVS task should be completed first to avoid missing OpenEdition/MVS procedure errors during the RISC System/6000 checklist task that will be done later. If you want to do each sub-task concurrently on both OpenEdition/MVS and the RISC System/6000, then use the checklist provided in *OpenEdition MVS Communications Server Guide*, SC23-3883-00.

5.1 OpenEdition/MVS Checklist

__ 1. Define the connection between the OpenEdition/MVS host and OCS:

   __ a. Make sure the hardware IOCP configuration that provides channel or LAN access to the OCS is defined. Refer to *ES/9000, ES/3090 IOCP User s Guide* for the description of the configuration procedure.

   __ b. Make sure the hardware configuration that provides access to the OCS is defined to TCP/IP. Refer to *IBM TCP/IP for MVS: Customization and Administration Guide* for the description of the configuration procedure.

__ 2. Define the names and addresses for nodes in the OCS system configuration:

   __ a. Update the TCP.HOST.LOCAL (or /etc/hosts) file to include the IP addresses for the OCS node, and the corresponding OCS node names. If you add the host entry to the TCP.HOST.LOCAL file, you must use the MVS for TCP/IP MAKESITE utility to generate the corresponding TCPIP.HOSTS.SITEINFO and TCPIP.HOSTS.ADDRINFO datasets.

      You do not need to add a local host entry for the OCS node if it is registered in a domain name server.

__ 3. Define the terminals and pseudoterminals on the OCS to be used as if directly connected to the host:

   __ a. Be aware of the special file names assigned to the terminals and pseudoterminals on the OCS.

__ 4. Define the same terminals and pseudoterminals to OpenEdition/MVS, mapping the OpenEdition/MVS special file names for the terminals to the corresponding device special file names on the OCS:

   __ a. Define the terminals and pseudoterminals as serial line terminals and virtual terminals in the OCS node stanza statement in the /etc/system file.

__ 5. Define the login monitor and Terminal Buffer Manager TCP/IP services:

   __ a. Add the following entries to the TCPIP.ETC.SERVICES or /etc/services file:

```
lm      801/tcp    # OCS Login Monitor Port
tbm     802/tcp    # OCS TBM Port
```

__ 6. Optionally define a herald message for the login monitor:

__ a. Create or edit the /etc/banner file with the desired herald message. This message is displayed before the login prompt.

__ 7. If the TCP/IP connection between the OpenEdition/MVS host and OCS node is not fully configured, then activate the connection.

Refer to *IBM TCP/IP for MVS: Customization and Administration Guide* for this procedure.

__ 8. Configure the OCS-attached terminals and OCS pseudoterminals into the system configuration:

__ a. Use the `ocsconfig` command with the `-c` option against the configuration file stanza that defines the OCS terminal configuration.

To have OCS configured automatically during the OpenEdition MVS kernel initialization, add the `ocsconfig` command with the `-c` and `-a` options to the /etc/rc shell script.

Note: To confirm that the OpenEdition/MVS—OCS connection is established, use the `ocsconfig` command with the `-q` option. Verify that the state of the OCS node is connected.

__ b. If reconfiguring an OCS node, issue the `kill -HUP` command against the process identifier (PID) of the `lm` process to signal the login monitor to update its view of the OCS configuration.

__ 9. Start the login monitor to process inbound OCS login requests:

- If you choose to start the logon monitor automatically during OpenEdition/MVS kernel initialization:

__ a. Create a shell script to issue the `lm` command. For example, you could create a shell script /etc/lm.rc as follows:

```
export _BPX_JOBNAME=LM
/usr/sbin/lm 2> /etc/lm.log
```

The statements perform the following functions:

– The first statement sets the jobname for the logon monitor address space to LM, instead of the default name of ETCINIT or ETCRC.

– The second statement starts the login monitor. Because the login monitor executes as a daemon, you do not need to start the login monitor as a background job.

In addition, all messages that the login monitor issues are redirected to /etc/lm.log.

__ b. Add the invocation of /etc/lm.rc to /etc/rc.

- If you choose to start (or restart) the logon monitor as a cataloged procedure, you need to define the MVS PROC and security environment. See *OpenEdition MVS Communications Server Guide*, SC23-3883-00 for more details.

5.2 AIX/6000 Checklist

__ 1. Define the connection between the OpenEdition/MVS host and OCS:

__ a. For a channel-attached OCS, follow the instructions in the *Block Multiplexer Channel Adapter: Users Guide and Service Information*, SC23-2427-02

__ b. For a LAN-attached OCS, use SMIT to define the LAN connection to the OpenEdition/MVS host. See *System Management Guide: Communications and Networks*, SC23-2526-02 for more details.

__ 2. Define the names and addresses for nodes in the OCS system configuration:

__ a. Update the /etc/hosts file to include the IP address for the host node, and the corresponding host names. Use the following SMIT menu selections: Communications Applications and Services, TCP/IP, Further Configuration, Name Resolution, Hosts Table (/etc/hosts), Add a Host.

__ b. Use SMIT to add the host to the OCS system configuration with the following menu selections: OCS Support, OCS Host Name Support, Add an OCS Host Name.

You do not need to add a local host entry for the OpenEdition/MVS node if it is registered in a domain name server.

__ c. If you are offloading virtual terminal support to the OCS and need internet alias addresses on the OCS to represent attached host nodes, then use the ifconfig command to assign alias addresses to the network interfaces on the OCS. Place the ifconfig commands in the /etc/rc.net file to have the alias addresses automatically assigned during the OCS system boot.

__ d. If you are offloading virtual terminal support to the OCS, use SMIT to create a virtual host for each attached host node.

__ 3. Define the terminals and pseudoterminals on the OCS to be used as if directly connected to the host:

__ a. If a terminal to be used as an OCS tty is currently in the available state, you need to change its state to defined, as follows:

__ 1) Remove the reference to the terminal device in the /etc/inittab file. For example, for terminal device /dev/tty20, issue:

```
rmitab tty20
```

__ 2) Use the telinit command to direct the init process to reread the /etc/inittab file:

```
telinit q
```

Active getty processes will exit.

__ 3) Use SMIT to change the terminal state to defined with the following menu selections: Devices, TTY, Remove a TTY, name of TTY.

__ b. Use SMIT to define terminals on asynchronous ports as OCS serial line terminals with the following menu selections: OCS

Support, OCS TTY, Add an OCS TTY. Fill in the key required fields (Port Number, Hosts) on the Add an OCS TTY menu after selecting the OCSTTY Type and Parent Adapter on the two submenus that precede it. Change other values, if appropriate.

__ c. If necessary, use SMIT to define additional pseudoterminals to support remote login to OCS-attached hosts with the following menu selections: Devices, PTY, Change / Show Characteristics of the PTY. The same pool of pseudoterminals is used to support remote logins both to the OCS and to the host through the OCS.

__ 4. Define the login monitor and Terminal Buffer Manager TCP/IP services:

__ a. Add the following lm and TBM entries to /etc/services that were assigned the same port numbers and protocol that OpenEdition/MVS uses:

```
lm      801/tcp    # OCS Login Monitor Port
tbm     802/tcp    # OCS TBM Port
```

If the port numbers are already assigned to other services, then pick unused port numbers, but be sure to update the TCPIP.ETC.SERVICES or /etc/services file on OpenEdition/MVS to use consistent port numbers.

__ 5. Make sure the terminal solicitor will run on the OCS. You need the OCS terminal solicitor if you have OCS-attached serial terminals in your configuration:

__ a. Add an entry to /etc/inittab to respawn the terminal solicitor. The entry should be something like this:

```
startts:123:respawn:/usr/sbin/ts
```

__ b. Make sure that terminal solicitor support is enabled for the appropriate terminals. Use SMIT, either when the OCS tty is originally defined on the Add an OCS tty menu, or with the following menu selections: OCS Support, OCS TTY, Change / Show Characteristics of an OCS TTY. Specify enable for the Enable Login field.

__ 6. If the TCP/IP connection between the OpenEdition/MVS host and OCS node is not fully configured, then activate the connection:

__ a. Use SMIT to configure the appropriate channel or LAN interfaces that give access to the host node.

__ b. Use SMIT or the ifconfig command to assign IP addresses for the network interfaces, as necessary.

__ 7. Configure the OCS-attached terminals and OCS pseudoterminals into the system configuration:

__ a. Use SMIT to configure the host name. If the host is not in the fully configured state (available), then change its state to available with the following menu selections: OCS Support, OCS Host Name Support, Configure a Defined OCS Host Name. Select the terminal name from the Host Name submenu.

__ b. Use SMIT to configure the terminals. If a terminal is not in the fully configured state (available), then change its state to available with the following menu selections: OCS Support, OCS TTY, Configure a Defined OCS TTY, and then select the terminal name from the OCSTTY submenu.

Chapter 6. Tuning OCS

This chapter explains how to tune the OCS configuration for each OCS node. For more detailed tuning information, see *OpenEdition MVS Communications Server Guide*, SC23-3883-00.

We do not provide suggested tuning parameter values. At the time of publication of this redbook, *OpenEdition MVS Communication Server Implementation Guide*, SG24-4619-00, there was no available performance data that we could include.

6.1 Tuning OCS for OpenEdition/MVS

Tuning is beyond the scope of this project. We can only offer references were to look for tuning information.

There is no direct tuning of OpenEdition/MVS OCS. Each of the following components MVS, TCP/IP, and OpenEdition/MVS will have to be tuned individually. The following manuals are the tuning guides for the different components:.

1. *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02
2. *IBM MVS/ESA Initialization and Tuning Reference*, SC28-1452
3. *IBM TCP/IP Performance Tuning Guide*, SC31-7188-00

6.2 Tuning OCS on the RISC System/6000

There are no tuning parameters at the OpenEdition/MVS host system; however, you can use SMIT to tune your OCS node. To accomplish this:

1. Use SMIT Display OCS Terminal Statistics to start or stop recording OCS statistical information, or to display OCS statistical information.
2. Use SMIT Set or Display OCS Parameters to set or display system parameters related to OCS terminal support.

These steps are outlined in the section titled “Advanced Administration – Tuning OCS” of the *OpenEdition MVS Communications Server Guide*, SC23-3883-00.

For each host node, you can set the following:

- The input buffer size
- The output buffer size
- The number of messages accumulated before the output buffer is sent
- The time elapsed before the output buffer is sent

The OCS node has a set of these parameters for each host in the OCS node′s configuration. You can also set the following OCS system parameters:

- The number of hash sets for rty device table entries
- The maximum number of free server processes to handle OCS terminal support messages forwarded from host nodes

Attention

Changing the values of the parameters affects the performance of the OCS configuration—change them carefully.

6.2.1 Guidelines for Choosing OCS Parameters

This section provides guidelines for changing each of the tunable parameters on the OCS system. See *OpenEdition MVS Communications Server Guide*, SC23-3883-00 for SMIT procedures to change these parameters.

The following parameters are set for each OCS node—OpenEdition/MVS host connection:

- The Input buffer size (inbuf)
- The output buffer size (outbuf)
- The maximum number of messages (mlimit)
- The new time interval value (tlimit)

The following parameters are global for the OCS node:

- The maximum number of free server processes to handle OCS terminal support messages forwarded from host nodes (slimit)
- The number of hashing sets for rty device table entries (rhash)

6.2.2 Input Buffer Size (inbuf)

The inpuf parameter specifies the size, in bytes, of the input buffer used by the OCS terminal support to receive messages from the OpenEdition/MVS host node. The OCS default value is 1024 bytes.

The rhash parameter corresponds with the size of the OpenEdition/MVS host output buffer size, which has a static value of 1024 bytes. The value of inbuf should always be greater than or equal to the size of the OpenEdition/MVS host output buffer.

6.2.3 Output Buffer Size (outbuf)

The outbuf parameter specifies the size, in bytes, of the output buffer used by the OCS terminal support to receive messages from the OpenEdition/MVS host node. The OCS default value is 1024 bytes.

This parameter corresponds with the size of the OpenEdition/MVS host input buffer size, which has a static value of 1024 bytes. The value of inbuf should always be less than or equal to the size of the OpenEdition/MVS host input buffer.

6.2.4 Maximum Number of Messages (mlimit)

The mlimit parameter specifies the number of messages that can accumulate in the output buffer used by OCS terminal support. When this limit is reached, the contents of the buffer are forwarded to the OpenEdition/MVS host node. The default value is 16.

mlimit should be used together with tlimit, described in 6.2.5, “New Time Interval Value (tlimit)” on page 89. Together they control the frequency that the outbound message queue is flushed, causing the data to be sent to the

OpenEdition/MVS host system. When the first message is placed in the outbound data queue, a timer is started for a duration specified by `tlimit`. When either `mlimit` is reached (because of more outbound messages being queued) or the timer expires, the output data is sent to the OpenEdition/MVS host system.

The values specified for `mlimit` and `tlimit` have the greatest impact. Of all the OCS parameters for OCS system performance, choose these values carefully.

6.2.5 New Time Interval Value (tlimit)

The `tlimit` parameter specifies the time interval, expressed in fractions of a second, that can elapse before the contents of the output buffer is sent to the OpenEdition/MVS host node. The default value is `.02` seconds.

6.2.6 Maximum Number of Rty Servers (slimit)

The `slimit` parameter specifies the maximum number of free server processes available in the server pool to handle OCS terminal support messages forwarded from the host. If this maximum has not been reached, a server process enters the pool when it has finished handling a message. If the pool is full, the server process exits. The default value for `slimit` is `1024`.

This parameter should be less than or equal to the expected peak number of active OCS users in your configuration. Because each server process requires memory resources, if you are using the AIX system for work other than OCS, you may need to decrease this value.

6.2.7 Number of Hash Sets for the Rty Table (rhash)

This parameter specifies the number of hash sets (equivalence classes) into which rty device table entries are distributed. The default value for this parameter is `512`.

This parameter should be less than or equal to the expected number of active OCS users in your configuration.

Chapter 7. OCS Error Recovery Scenarios

This chapter describes a number of typical OCS error recovery scenarios.

A number of procedures were tried in each of the scenarios and what we present here is the optimal solution for each. Also, there will be warnings of what not to do.

7.1 LAN Connection Severed from AIX OCS node

- **Notification or detection of the error**
  - Workstation terminals hang. All sessions to the OpenEdition/MVS system are suspended.
- **OpenEdition/MVS procedures**
  - Issue the `ocsconfig -q` command. The query will indicate that all OCS servers are connected and the rty's are ¢open¢ in the states field for the hung terminals that are associated with the failing OCS server connection.

    Attention

    Do not re-config the OCS host by issuing the `ocsconfig -u` or the `ocsconfig -c` commands. This will only mask the current error situation and lead to more confusion.

  - Issuing the `ping` command to the affected OCS server will time out. Issuing `ping` to the other host will echo without errors. This indicates that there is a connection problem to this particular OCS server.
- **AIX procedures**
  - Issuing the `ping` command to the OpenEdition/MVS system will time out. Issuing `ping` to the OCS server will echo without errors. This indicates that there is a connection problem from this particular OCS server.
  - Issuing the `ps -A` command at the OCS server will indicate that the terminals in session will be running a ¢kproc¢ (Kernel PROCcess) process. These are the terminals that will show a rty ¢open¢ state on the OpenEdition/MVS host via the `ocsconfig -q` command. Terminals that were not in session will be in a ¢ts_getty¢ process.
- **Solution**
  - Correct the broken connection. We found a bad token-ring cable.
  - After re-connection the following error messages were posted on the MVS console:

```
BPXB004E OCS HAS LOST ITS CONNECTION TO THE FOLLOWING NODE(S): risc71
BPXF024I (OMVSKERN) Oct 17 12:44:36 lm 262156 : FOMC1015 Device open 966
of </dev/sty10> failed. No login session was started.
BPXF024I (OMVSKERN) Oct 17 12:44:36 lm 262156 :  EDC5112I Resource 967
temporarily unavailable.
BPXF024I (OMVSKERN) Oct 17 12:44:48 lm 327692 : FOMC1015 Device open 968
of </dev/sty11> failed. No login session was started.
BPXF024I (OMVSKERN) Oct 17 12:44:48 lm 327692 :  EDC5112I Resource 969
temporarily unavailable.
BPXF024I (OMVSKERN) Oct 17 12:44:51 lm 393228 : FOMC1015 Device open 970
of </dev/sty11> failed. No login session was started.
BPXF024I (OMVSKERN) Oct 17 12:44:51 lm 393228 :  EDC5112I Resource 971
temporarily unavailable.
```

Figure 86. MVS Console Output

Attention

The error messages did not post during time of failure. Only after re-connection did the messages get posted. This is because of the nature of LAN networks (non-polling environment).

- The terminal sessions will resume or reestablish with the terminal login screen with no actions required on the OCS server or OpenEdition/MVS system.

7.2 OpenEdition/MVS OCS Node Unconfigured

- **Notification or detection of the error**
  - Serial terminal sessions close and return to the login prompt. Virtual terminals will terminate sessions and return to terminal operating system prompt. All sessions to the OpenEdition/MVS system are lost.
- **OpenEdition/MVS procedures**
  - Issuing the `ping` command to any OCS server will echo without errors.
  - Issue the `ocsconfig -q` command. The query will indicate that all OCS servers are unconfigured.

```
-----------------
OCS query results
-----------------
Most recent ocsconfig configure/unconfigure command:
Command  : ocsconfig -u
UID      : 0
User name: SKOREY
Time     : Wed Oct 18 10:57:14 1995
OCS daemon state: Inactive                    Active nodes = 0
RTYs            : Configured RTYs = 0         Maximum RTYs = 256
OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 0         Open RTYs = 0
  State         : Not configured
OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 0         Open RTYs = 0
  State         : Not configured
```

Figure 87. Results from the ocsconfig -q Command

- **AIX procedures**
  - Issuing the ping command to the OpenEdition/MVS system will echo without errors. Issuing ping to the OCS server will also echo without errors.
  - Issuing the ps -A command at the OCS server will indicate that the terminals are not in session and will be in a ¢ts_getty¢ process. The serial terminals will be at the login prompt.
- **Solution**
  - The interrogation of the ocsconfig -q indicated that most likely the ocsconfig -u command was issued by error. This is further confirmed by the ocsconfig -u command that appears in Figure 87 command field. This field contains the last ocsconfig command issued.
  - Reissue the ocsconfig command to reconfigure the OCS servers.
  - Users will need to login to reestablish their sessions.

7.3 OpenEdition/MVS OCS Terminal Unconfigured

- **Notification or detection of the error**
  - Some serial terminals cannot connect to the OpenEdition/MVS host. The terminals are configured on the OCS server and have the login prompt, but when selecting wtscsl2 from the login prompt on ocstty0, the following message is displayed:

    Connecting to wtscsl2 ...
  - After attempting the connection, the login prompt is redisplayed.
- **OpenEdition/MVS procedures**
  - Issue the ocsconfig -Q command. The query in Figure 88 on page 94 will indicate that all OCS server nodes are connected. Notice that the /dev/sty11 for OCS node name risc71 does not fall on the correct boundary for the /dev/ocstty0 terminal definition.

```
-------------------------
OCS extended query results
-------------------------
Most recent ocsconfig configure/unconfigure command:
Command  : ocsconfig -a
UID      : 0
User name: SKOREY
Time     : Wed Oct 18 12:31:56 1995
OCS daemon state: Waiting for connections     Active nodes = 2
RTYs            : Configured RTYs = 9         Maximum RTYs = 256
OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 5         Open RTYs = 0
  State         : Connected
  Debug flags   : C8300000
  Terminal devices
  ------------------------------------------------------------------------------
  0     /dev/sty00                            /dev/ocstty0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  1     /dev/sty01                            /dev/ocstty1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  2     /dev/vty2                             /dev/pts/0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  3     /dev/vty3                             /dev/pts/1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  4     /dev/vty4                             /dev/pts/2
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 4         Open RTYs = 0
  State         : Connected
  Debug flags   : C8300000
  Terminal devices
  ------------------------------------------------------------------------------
  10    /dev/sty11                            /dev/ocstty1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  11    /dev/sty12                            /dev/ocstty2
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  20    /dev/vty20                            /dev/pts/0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  21    /dev/vty21                            /dev/pts/1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
```

Figure 88. Results from the ocsconfig -Q Command

- Browse or edit the /etc/system and validate the stanza for the affected OCS server. Check the correctness for the following parameters:
  - MVS_Device_Char_Minor
  - MVS_Device_Char_Files
  - Ocs_Device_Char_Files
- Browsing the /etc/system file resulted in the following display:

```
################################################################################
# ===========
# First stanza:   OCS object name = ocsnode1
# ===========    -------------------------
ocsnode1:
      Device_Description    = †OCS1 - Channel-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc473
      Automatic             = yes
      MVS_Device_Char_Minor = †[0-4]†
      MVS_Device_Char_Files = †sty0[0-1]†    , \
                              †vty[2-4]†
      Ocs_Device_Char_Files = †ocstty[0-1]†  ,  \
                              †pts/[0-2]†
# ============
# Second stanza:   OCS object name = ocsnode2
# ============    -------------------------
ocsnode2:
      Device_Description    = †Ocsnode2 - Lan-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc71
      Automatic             = yes
      MVS_Device_Char_Minor = †[10-11]†, †[20-21]
      MVS_Device_Char_Files = †sty[11-12]†    , \
                              †vty[20-21]†
      Ocs_Device_Char_Files = †ocstty[1-2]† , \
                              †pts/[0-1]†
######## End of Configuration File ###########################################
```

Figure 89. Results from Browsing the /etc/system File

- **AIX procedures**
  - Verify that the ocstty addresses are correct and correspond to the correct range of stys on the OpenEdition/MVS system.
- **Solution**
  - Verify and correct your sty, vty and pts ranges in the /etc/system file on the OpenEdition/MVS system. The correct range must match with the OCS server defined ocsttys. For example:

    /dev/sty10 ==> /dev/ocstty0
    /dev/sty11 ==> /dev/ocstty1

    Attention

    The range of ocsttys must start at zero to match the starting ocstty definition at zero (ocstty0) on the RISC System/6000. See 4.2.4, “Defining TCP/IP Hosts For OCS Nodes” on page 29 for more information on coding the OpenEdition/MVS configuration parameters for the OCS server and terminals.

  - By editing the /etc/system file and correcting the stanza for ocsnode2 with the following correction:

    Ocs_Device_Char_Files = †ocstty[0-1]†   , \

    the corrected file appears as follows:

```
################################################################################
# ===========
# First stanza:   OCS object name = ocsnode1
# ===========    -------------------------
ocsnode1:
      Device_Description    = †OCS1 - Channel-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc473
      Automatic             = yes
      MVS_Device_Char_Minor = †[0-4]†
      MVS_Device_Char_Files = †sty0[0-1]†    , \
                              †vty[2-4]†
      Ocs_Device_Char_Files = †ocstty[0-1]†  ,  \
                              †pts/[0-2]†
# ============
# Second stanza:   OCS object name = ocsnode2
# ============    -------------------------
ocsnode2:
      Device_Description    = †Ocsnode2 - Lan-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc71
      Automatic             = yes
      MVS_Device_Char_Minor = †[10-11]†, †[20-21]†
      MVS_Device_Char_Files = †sty[10-11]†    , \
                              †vty[20-21]†
      Ocs_Device_Char_Files = †ocstty[0-1]† , \
                              †pts/[0-1]†
######## End of Configuration File ############################################
```

Figure 90. Results from Editing the /etc/system File

- After correcting the OCS configuration file, issue the `ocsconfig -c ocsnode2` command to have the new configuration take effect. Then issue the `ocsconfig -q` command to display the status of the corrected sty configuration shown in Figure 91 on page 97.
- Recycle Login Manager (lm) with the following command:

  `kill -HUP [pid number of lm]`
- Workstation ocstty0 can now successfully login to the OpenEdition/MVS system.

```
-------------------------
OCS extended query results
-------------------------
Most recent ocsconfig configure/unconfigure command:
Command  : ocsconfig -Q
UID      : 0
User name: SKOREY
Time     : Wed Oct 18 12:36:06 1995
OCS daemon state: Waiting for connections     Active nodes = 2
RTYs            : Configured RTYs = 9         Maximum RTYs = 256
OCS object name : ocsnode1
  OCS node name : risc473                     TCP/IP address = 9.12.4.102
  RTYs          : Configured RTYs = 5         Open RTYs = 0
  State         : Connected
  Debug flags   : C8300000
  Terminal devices
  ------------------------------------------------------------------------------
  0     /dev/sty00                            /dev/ocstty0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  1     /dev/sty01                            /dev/ocstty1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  2     /dev/vty2                             /dev/pts/0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  3     /dev/vty3                             /dev/pts/1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  4     /dev/vty4                             /dev/pts/2
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
OCS object name : ocsnode2
  OCS node name : risc71                      TCP/IP address = 9.12.1.71
  RTYs          : Configured RTYs = 4         Open RTYs = 0
  State         : Connected
  Debug flags   : C8300000
  Terminal devices
  ------------------------------------------------------------------------------
  10    /dev/sty10                            /dev/ocstty0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  11    /dev/sty11                            /dev/ocstty1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  20    /dev/vty20                            /dev/pts/0
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
  21    /dev/vty21                            /dev/pts/1
        User =            UID = 0             Sent = 0          Pending = 0
        RTY states = Closed
```

Figure 91. Results from the ocsconfig -Q Command

7.4 AIX OCS Reconfigured without Notification of OpenEdition/MVS OCS

- **Notification or detection of the error**
  - The AIX system administrator adds an additional serial terminal to the OCS server. The new terminal, ocstty2, is configured via SMIT and has a `ts_getty` process assigned. But when selecting `wtscsl2` from the login screen, the following message results on the serial terminal's screen:

```
Unable to reach Login Manager on wtscsl2 ...
```

- **OpenEdition/MVS procedures**
  - Browse or edit the /etc/system and validate the stanza for the OCS server (ocsnode2).
  - Browsing the /etc/system file resulted in the following display:

```
################################################################################
# ===========
# First stanza:   OCS object name = ocsnode1
# ===========    -------------------------
ocsnode1:
      Device_Description    = †OCS1 - Channel-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc473
      Automatic             = yes
      MVS_Device_Char_Minor = †[0-4]†
      MVS_Device_Char_Files = †sty0[0-1]†    , \
                              †vty[2-4]†
      Ocs_Device_Char_Files = †ocstty[0-1]†  ,  \
                              †pts/[0-2]†
# ============
# Second stanza:   OCS object name = ocsnode2
# ============    -------------------------
ocsnode2:
      Device_Description    = †Ocsnode2 - Lan-attach interface†
      Device_Type           = OCSRTY
      Ocs_Node_Name         = risc71
      Automatic             = yes
      MVS_Device_Char_Minor = †[10-11]†, †[20-21]†
      MVS_Device_Char_Files = †sty[10-11]†    , \
                              †vty[20-21]†
      Ocs_Device_Char_Files = †ocstty[0-1]† , \
                              †pts/[0-1]†
######## End of Configuration File ###########################################
```

Figure 92. Results from Browsing the /etc/system File

  - Notice in Figure 92 that we have only configured two stys (sty10 and sty11). This configuration will only allow ocstty0 and ocstty1 from ocsnode2 to connect to the OpenEdition/MVS system.

- **AIX procedures**
  - Use SMIT to display the defined and available ocstty terminals.
    1. Invoke SMIT on the OCS server by entering smit.
    2. On the System Management menu, select **OCS Support**.
    3. On the OCS Support menu, select **OCS TTY**.
    4. On the OCS TTY menu, select **List All Defined OCS TTYs**.
  - This will result in the following SMIT panel being displayed:

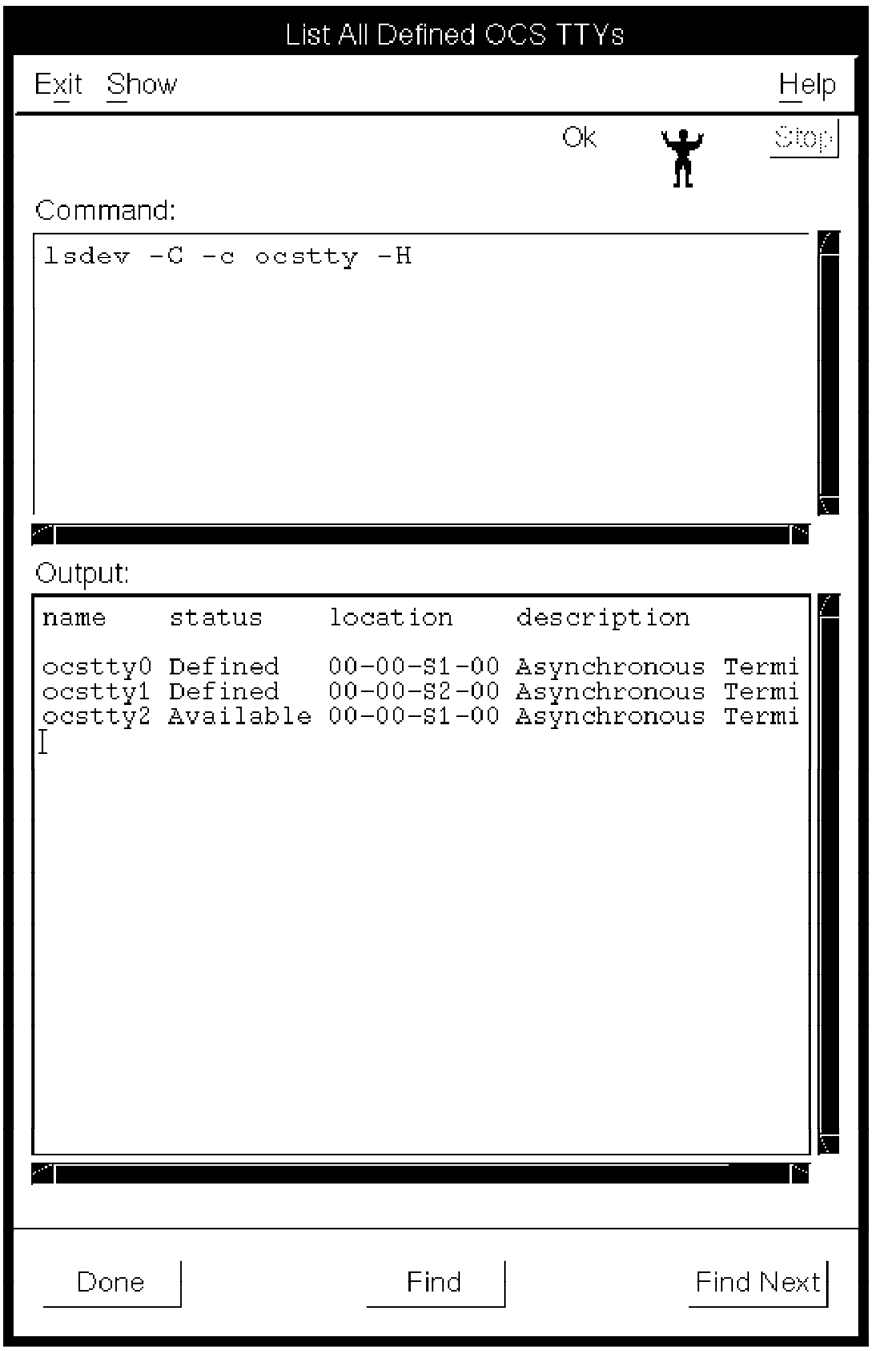


Figure 93. SMIT Output Panel

– Verify the definitions for your ocsttys. Then, check the OpenEdition/MVS system configuration for a proper match of ocsttys to stys.

Note

In our limited configuration we only had two serial terminals to attach. This scenario required that we change ocstty0 and ocstty1 to *defined*. In your environment, the SMIT display would normally indicate all three or more serial terminals as *available*.

- **Solution**
  - On OpenEdition/MVS, edit the /etc/system file to add the additional range for the serial terminals. For details see 4.3.1, “Adding and Deleting a Terminal” on page 33.
  - Issue the `ocsconfig -c ocsnode2` command to accept the changes to the /etc/system file.
  - Recycle Login Manager (lm) with the following command:

    `kill -HUP [pid number of lm]`

    See 4.3.1, “Adding and Deleting a Terminal” on page 33 for more detail on recycling lm.

Chapter 8. OCS Diagnostics

This chapter describes the techniques for obtaining information on both the OpenEdition/MVS host and the RISC System/6000 server machine to help diagnose OCS problems.

Depending on the type of problem, you may need to collect and review diagnostic information at either—or both—systems to resolve the problem.

8.1 Diagnostics on the OpenEdition/MVS Host System

- **The `ocsconfig` command with the query (`q` and `Q`) options**

  Use the `ocsconfig` command to query the state of the OCS daemon, individual OCS nodes (whether or not they are connected), and OCS terminal states.

  Attention

  The OCS daemon state must be `Waiting for Connections` before any connections to OCS nodes can be made.

  The OCS node state must be `Connected` before any terminal login requests can occur.

  For more details on the `ocsconfig` command, refer to the *OpenEdition MVS Communications Server Guide*, SC23-3883-00.

- **Syslog entries**

  The login monitor records significant error conditions at the syslog. The entries have a facility indicator of **LOG_DAEMON** and a tag string of **lm**.

- **Logon monitor messages**

  During initialization, the login monitor sends error messages to **stderr**. These messages are prefixed with **FOMC**.

- **MVS operator messages**

  The OCS daemon sends messages to the MVS console for significant error conditions. OCS messages are prefixed with **BPXB**.

- **CTRACE support**

  MVS/ESA IPCS provides CTRACE support for the OpenEdition MVS component. OCS provides the **DEVRTY** CTRACE filter to record and analyze remote-terminal device driver and TBM events within the OpenEdition/MVS component.

  Refer to *MVS/ESA Planning:OpenEdition MVS*, SC23-3015-02 for information about using CTRACE with OpenEdition/MVS.

- **IPCS formatting support**

  OCS provides IPCS formatting support for significant OCS remote-tty and TBM control structures that reside in the OpenEdition/MVS kernel.

  For more information, see the following:

  – *MVS/ESA Interactive Problem Control System (IPCS) Commands*, GC28-1491 for the syntax of the IPCS OMVSDATA command.

– *MVS/ESA Diagnosis: Procedures*, LY28-1844 for general determination procedures.

8.2 Diagnostics on the RISC System/6000 Server

- **Trace support**

  You can use the existing AIX `trace` command to trace significant AIX events. Additionally, OCS support provides the `ocsttrc` command to trace OCS-unique events.

- **Error log records**

  OCS support provides error log records to record significant error conditions.

- **Syslog records**

  The `telnet` and `rlogin` daemons generate syslog error and warning records for significant OCS events.

- **LED display messages**

  LED values are displayed if errors are detected during the configuration of OCS resources.

For more information on describing the OCS diagnostic facilities, see:

- *OpenEdition MVS Communications Server Guide*, SC23-3883-00 for “Using and Interpreting the OCS Trace” and “Interpreting OCS Error Log and Syslog Records”

For a complete description of AIX diagnostic procedures, refer to

- *AIX Version 4.1 Problem Solving Guide and Reference*, SC23-2606

Appendix A. OpenEdition/MVS and OCS Code Page Translation

This appendix describes the character encoding considerations that are specific to the OpenEdition/MVS environment.

A.1 EBCDIC Character Encoding

Probably the most significant consideration working with AIX and OpenEdition/MVS systems is that MVS uses a completely different set of hexadecimal character representations than AIX systems. There is a requirement at times to translate between the two different systems and between variations of each. Often this is done automatically; sometimes it is a programmer's responsibility.

Our experience was that all the required tools, tables and information are available. Programmers who are porting data to OpenEdition MVS need to keep a clear head as to where they are and who is responsible for translation. When debugging, it is always worth asking yourself if the problem could be an ASCII/EBCDIC issue first.

A.1.1 Background

Given eight-bits to a byte, then there are 256 different possible bytes available for representing characters. Each different byte can then arbitrarily be assigned to a particular character. This process is called character encoding (or decoding).

IBM defined EBCDIC (Extended Binary-Coded Decimal Interchange Code) as one particular character encoding scheme for use in its computers, and the American National Standards Institute (ANSI) defined a different code called ASCII (American National Standard Code for Information Interchange).

All AIX and PC systems use ASCII in one form or another, but IBM mainframes, among others, continue to use EBCDIC. Therefore, when porting from any AIX platform to OpenEdition/MVS, ASCII and EBCDIC dependencies must be taken into account. This is a unique consideration for OpenEdition/MVS and something to which even experienced porters will not be accustomed.

A.1.2 Code Pages

A code page for a specific character set determines the graphic character produced for each hexadecimal encoding. The code page used is determined by the programs and national languages being used. For internal processing, OpenEdition/MVS uses EBCDIC. To be specific, it uses the character set in the *EBCDIC Latin 1/Open Systems Interconnection Code Page 01047*. Any text to be used in OpenEdition/MVS shell processing must be converted to code page 01047. Depending on its origins, it could be in any one of a number of different code pages, both ASCII and EBCDIC. For example, a tar file would normally be stored using an ASCII code set. See *OpenEdition/MVS User's Guide* for more information.

A.1.3 Working in the Shell When You Have Different Code Pages

Differences between the code page used internally by OpenEdition/MVS and the code page used by a workstation or 3270 terminal can cause problems in entering and displaying certain characters.

Characters entered at a non-3270 workstation are normally translated by the 3270 emulation program into the CECP used by the MVS system. Any differences between the CECP and Code Page 01047 (the code page used by OpenEdition/MVS; see A.1, “EBCDIC Character Encoding” on page 103) need to be allowed for. In our case the CECP used was *U.S./Canada Country Extended Code Page 00037.* There are four characters with different representations. They are:

- Left square bracket ([)
- Right square bracket (])
- Circumflex (caret) (^)
- Not symbol (¬)

The square bracket characters are significant in both C code and shell scripts. It is therefore important that:

1. They are displayed correctly at the workstation being used.
2. The user has the ability to enter them from the workstation.
3. Printed output shows them correctly.

It is recommended that you determine the CECP used in your MVS installation (ask your friendly MVS systems programmer) and the differences between it and Code Page 1047.

A.1.3.1 Customizing a Terminal Emulator

When using workstations as 3270 emulators, we needed to change the code page used by the emulator so that it converted from the ASCII code of the workstation to the correct EBCDIC code for OpenEdition/MVS. For example, Communications Manager/2 for OS/2 has the facility to use user-defined code pages. These are simply created by editing any IBM supplied code page. This process ensures that the correct characters will be displayed at the workstation. It may also be necessary to use the keyboard mapping functions of the workstation software to allow square brackets to be entered. Again, for example, Communications Manager/2 has such a function. It may also be necessary to check font tables if printing on your workstation.

A.1.3.2 Square Bracket Solution

As shown in Figure 94 on page 105, the translation and mapping of the square brackets within code page 0037 does not conform to the mapping that is defined in code page 1047: the brackets are located on positions X¢AD¢ and X¢BD¢ in code page 1047 and are mapped to the • (Y acute) and the ¨ (umlaut) in code page 0037. The actual translation into the character actually displayed is done at the workstation according to the setting of code pages. In our implementation we used a PS/2 running OS/2 2.1 and CM/2 1.1.

```
pwd=/: >date -g
FSUM9380 date: Unknown option rcv -g
FSUM9299 Usage: date •-cu¨ •+format¨
 date •-cut¨ <date_time>
pwd=/: >
```

Figure 94. Square Bracket with Code Page 0037

Note: Though we show in Figure 94 the ¨ it was actually not shown on our system due to the final translation from 37 to 437. In fact, the ″ (quote mark) was shown.

The OMVS command sends the 3270 data stream according to code page 1047 to the workstation, so the final translation has to be corrected. We provide a short overview of the process.

1. Extract the code pages using ACSGCCRT.
2. Copy the code page 037 contained in file 037.cpt as a "user code page." A user code page must have a number in the range from 65280 to 65534.
3. The square brackets are located on X¢AD¢ and X¢BD¢ corresponding to lines 174 and 191 respectively. The translations are defined symbolically, that means "Right_Bracket" designates the "]" character.
4. We replaced current translations of [and] located at X¢BA¢ and X¢BB¢ into "Undefined." The translations in line 174 and 191 were replaced with the symbolic names for the brackets.

After these actions, the square brackets were displayed correctly, as you can see in Figure 95.

```
pwd=/u/hdm: >date -g
FSUM9380 date: Unknown option rcv -g
FSUM9299 Usage: date [-cu] [+format]
 date [-cut] <date_time>
pwd=/u/hdm: >
```

Figure 95. Square Bracket with a Modified 037 Code Page

A.1.4 Variant Characters

The POSIX Portable Character Set (PPCS) identifies the core set of 128 characters that are needed to write code and run applications. Of these, 13 characters are variant among the EBCDIC coded character sets. Table 1 lists these 13 characters and shows how the character mapping in the code pages is actually done.

Table 1 (Page 1 of 2). Mappings of the 13 PPCS Variant Characters

Character	Open Systems Hex Value (Default)	Open System IBM-1047 view	APL IBM-293 view	Inter-national IBM-500 view	France IBM-297 view	Germany IBM-273 view	US/Can IBM-0037 view
left bracket	AD	[	]	•	•	•	•
right bracket	BD	]	[	ü	~	ü	¨
left brace	C0	{	{	{	é	ä	{
right brace	D0	}	}	}	è	ü	}

Table 1 (Page 2 of 2). Mappings of the 13 PPCS Variant Characters

Character	Open Systems Hex Value (Default)	Open System IBM-1047 view	APL IBM-293 view	Inter-national IBM-500 view	France IBM-297 view	Germany IBM-273 view	US/Can IBM-0037 view
backslash	E0				ç	Ö	
circumflex	5F	^	¬	^	^	^	¬
tilde	A1	~	~	~	ü	ß	~
exclamation mark	5A	!	!	]	§	Ü	!
pound (number sign)	7B	#	#	#	£	#	#
vertical bar	4F	|	|	!	!	!	|
accent grave	79	`	`	`	µ	`	`
dollar sign	5B	$	$	$	$	$	$
commercial "at"	7C	@	@	@	á	§	@

In an environment working with a codepage mentioned in Table 1 on page 105, special care must be taken when copying data containing source type data into the HFS.

A.1.4.1 Using the Convert Option on the OMVS Command

By default, the OMVS command uses a null character conversion table, (BPXFX100). This does no translation between code pages. The OMVS command has a CONVERT option that lets you specify a conversion table for converting between code pages. The table you want to specify depends on the code pages you are using in MVS and in the shell. For example, if you are using code page 0037 on your MVS system and code page 1047 in the shell, specify the following when you enter the OMVS command:

```
OMVS CONVERT((BPXFX111))
```

Conversion table BPXFX111 will display square brackets correctly for operations that are performed in the shell. For example, square brackets will be displayed correctly in a file that is processed with the `cat` shell command. Using the `ed` editor from the shell will also display the square brackets correctly if the correct CONVERT option is used. This technique works from both 3270 type terminals and workstations running a 3270 emulator. For more information, see the OMVS command description in *OpenEdition MVS: Command Reference*.

A.1.5 Summary

Translation to and from EBCDIC is a requirement unique to porting to some selected platforms including OpenEdition/MVS. However, conceptually it is no different than translating between different ASCII character representations. So long as the application logic does not rely on a particular encoding for any character, then EBCDIC translation need not be a major issue.

Attention should be paid at installation time to ensuring that all required characters can be correctly entered, displayed and printed from all workstations.

Appendix B. CLAW Tuning

The overall performance of both nodes are workload and event-independent. Many factors can constrain communication. Basic guidelines for better data throughput are given below.

Overall system considerations:

- Use user-dedicated devices on the channel instead of a multidrop configuration.
- Ensure that the channel is operating at 4.5 MB per second in data streaming mode.
- When setting up the device driver, set the number of transmit and receive buffers to 26 each, with a size of 4096.

TCP/IP protocol considerations:

> **Attention**
>
> Be careful when changing the window size, because this can affect the performance of other LAN adapters.

- For the TCP/IP protocol suite, set the window size to 32KB on both the workstation and the VM and MVS modes. In the workstation, issue the following commands to set the window size:

  ```
  no -o tcp_sendspace=32768
  no -o tcp_recvspace=32768
  ```

- The above changes are not permanent and are reset by the next reboot. Update the rc.tcpip file to make the changes permanent. The optimal size for the Block Multiplexer Channel is 32KB. The system default is 12KB, which is optimal for LANs.

  On the MVS host, add the following line to define the window size in the profile TCP/IP file:

  ```
  DATABUFFERPOOLSIZE  300  32768
  LARGEENVELOPEPOOLSIZE  300
  ```

  where `300` is the number of buffers, and `32768` is the buffer size.

- Set the MTU size on both the System/390 and workstation to 4096.

  On the workstation, use the `ifconfig` command with the `MTU` parameter to set the MTU size to 4096. On the VM or MVS mainframe, set the gateway statement in the TCP/IP profile to an IP datagram size of 4096.

  On the 390 Host side, use the Gateway statement to set the packet size to 4096.

- A DASD Blocksize of 23440 bytes is recommended.
- When in FTP, enter `quote site ncp=20` to set the disk I/O buffers.

Appendix C. OCS Experiences

While installing and using OCS and associated programs, you may encounter some problems. This chapter provides some helpful information and hints that can help you solve these problems.

Each section describes the problems with a symptom (message, reason, scenario, and so forth) and then gives the possible solution.

C.1 OpenEdition/MVS Problems

This section describes the problem encountered with the OpenEdition/MVS component of OCS.

Symptom: Login Monitor not establishing POSIX locale correctly.

If the Login Monitor is invoked with a locale other than “POSIX” or “C” then user login requests will fail indicating that RACF authorization for the user login request failed.

Solution:

The Login Monitor was changed to correctly establish itself in the POSIX locale.

Apply the PTF fix for APAR OW15612.

Symptom: Could not correctly alter the special characters ([,]) in the /etc/system file using oedit on the OpenEdition/MVS system.

Solution:

Please see section Appendix A, “OpenEdition/MVS and OCS Code Page Translation” on page 103 for a complete description for correctly setting the code page variables. Our system variables were set as follows:

Component	Code Page Setting
MVS/ESA SP 5.2.2	En_US.IBM-037
OpenEdition/MVS	En_US.IBM-037

We also started our OpenEdition/MVS shell with the following `convert` parameter:

```
omvs convert((bpxfx437))
```

Our workstation was an IBM PS/2 Model 80, running OS/2 Version 2.1.0 with Communications Manager/2 Version 1.10. The keyboard was configured for code page 0037 (US). We had to run the Keyboard Remap Utility to modify the keyboard to include the left bracket ([) and right bracket (]) support.

C.2 OCS Problems

This section describes the problems encountered with the OCS server component of OCS.

Symptom: Can´t configure OCS host nodes on the server (AIX 4.1.3) system if device major number = 30 is already in use.

The configuration of the OCS administrative device, /dev/ocsadmin, was using device major=30, minor=0 for this device. If this device was already in use on the OCS server system, then the configuration of the first OCS host node would fail. This is a known bug.

Solution:

Support was changed to dynamically obtain a device major & minor number of the OCS administrative device, using AIX getmajor() and getminor() services, respectively.

Apply the PTF fix for APAR OW16220.

Symptom: RISC System/6000 system hangs with the status indicator displaying X¢888¢.

Altering the 'Number of Hash Sets for the Rty Table' (rhash) causes the OCS Server and the RISC System/6000 system to hang with the status indicator displaying X¢888¢. This is a known bug.

Solution:

Apply the PTF fix for APAR OW16220.

C.3 AIX/6000 V4R1.3 Problems

This section describes the problems encountered with the AIX/6000 Version 4 Release 1.3 system running the OCS server component of OCS.

Symptom: No syslog records found for rlogin and telnet connection errors.

The rlogin and telnetd servers were corrected to issue syslog records if OCS connections could not be established.

Solution:

The syslog records are documented in the OCS Guide.

Apply the PTF fix for APAR IX53666

C.4 TCP/IP V3R1 Problems

This section describes the problems encountered with TCP/IP Version 3 Release 1.

Symptom: Not able to establish terminal connections from the OCS server attached terminals to the OpenEdition/MVS system.

Solution:

Needed to assign the **TINYDATABUFFERPOOLSIZE** parameter in the TCP/IP profile.

See section 4.2.3, “Defining the Interface to MVS TCP/IP” on page 28 for more details.

C.5 TCP/IP Name Server Problems

This section describes the problems encountered with TCP/IP Name Server.

Symptom: Not able to respawn the ts_getty process on the CLAW-attached OCS System which would re-establish the serial terminal connections to the OpenEdition/MVS system after exiting from the OMVS shell.

Solution:

Needed to add the CLAW IP address (9.12.4.102) as a network in the Routing Table using the command:

```
route add -net 9.12.4.102 9.12.1.32
```

This allows the serial terminal using TCP/IP Socket Services to disconnect from the CLAW IP Socket Interface at exit time.

Glossary

This glossary defines terms and abbreviations used in this book. If you do not find the term you are looking for, refer to the index portion of this book or to *IBM Dictionary of Computing* (New York: McGraw-Hill, 1994), ISBN 0-07-031488-8 (North American hardcover, $39.50), ISBN 0-07-031489-6 (North American paperback, $24.95), ISBN 0-07-113383-6 (international).

This glossary includes terms and definitions from:

- *IBM Dictionary of Computing* (New York: McGraw-Hill, 1994).
- *Information Technology—Portable Operating System Interface (POSIX),* from the POSIX series of standards for applications and user interfaces to open systems, copyrighted by the Institute of Electrical and Electronics Engineers (IEEE). Copies of all POSIX drafts and standards may be purchased from IEEE at 1-800-678-IEEE.
  - Definitions identified by [*POSIX.0*] are from *Part 0: Standards Project, Draft Guide to the POSIX Open System Environment,* P1003.0 Draft 15 (June 1992), an unapproved draft subject to change.
  - Definitions identified by [*POSIX.1*] are from *Part 1: System Application Program Interface (API)* [*C Language*], approved September 28, 1990, as IEEE Std 1003.1-1990 by the IEEE Standards Board, and adopted in 1990 as an International Standard (ISO/IEC 9945-1: 1990) by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).
  - Definitions identified by [*POSIX.2*] are from *Part 2: Shell and Utilities,* P1003.2.
- *American National Standard Dictionary for Information Systems,* ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018. Definitions are identified by the symbol [*A*] after the definition.
- *Information Technology Vocabulary,* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions from published sections of these vocabularies are identified by the symbol [*I*] after the definition. Definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol [*T*] after the definition, indicating that final agreement has not yet been reached among the participating national bodies of SC1.
- *CCITT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the International Telecommunication Union, Geneva, 1978. These are identified by the symbol [*CCITT/ITU*] after the definition.
- Open Software Foundation (OSF). These are identified by the symbol [*OSF*] after the definition. Copies of OSF documents may be obtained from Open Software Foundation, Inc., 11 Cambridge Center, Cambridge, MA 02142.

Sequence of Entries: For clarity and consistency of style, this glossary arranges the entries alphabetically on a letter-by-letter basis. In other words, only the letters of the alphabet are used to determine sequence; special characters and spaces between words are ignored.

Organization of Entries: Each entry consists of a single-word or multiple-word term or the abbreviation or acronym for a term, followed by a commentary. A commentary includes one or more items (definitions or references) and is organized as follows:

1. An item number, if the commentary contains two or more items.
2. A usage label, indicating the area of application of the term, for example, “In programming,” or “In TCP/IP.” Absence of a usage label implies that the term is generally applicable to the OpenEdition MVS interface, to IBM, or to data processing.
3. A descriptive phrase, stating the basic meaning of the term. The descriptive phrase is assumed to be preceded by “the term is defined as ...” The part of speech being defined is indicated by the opening words of the descriptive phrase: “To ...” indicates a verb, and “Pertaining to ...” indicates a modifier. Any other wording indicates a noun or noun phrase.
4. Annotative sentences, providing additional or explanatory information.
5. References, directing the reader to other entries or items in the dictionary.
6. A source label—for example, [*A*], [*I*], [*T*], [*CCITT/ITU*], [*OSF*], [*POSIX.0*], [*POSIX.1*], or [*POSIX.2*]—that follows the definition and identifies the originator of the definition. Definitions without source labels are IBM definitions.

References: The following cross-references are used in this glossary:

Contrast with. This refers to a term that has an opposed or substantively different meaning.

Synonym for. This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with. This is a backward reference from a defined term to all other terms that have the same meaning.

See. This refers you to multiple-word terms that have the same last word.

See also. This refers the reader to related terms that have a related, but not synonymous, meaning.

Deprecated term for or **Deprecated abbreviation for.** This indicates that the term or abbreviation should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

Selection of Terms: A term is a word or group of words to be defined. In this glossary, the singular form of the noun and the infinitive form of the verb are the terms most often selected to be defined. If the term may be abbreviated, the abbreviation is given in parentheses immediately following the term. The abbreviation is also defined in its proper place in the glossary.

A

AIX. Advanced Interactive Executive.

AIX operating system. IBM′s implementation of the UNIX operating system. The RISC System/6000 system, among others, runs the AIX operating system.

alias address. An alternate address for a network interface that can be used in place of the real address.

American National Standard Code for Information Interchange (ASCII). The code developed by ANSI for information interchange among data processing systems, data communications systems, and associated equipment. The ASCII character set consists of 7-bit control characters and symbolic characters. [OSF] See also *extended binary-coded decimal interchange code.*

ASCII. American National Standard Code for Information Interchange.

B

block multiplexer channel (BMX). A multiplexer channel that interleaves blocks of data.

C

channel. A functional unit, controlled by the processor, that handles the transfer of data between processor storage and local peripheral equipment.

channel-to-channel adapter. A hardware device that can be used to connect two channels on the same computing system or on a different system.

character special file. (1) A special file that provides access to an input or output device. The character interface is used for devices that do not use block I/O. (2) A file that refers to a device. One specific type of character special file is a terminal device file. Other character special files have no structure defined by POSIX.1, and their use is unspecified by POSIX.1. [POSIX.1] The only character special file supported by OpenMVS is the pseudo-TTY. (3) See also *special file.*

CLAW. Common link access to workstations.

common link access to workstations (CLAW). A protocol supporting a RISC System/6000 attached by channel to an S/390 system. The CLAW protocol improves host system performance by reducing the number of I/O interrupts to the host.

Communications Server. A RISC System/6000 connected to one or more OpenEdition MVS systems, allowing terminals on asynchronous ports of the RISC System/6000 to operate as if they were connected directly to the OpenEdition MVS system. Synonymous with OCS.

configure. To describe to a system the devices and optional features installed on a system.

D

daemon. In the AIX operating system, a program that runs unattended to perform a standard service. Some daemons are triggered automatically to perform their service; others operate periodically.

device driver. A collection of subroutines that control the interface between I/O device controllers and the processor.

dynamic configuration. The addition of a subsystem into a running kernel.

E

EBCDIC. Extended binary-coded decimal interchange code.

extended binary-coded decimal interchange code (EBCDIC). A coded character set consisting of 8-bit coded characters. [A] See also *American National Standard Code for Information Interchange (ASCII).*

F

file. (1) A set of related records treated as a unit. (2) A sequence of records. If the file is located in internal storage, it is an internal file; if it is on an input/output device, it is an external file. [OSF] (3) A collection of related data that is stored and retrieved by an assigned name. [OSF] (4) Linear data that can be opened, written, read, and closed. A file can also contain information about the file, such as authorization information. The name used to obtain a file includes the directories in the path to the file. (5) Strings of characters with no additional structure. Structure is assumed only by the processing programs. Files can be located relative to the current directory or by an absolute pathname. (6) An object that can be written to, or read from, or both. A file has certain attributes, including access permissions and type. File types include regular file, character special file, block special file, FIFO special file, and directory. Other types of files may be defined by the implementation. [POSIX.1] In the OpenEdition implementation, the file system does not support block special files, but it does support symbolic link files. (7) A collection of information or data that is organized by some method (relative, indexed, or serial, for example) and stored on a device such as a disk. (8) Synonym for *data set.* (9) In word processing, synonym for *document.*

FMID. Function modification identifier.

function identifier. Any combination of letters, symbols, or numbers used to identify function keys.

G

getty. A process spawned by the login monitor to establish the initial terminal settings for OCS serial-line terminals. Once the initial terminal setup is complete, getty processes log in to the system.

H

hierarchical file. See *file*.

I

internet. A collection of packet-switching networks that are physically interconnected by Internet Protocol (IP) gateways. These networks use protocols that allow them to function as a large, composite network.

Internet. A wide area network connecting thousands of disparate networks in industry, education, government, and research. The Internet network uses TCP/IP as the standard for transmitting information.

Internet address. The numbering system used in TCP/IP Internetwork communications to specify a particular network or a particular host on that network which which to communicate. Internet addresses are commonly denoted in dotted decimal form.

Internet protocol (IP). A protocol used to route data from its source to its destination in an Internet environment.

internetwork. Any wide area network connecting more than one network.

I/O Configuration Program (IOCP). A program that defines to a system all available I/O devices and channel paths.

Note: The configuration program is available in three versions: stand-alone, VM/370, and MVS.

L

LAN. Local area network.

local area network (LAN). (1) A network in which communication is limited to a moderate-sized geographic area (1 to 10 km) such as a single office building, warehouse, or campus, and which does not generally extend across public rights-of-way. [OSF] A local network depends upon a communication medium capable of moderate to high data rate (greater than 1Mbps), and normally operates with a consistently low error rate. (2) A data network located on the user's premises in which serial transmission is used for direct data communication among data stations. [T]

login. Processing consisting of obtaining and validating the user's password, setting up the appropriate shell environment, and running the OpenEdition MVS shell. A login is either run from the getty process (for serial-line terminals) or spawned from the login monitor (for virtual terminals).

login monitor. A daemon that listens for login requests on any OpenEdition MVS device that corresponds to a device on the OCS server.

When a login request is received from an OCS serial-line terminal, a *getty* process is spawned to set

the correct terminal attributes and facilitate a login to the OpenEdition MVS shell.

When a login request is received from an OCS *telnet* or *rlogin* user, a *login* process is spawned to facilitate a login to the OpenEdition MVS shell.

There is one login monitor daemon for each OpenEdition MVS system.

M

multiplexer channel. A channel designed to operate with a number of I/O devices simultaneously. Several I/O devices can transfer records at the same time by interleaving items of data.

MVS/ESA. Multiple Virtual Storage/Enterprise Systems Architecture

N

network. (1) A collection of data processing products that are connected by communication lines for information exchange between locations. [OSF] (2) A connection between hosts that allows them to exchange information. [OSF] UNIX networks include: circuit, packet-switched, Ethernet, long-haul, short-haul, ProNET, and token ring. (3) An interconnected group of nodes. (4) A system consisting of two or more interconnecting computing units. (5) In data processing, a user-application network.

node. (1) An endpoint of a link, or a junction common to two or more links in a network. Nodes can be processors, controllers, or workstations, and they can vary in routing and other functional capabilities. [OSF] (2) In a tree structure, a point at which subordinate items of data originate. [A] (3) In SNA, an endpoint of a link or a junction common to two or more links in a network. Nodes can be distributed to host processors, communication controllers, or terminals. Nodes can vary in routing and other functional capabilities. (4) In ACF/VTAM, a point in a network defined by a symbolic name.

O

object data manager (ODM). In the AIX operating system, a data manager intended for the storage of system data.

OCS. Outboard Communications Server. See *Communications Server*.

OCS administrative driver. The OCS administrative driver provided with OpenEdition MVS that supports access by user administrative programs (commands) to query the active OCS configuration.

OCS system. A combination of one or more OpenEdition MVS systems and attached AIX Communications Servers.

ODM. Object data manager

OpenEdition. Pertaining to IBM services that support an environment in which operating systems, servers, distributed systems, and workstations share common interfaces and standard application development across multi-vendor systems.

OpenEdition environment. An environment that includes existing MVS standards, OpenEdition system services, DFSMS/MVS support, MVS/DCE, AD/Cycle LE/370, and other programs that fulfill IBM′s statement of direction concerning openness.

OpenEdition MVS. MVS/ESA services that support an environment within which operating systems, servers, distributed systems, and workstations share common interfaces. OpenEdition MVS supports standard application development across multivendor systems. It is required if you want to create and use applications that conform to the POSIX standard. OpenEdition MVS combines the personal power of the workstation, the flexibility of open systems, and the strength of MVS. It supports and fosters a super-environment of larger operating systems or servers and of distributed systems and workstations that share common interfaces. Users can switch back and forth between the traditional TSO/E interface and the OpenEdition shell interface. UNIX-skilled users can interact with the system, using a familiar set of standard commands and utilities. MVS-skilled users can interact with the system, using familiar TSO/E commands and interactive menus to create and manage hierarchical file system files and to copy data back and forth between MVS data sets and files. Application programmers and users have both sets of interfaces to choose from and, by making appropriate trade-offs, can choose to mix these interfaces.

Outboard Communication Server (OCS). See *Communications Server*.

P

port. An entrance to or exit from a network.

port number. The part of a socket address that identifies a port within a host.

Portable Operating System Interface. POSIX.

POSIX. Portable Operating System Interface for Computer Environments, an interface standard governed by the IEEE. POSIX is not a product. Rather, it is an evolving family of standards describing a wide spectrum of operating system components ranging from C language and shell interfaces to system administration.

pseudoterminal. A special file in the **/dev** directory that effectively functions as a keyboard and display device. Synonymous with pseudo-tty.

pseudo-TTY. Synonym for *pseudoterminal*.

R

RTY device driver. The rty (remote-tty) device driver presents a standard general terminal interface (XPG4) to the shell and application programs.

S

session. The period of time during which the user of a terminal can communicate with an interactive system, usually elapsed time between logon and logoff.

In POSIX, every process group and associated process belongs to a session. Any new process also belongs to the session of the process that created it.

SMIT. System Management Interface Tool.

special file. A file that provides an interface to input/output devices, a pipe, or a FIFO special file.

stanza. A group of lines in a file that together have a common function or define a part of the system. Stanzas are usually separated by blank lines, and each stanza has a name.

System Management Interface Tool (SMIT). An interactive interface application designed to simplify AIX system management tasks. SMIT displays a hierarchy of menus that lead to interactive dialogs.

T

TCP/IP. Transmission Control Protocol/Internet Protocol. A set of communication protocols that support peer-to-peer connectivity functions for both local and wide area networks.

Telnet. In TCP/IP, an application protocol that allows a user at one site to access a remote system as if the user′s display station were locally attached. Telnet uses the Transmission Control Protocol as the underlying protocol.

terminal. (1) A device, usually equipped with a keyboard and a display device, capable of sending and receiving information. (2) A functional unit in a system or communication network at which data may enter or leave. [T] A terminal is a type of workstation. (3) A point in a system or communication network at which data can either enter or leave. [A] (4) See also *remote terminal, workstation*.

terminal buffer manager (TBM). In OCS support, the component that collects input from or output to several terminals in a packet for forwarding.

The TBM multiplexes output from the multiple rty devices, buffers the input, and periodically sends it across the kernel socket interface to the OCS server system. It receives input from the remote OCS server through the kernel socket interface, parses it into component messages, and delivers the messages to the rty reply dispatcher for delivery to the destination rty device driver.

The TMB runs in the kernel space. For inbound data flows, the TBM runs on a kernel thread that is specifically attached to support the OCS node. (That is, there is one TBM kernel thread for each active OCS node.) For outbound data flows, the TBM runs on the user thread.

terminal solicitor. A daemon that spawns a subprocess to handle the login of each serially-attached ACSII terminal.

text file. A file that contains characters organized into one or more lines. The lines do not contain NUL characters and none exceeds {LINE_MAX} bytes in length, including the <newline>. Although POSIX.1 does not distinguish between text files and binary files, many utilities produce predictable or meaningful output only when operating on text files. The standard utilities that have such restrictions always specify *text files* in their Standard Input or Input Files subclauses. [POSIX.2]

token. (1) In a local area network, the symbol of authority passed successively from one data station to another to indicate the station temporarily in control of the transmission medium. A token is a particular message or bit pattern that signifies permission to transmit. (T) (2) A sequence of bits passed from one device to another along the token ring. When the token has data appended to it, it becomes a frame.

token ring. A network with a ring topology that passes tokens from one attaching device to another; for example, the IBM Token-Ring Network.

Transmission Control Protocol. A communications protocol used in Internet and in any network that follows the U.S. Department of Defense standards for inter-network protocol. TCP provides a reliable host-to-host protocol between hosts in packet-switched communications, networks and in interconnected systems of such networks. It assumes that the Internet protocol is the underlying protocol.

Tty. Any device that uses the **termios** standard terminal device interface. Tty devices typically perform input and output on a character-by-character basis.

V

virtual host address. In an OCS system, an address of the RISC System/6000 associated with the Internet address of an AIX/ESA host.

virtual terminal. (1) A generalized logical model of different terminals of a certain class, describing how terminals of that class will perform in an OSI environment. (2) In the AIX operating system, any of several logical equivalents of a display station. A virtual terminal supports the illusion that more devices exist than are physically present. Virtual terminals are logically independent of each other, but share physical resources over time. (3) In TCP/IP, a system object, created and controlled by an application program that provides a functional representation or simulation of a physical display station.

Index

Special Characters

A

B

C

D

E

F

H

I

K

L

M

N

O

P

R

S

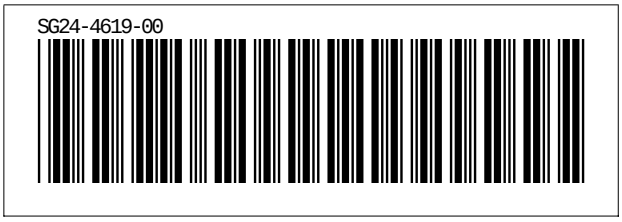
T

U

V

W

Printed in U.S.A.



Artwork Definitions

Figures

		35	26	
				34
OCSCFAF	4619CHP4			
		36	27	
				35
CONFIG2	4619CHP4			
		37	28	
				36
PSCMD	4619CHP4			
		37	29	
				37, 37, 39
UNCONFG	4619CHP4			
		38	30	
				38
REMOVE	4619CHP4			
		39	31	
				38
FTP0001	4619CHP4			
		40	32	
				39, 40
SMIT000	4619CHP4			
		42	33	
				49
SMIT001	4619CHP4			
		43	34	
SMIT002	4619CHP4			
		44	35	
SMIT003	4619CHP4			
		45	36	
SMIT004	4619CHP4			
		46	37	
				49
SMIT005	4619CHP4			
		46	38	
				49
SMIT006	4619CHP4			
		47	39	
				49
SMIT007	4619CHP4			
		47	40	
SMIT008	4619CHP4			
		48	41	
SMIT009	4619CHP4			
		49	42	
				49, 49
SMIT010	4619CHP4			
		51	43	
SMIT011	4619CHP4			
		52	44	
SMIT012	4619CHP4			
		53	45	
				56, 60
SMIT013	4619CHP4			
		54	46	
SMIT014	4619CHP4			
		55	47	
				55, 56
SMIT015	4619CHP4			
		56	48	
IFCONF1	4619CHP4			
		57	49	
				56, 60
NETADD1	4619CHP4			
		57	50	
				57
NAMED1	4619CHP4			
		58	51	
				57
CLAW001	4619CHP4			
		58	52	
				54
CLAW002	4619CHP4			
		59	53	
				59, 60
CLAW003	4619CHP4			
		60	54	
NETADD2	4619CHP4			
		61	55	
				61
NAMED2	4619CHP4			
		61	56	
				61
SMIT016	4619CHP4			
		62	57	
SMIT017	4619CHP4			
		63	58	
SMIT018	4619CHP4			

		64	59	
				65
SMIT019	4619CHP4			
		64	60	
				64, 65
SMIT020	4619CHP4			
		65	61	
				65
SMIT045	4619CHP4			
		66	62	
				75
SMIT021	4619CHP4			
		67	63	
				69
SMIT022	4619CHP4			
		68	64	
				69, 69
SMIT023	4619CHP4			
		68	65	
				68
SMIT044	4619CHP4			
		69	66	
SMIT024	4619CHP4			
		70	67	
				70
SMIT046	4619CHP4			
		71	68	
SMIT025	4619CHP4			
		72	69	
SMIT026	4619CHP4			
		73	70	
SMIT027	4619CHP4			
		73	71	
SMIT028	4619CHP4			
		74	72	
SMIT029	4619CHP4			
		74	73	
SMIT030	4619CHP4			
		75	74	
SMIT031	4619CHP4			
		76	75	
				75, 78
SMIT032	4619CHP4			
		76	76	
SMIT033	4619CHP4			
		77	77	
SMIT034	4619CHP4			
		77	78	
				77, 77
SMIT035	4619CHP4			
		78	79	
SMIT036	4619CHP4			
		79	80	
SMIT037	4619CHP4			
		80	81	
SMIT038	4619CHP4			
		80	82	
SMIT039	4619CHP4			
		81	83	
SMIT041	4619CHP4			
		82	84	
				82
SMIT042	4619CHP4			
		82	85	
ERROR1	4619CHP7			
		92	86	
ERROR2	4619CHP7			
		93	87	
				93
ERROR3	4619CHP7			
		94	88	
				93
ERROR4	4619CHP7			
		95	89	
ERROR5	4619CHP7			
		96	90	
ERROR6	4619CHP7			
		97	91	
				96
ERROR7	4619CHP7			
		98	92	
				98
SMIT047	4619CHP7			
		99	93	
SQB01	4619AX1			
		105	94	
				104, 105

SQB02	4619AX1		
		105	95 105

Headings

id	File	Page	References
NOTICES	4619FM	xiii	Special Notices ii
BIBL	4619PREF	xvi	Related Publications
INTRO	4619CHP1	1	Chapter 1, Introduction to OCS xv
BENF	4619CHP1	1	1.1, Benefits of OCS
PURP	4619CHP1	1	1.2, Purpose of This Project
REQ	4619CHP1	2	1.3, Minimum System Requirements 39
ENV	4619CHP1	2	1.4, ITSO Lab Environment
HRDWARE	4619CHP1	2	1.4.1, Hardware Environment
SFTWARE	4619CHP1	3	1.4.2, Software Environment
ENVIRN	4619CHP1	4	1.5, OCS Environment Diagram
DESC	4619CHP2	5	Chapter 2, Description of OCS xv
ATTO	4619CHP2	5	2.1.1, Attachment Options
PRE	4619CHP2	6	2.2, How Does Non-OCS Work?
POST	4619CHP2	8	2.3, How OCS Works
TERMS	4619CHP2	9	2.4, OCS Components 3
PLAN	4619CHP3	13	Chapter 3, Planning for OCS xv
MVSPLAN	4619CHP3	13	3.1, Planning for MVS/ESA OpenEdition/MVS
OEPLAN	4619CHP3	15	3.2.5, MVS/ESA OpenEdition Planning
TCPPLAN	4619CHP3	25	3.2.7, TCPIP 3.1 Plan
RISCPLN	4619CHP3	25	3.3, RISC System/6000 Planning
HWDPLN	4619CHP3	25	3.3.1, Hardware Requirements
CAPLN	4619CHP3	25	3.3.2, Channel Attachment
LANPLN	4619CHP3	25	3.3.3, For LAN Attachment
AIXPLN	4619CHP3	25	3.3.4, AIX Software Requirements
STORPLN	4619CHP3	26	3.3.5, Storage Planning
INST	4619CHP4	27	Chapter 4, Installation of OCS xv
MVSINST	4619CHP4	27	4.1, MVS Installation for OCS
MVSCONF	4619CHP4	28	4.2, MVS Configuration for OCS 39
TCPI	4619CHP4	28	4.2.3, Defining the Interface to MVS TCP/IP 111
DEFN	4619CHP4	29	4.2.4, Defining TCP/IP Hosts For OCS Nodes 95
CFILE	4619CHP4	29	4.2.5, Creating a Configuration File
STLM	4619CHP4	31	4.2.8, Starting and Customizing the Login Monitor Daemon
STOCS	4619CHP4	32	4.2.9, Starting OCS

Tables

Processing Options

Runtime values:

Document fileid	SG244619 SCRIPT
Document type	USERDOC
Document style	IBMXAGD
Profile	EDFPRF30
Service Level	0029
SCRIPT/VS Release	4.0.0
Date	96.01.14
Time	09:39:03
Device	3820A
Number of Passes	3
Index	YES
SYSVAR G	INLINE
SYSVAR V	ITSCEVAL

Formatting values used:

Annotation	NO
Cross reference listing	YES
Cross reference head prefix only	NO
Dialog	LABEL
Duplex	YES
DVCF conditions file	(none)
DVCF value 1	(none)
DVCF value 2	(none)
DVCF value 3	(none)
DVCF value 4	(none)
DVCF value 5	(none)
DVCF value 6	(none)
DVCF value 7	(none)
DVCF value 8	(none)
DVCF value 9	(none)
Explode	NO
Figure list on new page	YES
Figure/table number separation	YES
Folio-by-chapter	NO
Head 0 body text	Part
Head 1 body text	Chapter
Head 1 appendix text	Appendix
Hyphenation	NO
Justification	NO
Language	ENGL
Layout	OFF
Leader dots	YES
Master index	(none)
Partial TOC (maximum level)	4
Partial TOC (new page after)	INLINE
Print example id's	NO

Print cross reference page numbers	YES
Process value	(none)
Punctuation move characters	.,
Read cross-reference file	(none)
Running heading/footing rule	NONE
Show index entries	NO
Table of Contents (maximum level)	3
Table list on new page	YES
Title page (draft) alignment	RIGHT
Write cross-reference file	(none)

Imbed Trace

Page 0	4619SU
Page 0	4619VARS
Page 0	4619FM
Page i	4619EDNO
Page ii	4619ABST
Page xiii	4619SPEC
Page xiii	4619TMKS
Page xiv	4619PREF
Page xvii	4619ACKS
Page xviii	4619CHP1
Page 4	4619CHP2
Page 11	4619CHP3
Page 26	4619CHP4
Page 82	4619CHP5
Page 86	4619CHP6
Page 89	4619CHP7
Page 100	4619CHP8
Page 102	4619AX1
Page 106	4619AX2
Page 107	4619AX3
Page 112	4619GLOS
Page 123	4619EVAL