Unicenter® TCPaccess™ Communications Server

Planning Guide

r6.0 SP4

K02195-4E

Contents

Chapter 1: Architecture

Chapter 2: Preparing for Installation

Chapter 3: Customizing System Security

Chapter 4: UNIX System Services Support

Chapter 5: User Exits

Appendix A: Editing Tools For Installation

Appendix B: Configuring for Cisco Routers

Appendix C: Using a CDLC Driver

Index

Chapter 1

Architecture

This chapter describes the architecture of Unicenter® TCPaccess™ Communications Server. It includes the following sections:

- System Architecture—Describes the architecture of the system, including interaction with other systems
- TCP/IP Stack—Describes the layers of the TCP/IP stack.
- Transport Providers—Describes the access methods and transport providers supported by Unicenter TCPaccess
- IFS Services —Describes the IFS services in Unicenter TCPaccess, including dump and recovery, messages, SMF, latch, timing, tracing, and operator interface
- Network Features—Describes the network features of Unicenter TCPaccess
- Protocol Features—Describes the protocol features of Unicenter TCPaccess

Unicenter TCPaccess responds to the need of IBM mainframes to communicate with client systems on various platforms. Unicenter TCPaccess is a comprehensive software package providing resources and capabilities of IBM MVS mainframes to non-IBM hosts and workstations using Transmission Control Protocol and Internet Protocol (TCP/IP).

Network Features provide functions such as file transfer between an MVS host and nodes on a TCP/IP network (FTP client and server), electronic mail exchange (SMTP), bi-directional terminal emulation (Telnet support for NVT, TN3270 and TN3270E with SSL capabilities), access to IBM printers (LPR server), domain name resolution, and network management (SNMP agent).

The Unicenter TCPaccess application runs on IBM System/390 and z/OS mainframes running various releases of MVS (see *Getting Started*). The LE/370 runtime library (SCEERUN) must be either link-listed or included in the STEPLIB. If implementing Server Telnet with SSL, then SGSKLOAD and SCLBDLL must either be link-listed or included in the STEPLIB.

System Architecture

Unicenter TCPaccess runs as an MVS subsystem in its own address space with no operating system modifications. Interfaces to the external system security facility are implemented for signon, data set, and command access security using the MVS SAF router. Interprocess and inter-address space communication is accomplished using cross-memory services, ESA access registers, VTAM, and JES2/JES3. Unicenter TCPaccess is installed and maintained using SMP/E.

Unicenter TCPaccess runs within a runtime environment called the *Infrastructure* (IFS). IFS is a generic, multitasking, runtime environment for MVS system application address space that provides basic services such as cross-memory communications and storage management. A system using the infrastructure is an authorized, operator-started task or job that runs as a subsystem.

TCP/IP Stack

The TCP/IP stack is a layered set of routines which implement the various protocols to communicate on the Internet and Intranet. The layers within the TCP/IP stack are:

Transport Layer Consists of the protocol routines that implement a specific IP protocol.

This layer contains the:

- TCP protocol modules
- UDP protocol modules
- RAW modules

Although ICMP is one of the IP protocols, for this discussion, it is considered part of the Internet Layer.

Internet Layer Consists of the protocol routines to implement the IP and ICMP protocols.

Some of the functions included at this layer include:

- Choosing routes
- Creating IP protocol headers in outbound packets
- Processing and removing IP protocol headers from inbound packets
- Generating/processing of ICMP protocol messages

Link Layer	Consists of the device drivers. The drivers are responsible for the sending and receiving of data to the physical network controllers. Additionally, they build/process the media layer headers within each packet and generate and respond to address resolution messages from other hosts. Currently the drivers support Ethernet, Token Ring, FDDI ring, CLAW, Hyperchannel and Loopback type controllers.

Transport Providers

Unicenter TCPaccess supports the following access methods using three transport providers. The transport providers support access via the Transport Layer Interface (TLI), UNIX System Services (formerly OpenEdition) MVS sockets, and IUCV sockets.

All three transport providers use a common interface to the transport layer called the Socket API.

Socket API	A set of routines that logically sits above the Transport Layer to implement native sockets. It consists of: ■ Function Processing Routines—Called directly from various transport providers such as the Open Edition PFS Transport Provider and implement native sockets logic. These routines generally run under the control of the application address space. ■ Transport Layer Exit Routines—Called as exits from the TCP/IP stack. These routines perform processing for events such as write completion, new data received, connection and confirmation indications, and so forth.

Assembler API	A set of routines that logically sits above the Transport Layer to implement a variation of the AT&T Transport Layer Interface (TLI). It consists of: ■ Function Processing Routines—Called directly from various transport providers such as the Open Edition PFS Transport Provider and implement native sockets logic. These routines generally run under the control of the application address space. ■ Transport Layer Exit Routines that are called as exits from the TCP/IP stack. These routines perform processing for events such as write completion, new data received, connection and confirmation indications, and so forth.
UNIX System Services Transport Provider	A set of routines that logically is located above the Socket API and provides cross memory access from Open Edition MVS sockets applications to Unicenter TCPaccess. It consists of: ■ Function Processing Routines—Called directly from various transport providers such as the UNIX System Services PFS Transport Provider and implement native sockets logic. These routines generally run under the control of the application address space. ■ Transport Layer Exit Routines—Called as exits from the TCP/IP stack. These routines perform processing for events such as write completion, new data received, connection and confirmation indications, and so forth.
IUCV Transport Provider	A set of routines that logically is located above the Socket API and provides cross memory access from sockets applications to Unicenter TCPaccess via program calls that emulate the IUCV facility of VM. It consists of: ■ Function Processing Routines—Called directly from various transport providers such as the UNIX System Services PFS Transport Provider and implement native sockets logic. These routines typically run under the control of the application address space. ■ Transport Layer Exit Routines—Called as exits from the TCP/IP stack. These routines perform processing for events such as write completion, new data received, connection and confirmation indications, and so forth.

IFS Services

IFS Services are described next.

- Dump and Recovery Services include routines to capture dumps of Unicenter TCPaccess and any other involved address spaces (such as an application address space UNIX System Services—MVS) and provide recovery.
- Message Services include routines that write to the operator console or sysout data sets. Messages can be filtered by component and severity, either through product configuration, or an operator command.
- SMF Services includes a standard interface to write records to the SMF data set. SMF data is captured for numerous events, including FTP data transfer, Telnet session end, transport provider events, and protocol layer events. SMF Services can also be configured to capture data at a desired interval. Link layer device driver statistics and virtual storage statistics can be captured at intervals.
- Latch Services are a set of IFS routines that provide resource serialization at a level more granular than an address space. This serialization mechanism is referred to as IFS Latches, or Ilatches.
- Timing Services include routines that measure time intervals for various processes.
- Tracing services include routines and macros that keep track of the events that occur within the Unicenter TCPaccess address space. Tracing is done using the IFS internal trace table, the system trace table, through GTF and TCPEEP.
- Operator Interface is a set of various routines that allow significant operational control of the Unicenter TCPaccess address space and the TCP/IP stack.
- RTM Services collect response time data for the Unicenter TCPaccess Telnet Server. The data can be filtered and reported by the TNSTAT command, and an RTM-only address space can be used as a central collection point for all TCPaccess RTM data. For details on how to set up an RTM-only address space, see the "Running an RTM-Only Address Space" appendix in the *Unicenter TCPaccess Telnet Server Customization Guide*.

  Note: RTM Services are not available for the STELNET server.

Network Features

The network features are:

- Software interfaces for Ethernet, Token Ring, FDDI, and Hyperchannel networks
- Support for IBM Continuously Executing Transfer Interface (CETI) and Common Link Access to Workstation (CLAW) interface for high-speed network I/O
- Support for multi-homing with a virtually unlimited number of network segments
- Support for a wide variety of IBM channel-to-LAN controllers

Protocol Features

The protocol features are:

- Implementation of TCP and User Datagram Protocol (UDP) in accordance with MIL-STD 1778 and RFCs 768 and 793
- Implementation of IP and Internet Control Message Protocol (ICMP) in accordance with MIL-STD 1777 and RFCs 791 and 792
- Support of subnets in accordance with RFC 950
- Compatibility with UNIX, Macintosh, PC/DOS, Windows (95 and NT), OpenVMS, and OS/2 implementations of TN3270
- Implementation of Server Telnet with support for LU2 and LU0 3270 SNA protocols
- Provision for Telnet Server LU name support (LU security) that associates user ID and terminal access security to an individual Telnet user to use Unicenter TCPaccess in secure environments
- Implementation of Server Telnet TN3270E protocol in accordance with RFC2355 and Functional Extensions
  - Support of LU2 and LU0 3270 SNA protocols
  - Support of LU1 and LU3 SNA protocols
- Implementation of SSL protocol for Server Telnet

Chapter 2

Preparing for Installation

Before you install the Unicenter TCPaccess software, you must prepare your MVS system to accept the product. This chapter provides an overview of the major tasks required to modify the MVS operating system prior to installing Unicenter TCPaccess. These preliminary tasks are described in these sections:

- Pre-Installation Information—Identifies defining the hardware interface as a pre-installation step and refers you to the detailed instructions for configuring I/O devices to support Unicenter TCPaccess
- Determining the MVS Subsystem ID—Describes how to determine the Unicenter TCPaccess MVS subsystem ID
- Customizing the Security Interface—Identifies some considerations for customizing the security interface for RACF, eTrust CA-ACF2, or eTrust CA-TopSecret
- Authorizing Client Commands—Describes the process for authorizing client commands TCPEEP and ACCFTP2 to run as MVS programs
- Setting APF Authorization for Common Load Data Sets—Describes the process for authorizing common load data sets for Unicenter TCPaccess
- Updating the MVS Link List and Procedure Libraries—Describes two processes used for updating the LINKLIST

Pre-Installation Information

You must define the I/O devices used by Unicenter TCPaccess to gain access to the network. Get any additional information about the network interface devices directly from the network interface supplier.

Determining the MVS Dispatching Priority

For most efficient execution of Unicenter TCPaccess, set the dispatching priority the same as, or just below, VTAM.

Determining the MVS Subsystem ID

During the initialization process, Unicenter TCPaccess attempts to locate the subsystem control blocks it needs. It looks for the subsystem name in the control blocks. The default name of the subsystem is **ACSS**. You can override the default by changing the SSN=symbolic parameter in the RUNTCP JCL stream.

If you cannot locate the required subsystem control blocks, Unicenter TCPaccess builds them dynamically and places them on the MVS subsystem control block chain. Unicenter TCPaccess does not use the subsystem control blocks if they are in use by another address space.

Dynamic allocation of the subsystem control blocks is recommended. No IPL or maintenance of SYS1.PARMLIB is necessary.

Note: If you prefer to permanently define the subsystem control blocks in your installation, add an entry for the subsystem name in member IEFSSN*xx* in SYS1.PARMLIB. If you do not want to use the default name (ACSS), override the subsystem name on the SSN= parameter in the RUNTCP job stream. You must perform an IPL in order for the change to IEFSSN*xx* to take effect.

Customizing the Security Interface

In installations using external security systems, there may be data access restrictions. The security administrator must ensure that a TCP/IP implementation does not circumvent any restrictions already in place.

The SAF router provides access between Unicenter TCPaccess and the MVS security system, enabling Unicenter TCPaccess to perform functions included in FTP, FTP2, FTP3, and Telnet.

If you are using ACF2, all access is denied until explicitly permitted, requiring you to take a series of steps prior to starting Unicenter TCPaccess.

If RACF, ACF2, or CA-TOP SECRET is installed, perform at least the basic system security customization according to the steps described in the chapter "Customizing System Security."

If you are using the SSL capable Telnet server, the associated userid of the address space must have superuser privileges.

Authorizing Client Commands

These Unicenter TCPaccess client commands and their aliases can or cannot be run as authorized MVS programs and commands:

- ACCFTP2 (alias FTP2)
- FTP3
- T051C (FTP Server client, if installed)
- TCPEEP

When these commands run authorized, they extract encrypted passwords, groups, and TSO user IDs. The extracted information is used to sign on to the Unicenter TCPaccess address space on the local host.

The authorized versions significantly reduce the number of times a TSO user is prompted for an MVS user ID and password. Neither plain-text MVS passwords nor their associated TSO user IDs are sent across the TCP/IP network when the automatic signon feature of the authorized programs is used.

The automatic signon feature provides the additional benefit of not having to leave a TSO user ID and password in plain text in the batch input to these programs stored on DASD.

1. Add the command module names and their aliases to the AUTHCMD, AUTHPGM, and AUTHTSF sections of member IKJTSO*xx*.
2. Follow your installation's procedures for updating SYS1.PARMLIB members as shown in this example.

```
AUTHCMD NAMES(

   FTP
   FTP2
   FTP3
   TCPEEP
   ACCFTP2 )
AUTHPGM NAMES(
   FTP
   FTP2
   FTP3
   TCPEEP
   ACCFTP2 )
AUTHTSF NAMES(
   FTP
   FTP2
   FTP3
   TCPEEP
   ACCFTP2 )
```

3. The FTP3 authorized TSO command is in the FTPLOAD data set; all authorized TSO commands are included in the Unicenter TCPaccess LINK data set. Follow installation procedures at your site to provide access to these modules under TSO.

Setting APF Authorization for Common Load Data Sets

Unicenter TCPaccess LOAD, LINK, and FTPLOAD data sets require APF authorization. In order to set authorization for these common load data sets, modify the IEAAPF*xx* member of the SYS1.PARMLIB data set.

Modifying SYS1.PARMLIB(IEAAPFxx)

If you do not have a procedure in place for modifying PARMLIB members, use the following steps to update the SYS1.PARMLIB member IEAAPF*xx*:

1. Verify the target name and volume serial of these data sets before proceeding.
2. If you have a procedure in place for modifying PARMLIB members, follow that procedure; if you do not have a procedure in place, proceed to Step 3.
3. Create a full-back member by renaming the current IEAAPF*xx* member and giving it a backup suffix. Copy the renamed member and give it the current suffix. This provides you with a full-back member in the event an error is made during the editing process.
4. Edit the APF authorization member IEAAPF*xx* (or PROG*xx* for ESA Version 4.3 or higher) in SYS1.PARMLIB (where *xx* is the suffix of your member).
5. If you are using SSL with Telnet, then the GSK.SGSKLOAD data set must be APF authorized. If using the SSL capable server, then CEE.SCEERUN and CBC.SCLBDLL data sets must be APF authorized.
6. If you are the SSL capable Telnet, the GSK.SGSKLOAD data set must be APF authorized.
7. You must perform an IPL in order for the changes to take effect.

   Note: Some MVS monitoring packages and newer versions of MVS/ESA allow dynamic APF authorization of data sets while the MVS system is running. If you dynamically authorize APF data sets, you must still change the IEAAPF*xx* member or authorization will be lost at the next IPL.

   If you are not familiar with changing the IEAAPF*xx* member in SYS1.PARMLIB, consult the *MVS Initialization and Tuning Guide*.

WARNING! *Whenever you make changes to any SYS1.PARMLIB member, be sure you can perform an IPL of your system using an alternate IPL volume or an alternate SYS1.PARMLIB member. Typographical errors can cause catastrophic errors during system initialization, leaving your MVS system in an unusable state.*

Updating the MVS LINKLIST and Procedure Libraries

Unicenter TCPaccess software user interface programs can be executed from both batch and TSO address spaces. The following user interface programs and client commands are located in the Unicenter TCPaccess LINK library:

- FTP
- FTP2
- FTP3
- NETSTAT
- PING
- REMCMND
- TCPEEP
- TELNET
- TRACERT

To execute these programs, the LINK library must be available to batch jobs and TSO users for execution. These user interface programs also require the SASLINK library and LE runtime libraries are available for execution to the batch jobs and TSO users.

If you decide to place the LINK data set in the MVS linklist, the LINK data set must be cataloged in your master catalog. Therefore, the LNKINDX (LINK data set high-level index) must not start with a qualifier that is defined as an alias in your master catalog.

You can make the user interface programs available by modifying one of the following:

- SYS1.PARMLIB
- Batch jobs
- TSO procedures

If you are familiar with updating the linklist and your LINK library, and the LE runtime libraries are already available in the linklist, skip the rest of this section and continue with the chapter "Customizing System Security."

FTP3

The FTP3 client is more user-friendly and offers more functionality than the original FTP client. If you want to use FTP3 as your default FTP client, you can change its name to FTP by executing usermod UMFTP3A located in the SAMP library. You can still use the old FTP client under the name FTP1.

Modifying SYS1.PARMLIB(LNKLSTxx)

The most reliable way to ensure availability of user interface programs is to make the LINK data set available for execution globally. Make sure you are familiar with any policies your site has in place regarding the use of the LINK data set before proceeding with this method.

Edit the LNKLSTxx Member

If you do not have a procedure in place for modifying PARMLIB members, use the following steps to update the SYS1.PARMLIB member LNKLST*xx*:

1. Create a full-back member by renaming the current LNKLST*xx* member and give it a backup suffix.
2. Copy the renamed member and give it the current suffix. This provides you with a full-back member in the event an error is made during the editing process.
3. Edit the LNKLST*xx* (where *xx* is your local suffix) member in SYS1.PARMLIB.

WARNING! *The LOAD data set must* ***never*** *be added to the linklist. The Unicenter TCPaccess element names are not unique and therefore could affect the operation of other software. The LOAD data set should always be referenced through a STEPLIB or JOBLIB statement.*

   If you are not familiar with changing the LNKLST*xx* member in SYS1.PARMLIB, seek assistance from someone who is familiar with the process or consult the *MVS Initialization and Tuning Guide*.

4. Perform the IPL for the changes to take effect. If you do not plan to perform an IPL right away, you can change STEPLIB or JOBLIB DD statements to make the programs available for execution until the next time an IPL is done.

WARNING! *Whenever you make changes to any SYS1.PARMLIB member, be sure you can perform the IPL of your system using an alternate IPL volume or an alternate SYS1.PARMLIB member. Typographical errors can cause catastrophic errors during system initialization, leaving your MVS system in an unusable state.*

Testing the Link Data Set

If you have already modified LNKLST*xx* to add the LINK data set from a previous release of Unicenter TCPaccess, do not replace that entry with the new LINK data set until you are satisfied with the testing of the new release and are ready for migration.

Note: During testing, use JOBLIB or STEPLIB DD statements in TSO procedures, batch jobs, and the Unicenter TCPaccess job to reference the LINK data set for the new release.

Modifying Batch Jobs

You can make user interface programs available for execution by modifying the batch jobs or TSO procedures that use them.

To do so, add STEPLIB or JOBLIB DD statements for the LINK data set and LE runtime libraries to each batch job or TSO procedure that needs to execute them.

If the LINK data set and the LE runtime libraries are not available, the client commands will not work properly.

Modifying TSO Procedures

Several user interface programs can be executed from TSO address spaces. Some user interface programs have TSO help members that let the users find information on the use and format of each program. Any TSO users who need to reference these TSO help members must have their TSO procedures updated.

Note: Edit any TSO procedures that require access to TSO help members by concatenating a DD statement to SYSHELP. If TRGINDX is specified as T01TCP.V6R0, add the following DD statement to SYSHELP**:**

```
//    DD DSN=T01TCP.V5R2.HELP,DISP=SHR
```

Chapter

3

Customizing System Security

System security is an important consideration in data processing. Products like Access Control Facility 2 (eTrust CA-ACF2), eTrust CA-Top Secret, or Resource Access Control Facility (RACF) help many installations protect valuable data and preserve system integrity.

The following sections describe the security configuration procedures, as required by several security products.

Note: The examples in this chapter use default class and profile names for illustration only; alternate name selection is possible. See the description of the SECURITY statement in the IJTCFG*xx* member for details.

- Security Information in the Log File —Describes a parameter used to display information about the user signon
- Terminal Security Configuration—Describes the parameters that support the security products for the Unicenter TCPaccess terminal security or source security feature
- Types of eTrust CA-ACF2 Security—Describes the security options
- eTrust CA-Top Secret Options—Describes the eTrust CA-Top Secret security options
- RACF Options—Describes the RACF security options
- SAF Server Access Authorization (SERVAUTH)—Describes the optional SAF Server Access Authorization (SERVAUTH) options

In installations using external security systems, the security administrator usually establishes data access restrictions. The security administrator must ensure that Unicenter TCPaccess does not circumvent these restrictions.

Unicenter TCPaccess interfaces to the MVS security system, via the SAF router, to perform the following functions:

- User ID and password validation

  The user ID and password are validated when sent to Unicenter TCPaccess. Validation occurs at these points:

  - Direct signons through VTAM to Unicenter TCPaccess
  - Interface calls to the ACTEST debugging service through the VTAM interface by entering: `;VTAMTEST`
  - After connecting to FTP
  - The first time a user tries to use Server Telnet commands that are protected by external security

- User privileges verification

  FTP uses the validated user security authority to determine if the user is permitted access to specific data sets. Access to data sets is determined by the security information associated with the user, not the security information for the job. Unicenter TCPaccess optionally uses the SERVAUTH Security Access Facility (SAF) class to protect TCP/IP resources from unauthorized access.

- User authority verification to run ACTEST

  Unicenter TCPaccess validates a user's authority to execute the ACTEST debugging service through the VTAM interface.

  At eTrust CA-ACF2, and eTrust CA-Top Secret sites, the user ID associated with the Unicenter TCPaccess job needs no special privileges assigned, such as NON-CNCL, OPERATIONS, or DASDVOL authority, or PPTNOPAS specified in the Program Properties Table. In addition, the Unicenter TCPaccess user ID does not need access to user data sets for FTP to function properly.

  The user ID associated with a Unicenter TCPaccess job or started task is not allowed access to any services of Unicenter TCPaccess.

Security Information in the Log File

Security activity can be monitored by activating appropriate options, either at startup or dynamically via ACTEST. Several categories of security related events can be displayed at execution via messages T00IF070 through T00IF088. Many of these events are frequent occurrences and can quickly flood a log file.

The security categories eligible for monitoring can be initially activated via the XSEC keyword of the SECURITY statement in the IJTCFG*xx* member and can later be enabled or disabled via the ACTEST XSEC command.

The following events are eligible for monitoring:

- ACSECPC—All security calls
- COMMAND—Command authorization calls (for example, ACTEST)
- DATASET—Data set authorization calls
- LOGON—System entry attempts
- LOGOFF—System departures
- ACEE—All ACEE-associated activity

Two other global options are also in effect and are capable of totally disabling either **all** security calls, or just command authorization calls. If you disable security functions at a global level, monitoring cannot be performed. See your system administrator about selective security activation.

For example, you may need to monitor signons, signoffs, and filename accesses for a period. If the startup IJTCFG*xx* SECURITY statement contains XSEC(LOGON LOGOFF DATASET), then ACTEST can be executed with XSEC(LOGON LOGOFF DATASET OFF) after the monitoring period is over.

Alternatively, no change needs to be made to IJTCFG*xx* at startup, but ACTEST can be run specifying XSEC(LOGON LOGOFF DATASET ON). After the monitoring period, ACTEST can be again executed with the OFF option.

Configuring Terminal Security

This section describes the parameters that support the Unicenter TCPaccess terminal security or source security feature.

Terminal Security Configuration

The following parameters of the XSEC parameter on the SECURITY statement in IJTCFG*xx* member are:

- TERMID causes the Unicenter TCPaccess security interface to place a terminal ID into the Terminal field of the signon parameter list for any user attempting a signon to Unicenter TCPaccess. The terminal ID passed during signon attempts is either the remote IP address of the originating host for the user or a VTAM APPL LU name.
- NOTERMID causes the Unicenter TCPaccess security interface to not use the Terminal field in the signon parameter list during signon attempts.

Unicenter TCPaccess defaults to NOTERMID.

Note: In order to create separate VTAM resources for FTP, define an LUPOOL for FTP usage separate from the one used for Telnet. This allows different security rules to be defined for each set of LU names. See the *Customization Guide* for more information.

Telnet Signon Checking

Typically, the user ID of a Telnet user is not validated at signon because the service being accessed (typically, TSO) does validation. Sensitive commands, such as ACTEST and SYSSTAT, are validated. The following technique enables signon checking for general user access.

Add the CPASSWORD option to all SERVICE statements in the APPCFG*xx* member for Telnet ports (typically, 23,1023). Users are prompted for a user ID and password prior to the display of the *Enter command* or *Help* message or display of the USSTAB panel.

Terminal Security Settings

Under ACTEST the XSEC command accepts a new parameter called TERMID. The command XSEC TERMID ON|OFF used under ACTEST dynamically alters the passing of terminal IDs in the signon parameter list during signon attempts in an active Unicenter TCPaccess address space.

Terminal Security Activation

When an ACTEST user enters the following command, the Unicenter TCPaccess security interface places a terminal ID into the Terminal field of the signon parameter list for any user attempting a signon to Unicenter TCPaccess:

```
XSEC TERMID ON
```

The terminal ID passed during signon attempts is either the remote IP address of the originating host for the user or a VTAM APPL LU name. Then the XSEC command prints its global external security block and the following setting appears on the second line of the output:

```
TERMINAL SEC ACTIVE: YES
```

This is equivalent to specifying TERMID in the IJTCFG*xx* member.

Terminal Security Deactivation

When an ACTEST user enters the following command, the Unicenter TCPaccess security interface does not use the Terminal field in the signon parameter list during signon attempts:

```
XSEC TERMID OFF
```

The XSEC command prints its global external security block and the following setting displays on the second line of the output:

```
TERMINAL SEC ACTIVE: NO
```

This is equivalent to specifying NOTERMID in the IJTCFG*xx* member.

Types of eTrust CA-ACF2 Security

Unicenter TCPaccess uses these types of security with eTrust CA-ACF2:

- Signon security—All user ID and password combinations are validated by eTrust CA-ACF2
- Data set security—All FTP file transfers are validated by eTrust CA-ACF2
- Command security—Restricts service to SYSSTAT, ACTEST, and TCPEEP
- Source level security for FTP IP addresses, VTAM LUs, and Telnet services

The command security interface restricts access to application segment services. By default, the ACTEST, SYSTAT and TCPEEP are restricted under command security.

To maintain system security, only system programmers and operations personnel should have access to these services. Inexperienced users can cause serious damage to system performance and reliability through the ACTEST, SYSTAT, and TCPEEP services.

Because eTrust CA-ACF2 denies all access until permitted, additional steps are required to bring up Unicenter TCPaccess at a site where eTrust CA-ACF2 is installed.

Customizing eTrust CA-ACF2 Version 6 or Later

1. Create a logon ID (LID) record to associate with the startup JCL.

   Follow the installation procedures of your site to create an LID record; make sure these parameters are set in the Privileges Section—Group 2:

   ```
   MUSASS
   NO-INH
   BDT
   ```

   See the *eTrust CA-ACF2 Administrator's Guide* for instructions on creating LIDs.

   Place the LID in the USER field of the startup JCL job card.

   If your site runs eTrust CA-ACF2 6.0 or higher, it is not necessary to set NON-CNCL in the Unicenter TCPaccess LID record.

   Note: To prevent unauthorized users from attempting to use the production user ID for Unicenter TCPaccess, the TCP base product rejects all logon attempts to Unicenter TCPaccess from programs like FTP and ACTEST that use this ID.

2. Update GSO records for Unicenter TCPaccess.

   Note: Enter all commands exactly as shown. Do not change the SUBSYS=SNSTCP in the first SAFDEF record. SNSTCP in the SUBSYS parameter relates to parameters on the SAF security calls (not the LID chosen by the site).

```
ACF

SET CONTROL(GSO)

INSERT CLASMAP.AC#CMD RESOURCE(AC#CMD) RSRCTYPE(SAF) ENTITYLN(8)

CHANGE INFODIR TYPES(D-RSAF)

INSERT SAFDEF.ACSECPC1 ID(ACSECPC) MODE(GLOBAL)  REP
       PROGRAM(BYPASS#1) RACROUTE(SUBSYS=SNSTCP,REQSTOR=ACSECPC)

INSERT SAFDEF.ACSECPC2 ID(ACSECPC) MODE(GLOBAL)  REP
       PROGRAM(BYPASS#2) RACROUTE(SUBSYS=SNSTCP,REQSTOR=ACSECPC)

INSERT SAFDEF.ACSECPC3 ID(ACSECPC) MODE(GLOBAL)  REP
       PROGRAM(BYPASS#3) RACROUTE(SUBSYS=SNSTCP,REQSTOR=ACSECPC)

INSERT SAFDEF.ACSECPC4 ID(ACSECPC) MODE(GLOBAL)  REP
       PROGRAM(BYPASS#4) RACROUTE(SUBSYS=SNSTCP,REQSTOR=ACSECPC)

INSERT SAFDEF.ACCFTP2 ID(ACCFTP2) MODE(GLOBAL)   REP
       PROGRAM(ACCFPT2) RACROUTE(REQUEST=EXTRACT)

INSERT SAFDEF.FTP ID(FTP) MODE(GLOBAL)
       PROGRAM(FPT) RACROUTE(REQUEST=EXTRACT)

INSERT SAFDEF.FTP2 ID(FTP2) MODE(GLOBAL)
       PROGRAM(FPT2) RACROUTE(REQUEST=EXTRACT)

INSERT SAFDEF.FTP3 ID(FTP3) MODE(GLOBAL)
       PROGRAM(FPT3) RACROUTE(REQUEST=EXTRACT)
```

3. Use this command to build the INFODIR SAF records for Unicenter TCPaccess:

```
F ACF2,REBUILD(SAF),CLASS(R)
```

4. Update GSO records to allow password extraction for TCPEEP and FTP2.

   Certain Unicenter TCPaccess programs can extract encrypted passwords. The encrypted passwords can be used to sign a user on to the Unicenter TCPaccess address space. eTrust CA-ACF2 6.0 (and higher) systems can globally enable or disable password extraction.

   a. On systems running eTrust CA-ACF2 6.0, use these commands to see if password extraction is globally enabled or disabled:

```
ACF

SET CONTROL(GSO)
SHOW STATE
```

      If NOPSWDXTR is indicated, encrypted password gathering is globally disabled at the eTrust CA-ACF2 6.0 level.

b. Use these commands on systems running eTrust CA-ACF2 6.1 or higher, to see if password extraction is globally enabled or disabled:

```
ACF
SET CONTROL(GSO)
LIST PSWD
```

If NOPSWDXTR is indicated, encrypted password gathering was globally disabled.

c. To globally enable encrypted password gathering (on systems running eTrust CA-ACF2 6.0 or higher) issue these commands:

```
ACF
SET CONTROL(GSO)
CHANGE PSWD PSWDXTR
```

d. Use this operator command to activate the change to the GSO record:

```
F ACF2,REFRESH(PSWD)
```

e. Before an eTrust CA-ACF2 LID record that has the NOPSWD-XTR field set can use the changes described here for PSWDXTR, these steps must occur:

- The NOPSWD-XTR field in the LID record must be changed to PSWD-XTR.
- The user ID must be signed on to MVS and its password must be changed so the updated password can be stored in the eTrust Computer Associates-ACF2 database in a way that encrypted password signons can be used.

Note: You cannot use the ACF CHANGE command to turn on or off password extraction for individual LID records. The PSWD-XTR field cannot be set directly in the LID record (as it depends on the GSO option and the expiration of passwords). Read the PSWD section in the GSO records chapter of the *eTrust CA-ACF2* 6.X *MVS Administrator Guide* for information about how to change PSWD-XTR for the user community.

5. Set proper authority over mail data sets in the Unicenter TCPaccess LID record.

The LID associated with the Unicenter TCPaccess job must have allocation access authority to the HLQ(s) on the PATH parameter of the SMTP statement in member APPCFG*xx*. When you set the rules for the LID, set the ALLOC parameter to ALLOC(A).

The PATH parameter of the SMTP statement specifies the HLQ(s) for mail DASD data set names. SMTP requires that a data set naming convention be established for outgoing mail. The HLQ(s) for mail should be unique. If the HLQ for Unicenter TCPaccess data sets is CATCPIP, consider defining the HLQ for email as PATH(CATCPIP.EMAIL). If you assign PATH(CATCPIP) as the HLQ, the client mail handler tries to send all the Unicenter TCPaccess system data sets as mail data sets.

a. If the PATH parameter on the SMTP statement contains PATH(CATCPIP.EMAIL) and the Unicenter TCPaccess LID is CATCPIP, use the following eTrust CA-ACF2 commands to permit the CATCPIP and SYS1 LIDs alter authority:

```
ACF
SETRULE
COMPILE
$KEY(CATCPIP)
$OWNER('Production TCPACCES')
- UID(CATCPIP) READ (A) WRITE(A) ALLOC(A) EXEC(A)
- UID(SYS1-)  READ (A) WRITE(A) ALLOC(A) EXEC(A)
- UID(-) READ(A)
EMAIL.- UID(CATCPIP) READ(A) WRITE(A) ALLOC(A) EXEC(A)
EMAIL.- UID(SYS1-)  READ(A) WRITE(A) ALLOC(A) EXECA(A)
EMAIL.- UID(-) READ(A)
END
STORE
END
```

b. Adjust the user ID to the naming conventions of the installation site.

c. Avoid data set enqueue conflicts by choosing a unique PATH name for every Unicenter TCPaccess address space running at a site.

- PATH names of CATCPIP.EMAIL and CATCPIP.EMAIL2 are valid for separate Unicenter TCPaccess address spaces because the second level in the name is unique.
- The names CATCPIP.EMAIL and CATCPIP.EMAIL.A are not recommended for separate Unicenter TCPaccess address spaces because the second PATH name is a subset of the first.

6. Protecting packet trace programs from unauthorized use.

Packet tracing programs must be protected from unauthorized usage. Program T03PTCPE, and its alias TCPEEP, traces packets in and out of the network. Logon IDs, passwords, and perhaps proprietary installation data, can be seen with the packet trace programs.

Resource rules at the program level are the mechanism within eTrust CA-ACF2 to protect programs. Use the following commands to protect program T03PTCPE and its alias, TCPEEP, in library CATCPIP.LINKLIB where UIDs starting with SYS1 are granted access:

```
ACF

SET RULE
COMPILEA
$KEY(CATCPIP)
$OWNER('Production TCPacces')
LINKLIB UID(SYS1-) PGM(T03PTCPE) EXEC(A) READ(A) WRITE(A) ALLOC(A)
LINKLIB UID(SYS1-) PGM(TCPEEP) EXEC(A) READ(A) WRITE(A) ALLOC(A)
LINKLIB UID(-) EXEC(A) READ(A)
END
STORE
END
```

7. User validation is required for access to Unicenter TCPaccess internal debugging services ACTEST and SYSSTAT; validation is performed by checking resource name SYSTRAN in the SAF Resource Rule Entry.

   Users are prompted for a user ID and password when they invoke ACTEST or SYSSTAT. The user ID and password are validated by the eTrust CA-ACF2 security system. If the user ID and password are valid, the security system also checks to see if the user is authorized to access the resource name SYSTRAN in the SAF Resource Rule Entry. If the user ID is not authorized for a minimum of read access to the SYSTRAN resource name in the SAF Resource Rule Entry, then access to ACTEST or SYSSTAT is denied.

   In this example, user ID USER01 is the only user ID that has access to the Unicenter TCPaccess debugging services.

   Use the following commands to define the SAF Resource Rule Entry for resource name SYSTRAN (Replace user ID USER01 with the user ID of your local Unicenter TCPaccess systems programmer):

   ```
   ACF

   SET RESOURCE(SAF)
   COMPILE STORE
   $KEY(SYSTRAN) TYPE(SAF)
   UID(USER01) ALLOW SERVICE(READ)
   ```

   Note: If the resource name SYSTRAN in the SAF Resource Rule Entry does not exist user access is automatically denied by eTrust CA-ACF2 to the Unicenter TCPaccess internal debugging services ACTEST and SYSSTAT.

8. Activate Resource Rule Entry for Unicenter TCPaccess application services.

   You can use Unicenter TCPaccess command security to limit access to an application for any APPL statement in member APPCFG*xx*. Set the SECURITY parameter to something other than the default of SECURITY(NO).

   The Telnet commands ACTEST and SYSSTAT use the SYSTRAN resource name in the SAF Resource Rule Entry. To change the SYSTRAN resource name or to add security to other commands, use the SECURITY parameter of the APPL statement in member APPCFG*xx*. Refer to the *Customization Guide* for more information.

   Whenever an application defaults to, or sets, APPL SECURITY(NO) in member APPCFG*xx*, Unicenter TCPaccess allows universal access to the service.

   a. Define a NETSTAT application service in member APPCFGxx as follows:

      ```
      APPL NAME(NETSTAT) SECURITY(YES)
      ```

   b. Provide a valid user ID and password combination to Unicenter TCPaccess that has been authorized for access to the NETSTAT resource name in the SAF Resource Rule Entry to use the NETSTAT command.

The NETSTAT resource name in the SAF Resource Rule Entry name would be same as its service NAME (in this case NETSTAT) with SECURITY(YES) specified on an APPL statement. Unicenter TCPaccess checks the NETSTAT resource name in the SAF Resource Rule Entry for command security authorization before allowing a user ID access to the NETSTAT command. Issue this command to define the NETSTAT resource name in the SAF Resource Rule Entry:

```
ACF
SET RESOURCE(SAF)
COMPILE STORE
$KEY(NETSTAT) TYPE(SAF)
UID(USER01) ALLOW SERVICE(READ)
```

c. Define a NETSTAT application service in member APPCFG*xx* as follows:

```
APPL NAME(NETSTAT) SECURITY(SYSTEM)
```

d. Provide a valid user ID and password combination to Unicenter TCPaccess that has been authorized for access to the SYSTEM resource name in the SAF Resource Rule Entry to use the NETSTAT command.

Unicenter TCPaccess checks the SYSTEM resource name in the SAF Resource Rule Entry (as specified on the SECURITY parameter) for command security authorization before allowing a user ID access to the NETSTAT command.

e. Use this command to define the SYSTEM resource name in the SAF Resource Rule Entry:

```
ACF

SET RESOURCE(SAF)
COMPILE STORE
$KEY(SYSTEM) TYPE(SAF)
UID(USER01) ALLOW SERVICE(READ)
```

9. eTrust CA-ACF2: Using the Source Security within Unicenter TCPaccess.

Unicenter TCPaccess has the ability to pass a source terminal ID to eTrust CA-ACF2 during signon attempts. Unicenter TCPaccess passes either the remote IP address or the actual VTAM terminal ID in the Terminal field during signon attempts.

Source security customization is an optional feature. Any site that does not currently implement source security can skip this step.

For more detailed information about source security for terminals, see the *eTrust CA-ACF2 MVS Administrator Guide*.

To use the source security within Unicenter TCPaccess, follow these steps:

Step 1 SAMP member A03ACCES shows the VTAM APPL names starting with A03VLT. This member is a model to use or modify for local use.

The eTrust CA-ACF2 security administrator should group all the VTAM APPL names associated with Unicenter TCPaccess into an X-SGP source record. Currently, there is no mechanism within Unicenter TCPaccess to map VTAM LU usage to specific logon IDs at the VTAM logon points. You do not know which LU will be allocated at these logon points. The LUs used at VTAM logon points within Unicenter TCPaccess are allocated by ACCPOOL. Do not confuse the LUPOOL capability to map IP addresses to logon IDs with ACCPOOL LU customization. All logon ID records that need access to Unicenter TCPaccess through VTAM can then have the new source group added to their source GROUP records.

The eTrust CA-ACF2 security administrator can create X-SGP source records for the A03ACCES SAMP member by issuing these commands for VTAM usage:

```
SET ACF
SET X(SGP)
INSERT A03VLT SOURCE INCLUDE(A03VLT-) ADD
```

The eTrust CA-ACF2 security administrator should check with both the Unicenter TCPaccess and VTAM systems programmers to identify which VTAM LUs are being used by the site for access within Unicenter TCPaccess.

Step 2 All logon IDs that want to sign on to Unicenter TCPaccess must be permitted source authority to the Unicenter TCPaccess IP addresses as specified on the IP address parameter for every NETWORK statement in TCPCFG*xx* member.

A sample NETWORK statement in member TCPCFG*xx* may begin like this:

```
NETWORK IPADDRESS(138.42.224.15)
```

The security system accepts source IDs only in hexadecimal form, so the above IP address must be converted. IP address 138.42.224.15 would use a terminal ID of 8A2AE00F (where 138 = 8A, 42 = 2A, 224 = E0, and 15 = 0F).

Use these commands to create an X-SGP source record at a site to sign on to Unicenter TCPaccess using its default IP address of 138.42.224.15 for source 8A2AE00F:

```
SET ACF
SET X(SGP)
INSERT 8A2AE00F SOURCE INCLUDE(8A2AE00F) ADD
```

All logon ID records that need access to Unicenter TCPaccess must then have the new source entry 8A2AE00F added to their source GROUP records.

Step 3 Any individual logon ID that uses authorized Telnet commands or FTP from a remote site needs READ access authority for the terminal IP address of the remote site. The originating remote IP address is used for all signon attempts.

To create an X-SGP source record at a site to sign on to Unicenter TCPaccess using its host IP address 138.42.224.250 for source 8A2AE0FA, issue the following commands:

```
SET ACF
SET X(SGP)
INSERT 8A2AE0FA SOURCE INCLUDE(8A2AE0FA) ADD
```

This X-SGP source record can now be placed in the source group record for any logon IDs coming in from host 138.42.224.250.

Step 4 An eTrust CA-ACF2 administrator can create a generic X-SGP source record for 8A2AE0- for the local network of 138.42.220 using the following commands:

```
SET ACF
SET X(SGP)
INSERT 8A2AE0 SOURCE INCLUDE(8A2AE0**) ADD
```

You can now place this X-SGP source record in the source group record for any logon IDs coming in from the local network.

Step 5 Activate the X-SGP records using these eTrust CA-ACF2 operator console command:

```
F ACF2,NEWXREF,TYPE(SGP)
```

Step 6 Configure Unicenter TCPaccess to place the terminal ID on all security parameter lists passed to eTrust CA-ACF2 for all signon attempts to Unicenter TCPaccess. If you place TERMID into the XSEC parameter list on the SECURITY statement in the IJTCFGxx member, this happens automatically. By default, terminal IDs are not passed on any signon call.

To activate passing source terminal IDs on the security parameter list to eTrust CA-ACF2 for an active Unicenter TCPaccess address space, issue the following command under ACTEST:

```
XSEC TERMID ON
```

To deactivate passing source terminal IDs on the security parameter list to eTrust CA-ACF2 for an active Unicenter TCPaccess address space, use this command under ACTEST:

```
XSEC TERMID OFF
```

Step 7 To enable signon checking for Telnet users, add the CPASSWORD option to the Telnet related SERVICE statement(s) in APPCFG*xx* for Telnet ports (typically, 23,1023).

CAUTION! *Activate source security checking only after all eTrust CA-ACF2 customization for Unicenter TCPaccess is completed. eTrust Computer Associates-ACF2 source security can prevent anyone from signing on to MVS, as well as Unicenter TCPaccess, if the customization is performed incorrectly.*

Jobs submitted by TERMID checked logon IDs would fail security unless explicit user IDs and passwords are given when NO-INH is associated with the logon ID of the submitter.

Step 10 To activate the changes in numbers 2 through 8, perform an IPL or issue a GSO console operator refresh. Use this command for the refresh:

```
F ACF2,REFRESH(ALL)
```

eTrust CA-Top Secret Options

This section describes the types of security options available to sites running eTrust CA-Top Secret.

Types of eTrust CA-Top Secret Security

Unicenter TCPaccess uses the following types of security with eTrust CA-Top Secret:

- Signon Security

  All user ID/password combinations are validated by eTrust CA-Top Secret

- Data set Security

  All file transfers under FTP are validated by eTrust CA-Top Secret

- Resource eTrust CA-Top Secret Security

  Restricts service in the Server Telnet control table

- Source level security for FTP IP addresses, VTAM LUs, and Telnet services

The Unicenter TCPaccess command security interface restricts access to services in the Server Telnet control table. ACTEST, SYSSTAT and TCPEEP should be protected with resource security

To maintain system security, restrict access to system programmers and operations personnel. Inexperienced users can cause serious damage to system performance and reliability through the ACTEST, SYSSTAT and TCPEEP services.

eTrust CA-Top Secret Customization

The Unicenter TCPaccess address space functions as a true FACILITY to eTrust CA-Top Secret. Use this setup to enable Unicenter TCPaccess with eTrust CA-Top Secret:

1. Set up a Unicenter TCPaccess FACILITY entry with eTrust CA-Top Secret options as shown in this example:

```
FAC(USERx=NAME=CATCPIP)

FAC(CATCPIP=PGM=BYP)
FAC(CATCPIP=ACTIVE,NOABEND,NOASUBM,NOAUDIT,AUTHINIT,ID=c)
FAC(CATCPIP=NOINSTDATA,KEY=8,LCFCMD,LOCKTIME=0,NOLUMSG,LOG(NONE))
FAC(CATCPIP=NOMRO,MULTIUSER,NOPSEUDO,NORNDPW,RES,SIGN(M))
FAC(CATCPIP=SHRPRF,NOSTMSG,TENV,NOTSOC,WARNPW,NOXDEF)
```

   In the above example, the Unicenter TCPaccess FACILITY is named CATCPIP. You can use any name up to eight bytes in length. If another name is used, it must be substituted in the setup examples.

   - USER*x* can be any user-defined resource type available at the installation and the *x* value can be any keyboard character.
   - For ID=*c*, *c* is a single alphanumeric that represents the FACILITY for reporting purposes (see FACILITY under eTrust CA-Top Secret control options).
   - RNDPW (RaNDomPassWords, or return expired new random passwords) can be set on the TCP base product FACILITY. However, only FTP returns all the messages from eTrust CA-Top Secret when the password expires. When RNDPW is placed on your FACILITY definition in an eTrust CA-Top Secret environment, eTrust CA-Top Secret returns a new randomly generated password when an expired password associated with an ACcessor ID (ACID) is correctly presented during signon.

   Note: Do not place operands NOPSEUDO, NOMRO, and TENV on the FACILITY definition under eTrust CA-Top Secret 4.3 or above. These operands are no longer supported.

2. Give ACIDs access to the Unicenter TCPaccess FACILITY.

   To permit ACID USER01 access to the Unicenter TCPaccess FACILITY (CATCPIP), the security administrator must issue this command:

```
TSS ADD(USER01) FAC(CATCPIPA)
```

3. Create the Unicenter TCPaccess ACID.

   Build the ACID for the Unicenter TCPaccess address space with the TSS CREATE command.

   The following command creates ACID CATCPIPA to run as a started task:

   ```
   TSS CREATE(CATCPIPA) NAME('TCPACCES ACID') FAC(STC) TYPE(USER)
   PASS(NOPW) DEPT(dept_name) MASTFAC(CATCPIP)
   ```

   Note: To prevent unauthorized users from attempting to use the production ACID for Unicenter TCPaccess, the TCP base product rejects all logon attempts to Unicenter TCPaccess from programs like FTP and ACTEST that use this ACID.

4. The Unicenter TCPaccess ACID must have authority to access the data sets it needs to function in the customer's environment.

   If you unloaded all the Unicenter TCPaccess data sets with the HLQ CATCPIP and the Unicenter TCPaccess ACID is CATCPIPA, then the security administrator can grant access to the Unicenter TCPaccess ACID CATCPIPA by issuing this command:

   ```
   TSS PERMIT(TCPACCSA) DSN(CATCPIP) ACCESS(UPDATE)
   ```

5. Protecting packet trace programs from unauthorized use.

   Packet tracing programs must be protected from unauthorized usage. Program T03PTCPE, and its alias TCPEEP, traces packets in and out of the network. ACIDs, passwords, and perhaps proprietary installation data, can be seen with the packet trace programs.

   Resource rules at the program level are the mechanism within eTrust CA-Top Secret that protects programs. The following commands can be used to protect program T03PTCPE and its alias, TCPEEP, where department SYKSDEPT owns the programs and ACID SYSUSER has access:

   ```
   TSS ADD(SYSDEPT) PROG(T03PTCPE)
   TSS ADD(SYSDEPT) PROG(TCPEEP)
   TSS PER(SYSUSER) PROG(T03PTCPE)
   TSS PER(SYSUSER) PROG(TCPEEP)
   ```

6. Set up Unicenter TCPaccess as a started task.

   If Unicenter TCPaccess runs as a started task, the relationship also must be established in the eTrust CA-Top Secret STC record. The following TSS ADDTO command connects the started task with the ACID defined by the TSS CREATE command.

   Note: This example assumes that the Unicenter TCPaccess PROC name is SWPROC and the ACID defined for use by the Unicenter TCPaccess Task is CATCPIPA:

   ```
   TSS ADDTO(STC) PROC(SWPROC) ACID(CATCPIPA)
   ```

7. Set up Unicenter TCPaccess as a batch job.

   If Unicenter TCPaccess is run as a batch job, the relationship is established by the USER= value coded on the job card. In this example, the Unicenter TCPaccess job must be coded with USER=CATCPIPA.

8. If the SMTP email services are being used, that is, the PATH parameter of the SMTP statement in member APPCFG*xx* is specified, then the ACID associated with the Unicenter TCPaccess job or started task must have CREATE and SCRATCH access to the HLQ specified on the PATH parameter.

   If the PATH parameter is specified as PATH(CATCPIPS.EMAIL) and the ACID associated with the Unicenter TCPaccess job or started task is CATCPIPA, then the security administrator can allow access by issuing this command:

   ```
   TSS PER(CATCPIPA) DSN(CATCPIP.EMAIL.%) ACCESS(CREATE)
   ```

   To avoid data set enqueue conflicts, choose a unique PATH name for every Unicenter TCPaccess address space running at a site.

   - PATH names of CATCPIP.EMAIL and CATCPIP.EMAIL2 are valid for separate Unicenter TCPaccess address spaces because the second level in the name is unique.
   - The names CATCPIP.EMAIL and CATCPIP.EMAIL.A are not recommended for separate Unicenter TCPaccess address spaces because the second PATH name is a subset of the first.

   If you assign PATH(CATCPIP) as the HLQ, the client mail handler tries to send all the Unicenter TCPaccess system data sets as mail data sets.

9. User validation is required for access to Unicenter TCPaccess internal debugging services ACTEST and SYSSTAT. Validation is performed by checking to see if an ACID has access to entry SYSTRAN in eTrust CA-Top Secret User Resource Class UR1.

   Users are prompted for an ACID and password when they invoke ACTEST or SYSSTAT. The ACID and password are validated by the security system and if valid, the security system validates the user ID for authority to access the SYSTRAN entry in eTrust CA-Top Secret's User Resource Class UR1. If the user is not permitted a minimum of read access to the SYSTRAN entry in the User Resource Class UR1, access to ACTEST and SYSSTAT is denied.

   Use the following eTrust CA-Top Secret command to find the resource entry names being used:

   ```
   TSS WHOOWNS UR1(*)
   ```

   Use the following command to define Unicenter TCPaccess entry SYSTRAN in User Resource Class UR1 owned by user USER01:

   ```
   TSS ADDTO(USER01) UR1(SYSTRAN)
   ```

The security administrator can now permit user USER02 to use the SYSTRAN entry in class UR1 with this command:

```
TSS PERMIT(USER02) UR1(SYSTRAN) ACCESS(READ)
```

This permits user USER02 access to the ACTEST and SYSSTAT debugging services.

Note: If the Unicenter TCPaccess entry SYSTRAN in User Resource Class UR1 does not exist, user access is denied by eTrust CA-Top Secret to the Unicenter TCPaccess internal debugging services ACTEST and SYSSTAT.

10. You can use command security to limit access to any application with an APPL statement in member APPCFG*xx*. Set the SECURITY parameter to something other than the default of SECURITY(NO).

    The Telnet commands ACTEST and SYSSTAT use the entry SYSTRAN in User Resource Class UR1. To change the SYSTRAN entry or to add security to other commands, use the SECURITY parameter of the APPL statement in member APPCFG*xx*. See the *Customization Guide* for more information.

    Whenever an application defaults to, or sets, APPL SECURITY(NO) in member APPCFG*xx*, Unicenter TCPaccess allows universal access to that particular application.

    a. Use the following command to define a NETSTAT application service in member APPCFG*xx* as:

       ```
       APPL NAME(NETSTAT) SECURITY(YES)
       ```

    b. Provide a valid ACID password combination to Unicenter TCPaccess that has access to the NETSTAT entry in the User Resource Class UR1 to use the NETSTAT command.

       The NETSTAT entry in the User Resource Class UR1 should be same as its service NAME (in this case NETSTAT) with SECURITY(YES) specified on an APPL statement. Unicenter TCPaccess checks the NETSTAT entry in the User Resource Class UR1 for command security authorization before allowing an ACID access to the NETSTAT command.

       Use the following command to define the NETSTAT entry in the User Resource Class UR1:

       ```
       TSS ADDTO(USER01) UR1(NETSTAT)
       TSS PERMIT(USER02) UR1(SYSTRAN) ACCESS(READ)
       ```

       In the previous example the NETSTAT entry in User Resource Class UR1 is owned by ACID USER01. ACID USER02 has authority to issue the NETSTAT command.

    d. Define a NETSTAT application service in member APPCFG*xx* as follows:

       ```
       APPL NAME(NETSTAT) SECURITY(SYSTEM)
       ```

e. Provide a valid ACID password combination to Unicenter TCPaccess that has access to the SYSTEM entry in the SAF Resource Rule Entry to use the NETSTAT command.

   Unicenter TCPaccess checks the SYSTEM entry in User Resource Class UR1 (as specified on the SECURITY parameter above) for command security authorization before allowing an ACID access to the NETSTAT command.

f. Use the following command to define the SYSTEM entry in the User Resource Class UR1:

```
TSS ADDTO(USER01) UR1(SYSTEM)
TSS PERMIT(USER02) UR1(SYSTEM) ACCESS(READ)
```

   In the above example the SYSTEM entry in User Resource Class UR1 is owned by ACID USER01. ACID USER02 has authority to issue the NETSTAT command.

11. eTrust CA-Top Secret: Using the Terminal Security Class within Unicenter TCPaccess

Note: Terminal security customization is an optional feature. Any site that currently does not implement terminal security can skip this step.

Unicenter TCPaccess can pass a terminal ID to eTrust CA-Top Secret during signon attempts. Unicenter TCPaccess passes either the remote IP address or the actual VTAM terminal ID in the Terminal field during signon attempts.

To use the terminal security class within Unicenter TCPaccess, follow these steps:

a. Refer to the *eTrust CA-Top Secret Implementation: General Guide* for information about terminal security. Be careful when activating terminal security for the first time. If done incorrectly, no one will be able to sign on to either MVS or Unicenter TCPaccess.

   Research these sample commands for turning on terminal security:

```
TSS LIST(RDT) RESCLASS(TERMINAL)
TSS REPLACE(RDT) ATTR(GENERIC,NODEFPROT) DEFACC(READ)
```

   Issuing the following eTrust CA-Top Secret command prevents all undefined terminals from signing on to an address space using terminal security access to your site. This can be very useful in restricting access to Unicenter TCPaccess via IP addresses. Undefined IP addresses are not permitted to sign on to Unicenter TCPaccess.

```
TSS REPLACE(RDT) ATTR(GENERIC,DEFPROT) DEFACC(NONE)
```

b. All ACIDs must be permitted READ access authority to the VTAM Terminal APPL names (not ACBNAMEs) that need to access Unicenter TCPaccess through VTAM logon points. There is no current mechanism within Unicenter TCPaccess to map VTAM LU usage to specific ACIDs at the VTAM logon points. You do not know which LU will be allocated at these logon points. The LUs used at VTAM logon points within Unicenter TCPaccess are allocated by ACCPOOL. Do not confuse the LUPOOL capability to map IP addresses to an ACID with ACCPOOL LU customization.

   SAMP member A03ACCES shows VTAM APPL names starting with A03VLT. This member is a model to use or modify for local use.

   The eTrust CA-Top Secret security administrator can use the following commands to define the terminals and designate which ACID can access Unicenter TCPaccess via VTAM utilizing the A03VLT*xx* VTAM APPLs:

   ```
   TSS ADD(acid) TERM(A03VLT)
   TSS PER(acid) TERM(A03VLT) ACCESS(READ)
   ```

   If the eTrust CA-Top Secret system administrator gives access to Unicenter TCPaccess via the VTAM interfaces only to departments SYS1 and ENG, the terminals can be protected as defined in A03ACCES with the following eTrust CA-Top Secret commands:

   ```
   TSS ADD(SYS1) TERM(A03VLT)
   TSS PER(ENG) TERM(A03VLT) ACCESS(READ)
   ```

   The eTrust CA-Top Secret security administrator should check with both the Unicenter TCPaccess and VTAM system's programmers to identify which VTAM LUs are being used by the site for access within Unicenter TCPaccess.

c. All ACIDs wanting to sign on to Unicenter TCPaccess must be permitted READ access authority to the Unicenter TCPaccess IP addresses as specified on the IP address parameter for every NETWORK statement in the TCPCFG*xx* member.

   A sample NETWORK statement in member TCPCFG*xx* may begin like this:

   ```
   NETWORK IPADDRESS(138.42.224.15)
   ```

   The security system accepts terminal IDs only in hexadecimal form, so the above IP address must be converted. IP address 138.42.224.15 uses a terminal ID of 8A2AE00F (where 138 = 8A, 42 = 2A, 224 = E0, and 15 = 0F).

   To allow all ACIDs at a site to sign on to Unicenter TCPaccess for a default IP address of 138.42.224.15 with terminal 8A2AE00F, issue these commands:

   ```
   TSS ADD(acid) TERM(8A2AE00F)
   TSS PER(ALL) TERM(8A2AE00F) ACCESS(READ)
   ```

If you replace ACCESS(READ) on the above command with ACCESS(NONE), the eTrust CA-Top Secret security administrator must use the eTrust CA-Top Secret PERMIT command to allow READ access to all ACIDs or departments that need to sign on to Unicenter TCPaccess.

d. Any individual ACID using authorized Telnet commands or FTP from a remote site into Unicenter TCPaccess must have READ access authority for the terminal that represents the IP address of the remote site. The originating remote IP address is used for all signon attempts to Unicenter TCPaccess once a connection to Unicenter TCPaccess is made.

   If local ACID USER01 comes off the local network from host 138.42.224.250, then this ACID needs to be permitted access to this IP address via terminal 8A2AE0FA. This can be done with the following commands:

   ```
   TSS ADD(acid) TERM(8A2AE0FA)
   TSS PER(USER01) TERM(8A2AE0FA) ACCESS(READ)
   ```

   An eTrust CA-Top Secret administrator can allow everyone on his local network (138.42.224) access to Unicenter TCPaccess with the following eTrust CA-Top Secret commands:

   ```
   TSS ADD(acid) TERM(8A2AE0)
   TSS PER(ALL) TERM(8A2AE0) ACCESS(READ)
   ```

e. Configure Unicenter TCPaccess to place the terminal ID on all security parameter lists passed to eTrust CA-Top Secret for all signon attempts to Unicenter TCPaccess. If you place TERMID into the XSEC parameter list on the SECURITY statement in the IJTCFG*xx* member, this happens automatically. By default, Unicenter TCPaccess does not place the terminal ID on any signon call.

   To activate passing terminal IDs on the security parameter list to eTrust CA-Top Secret for an active Unicenter TCPaccess address space, use the following command under ACTEST:

   ```
   XSEC TERMID ON
   ```

   You can deactivate passing terminal IDs on the security parameter list to eTrust CA-Top Secret for an active Unicenter TCPaccess address space by issuing the following command under ACTEST:

   ```
   XSEC TERMID OFF
   ```

f. To enable signon checking for Telnet users, add the CPASSWORD option to the SERVICE statement in the APPCFG*xx* member for Telnet ports (typically, 23,1023).

RACF Options

If a computer site runs RACF, the Unicenter TCPaccess RACF interface automatically becomes active upon installation. This section describes the types of security options available to sites running RACF and how to customize security for Unicenter TCPaccess.

Types of RACF Security

With RACF, the following types of security are active in Unicenter TCPaccess:

- Signon security—All user ID/password combinations are validated by RACF
- Data set security—All file transfers under FTP are validated by RACF
- Source level security for FTP IP addresses, VTAM LUs, and Telnet services

In addition to automatic data set and signon security, Unicenter TCPaccess provides command security. See Customizing Command Security with RACF for instructions on customizing security.

The Unicenter TCPaccess security interface for commands restricts access to application segment services defined in member APPCFG*xx*. Unicenter TCPaccess command security protects the following:

- ACTEST
- SYSSTAT

To maintain a high level of system security, only system programmers and operations personnel should have access to these services. Inexperienced users can cause serious damage to system performance and reliability through the ACTEST, SYSSTAT and TCPEEP services.

Customizing Command Security with RACF

Unicenter TCPaccess uses the local installation-defined resource classes of RACF to implement command security. Refer to IBM document *SPL: RACF SC28-1343* for additional information regarding the macros and tables described in this section.

1. Modify the installation local class descriptor table.

   Place member ICHERCDE in the SAMP data set in your installation source of ICHRRCDE (local class descriptor table).

   Follow your site's installation procedures to update the local class descriptor table ICHERCDE.

   Member ICHRRCDE in the SAMP data set is an example of the general resource class AC#CMD of Unicenter TCPaccess. The Unicenter TCPaccess general resource class description must be used as shown except for the ID, OPER, and POSIT parameters:

```
AC#CMD ICHERCDE CLASS=AC#CMD,     class name; do not change.
    ID=128,                       Unique ID between 128-255.
    MAXLNTH=8,                    Up to 8 character profile name.
    FIRST=ALPHA,                  First character is alphabetic.
    OPER=YES,                     Allow operations people free reign.
    OTHER=ALPHANUM,
    POSIT=25,                     Must be in range 25-55.
    DFTUACC=NONE                  Must be NONE.
```

   Note: The first four characters of each resource class name must be different from the first four characters of all other class names. One of the four characters should be a national or numeric character to avoid inadvertently choosing a future IBM class name.

2. Modify the local installation-defined router table.

   Member ICHRFRTB in the SAMP data set is an example of the Unicenter TCPaccess router entry. Add the following line from member ICHRFRTB in the SAMP data set into your installation source of ICHRFR01 (local router table) exactly as shown. Do not modify any of its parameters.

```
AC#CMD  ICHRFRTB CLASS=AC#CMD,ACTION=RACF
```

   Follow the installation procedures of your site to update the local router table ICHRFR01.

3. Perform an IPL on the system with a CLPA to activate the local installation router and class descriptor tables.

4. Activate the AC#CMD resource class with the following command:

```
SETROPTS CLASSACT(AC#CMD)
```

5. Follow the installation procedures of your site to create a user ID associated with the Unicenter TCPaccess job or started task.

   This sample ADDUSER command creates user ID CATCPIP associated with Unicenter TCPaccess in group PROD:

```
ADDUSER CATCPIP OWNER(PROD) DFLTGRP(PROD)
                NAME('TCPACCES ACCESS') DATA('production job')
```

6. If your site is running Unicenter TCPaccess as a started task, update member ICHRIN03.

   Note: To prevent unauthorized users from attempting to use the production user ID for Unicenter TCPaccess, the TCP base product rejects all logon attempts to Unicenter TCPaccess from programs like FTP and ACTEST that use this ID.

7. Give the Unicenter TCPaccess user ID proper authority over mail data sets.

   The user ID associated with the Unicenter TCPaccess job must have an access level of ALTER for the high-level qualifier (HLQ) on the PATH parameter on the SMTP statement in member APPCFG*xx*.

   The PATH parameter of the SMTP statement specifies the HLQ for mail DASD data set names. SMTP requires that a data set naming convention be established for outgoing mail data sets.

   The HLQ for mail must be unique. For example, if the HLQ for Unicenter TCPaccess data sets is CATCPIP, define the email HLQ as two levels, such as in PATH(CATCPIP.EMAIL). If the HLQ for email is not unique, the client mail handler will attempt to send all the Unicenter TCPaccess system data sets as mail data sets.

   If the PATH parameter on the SMTP statement contains PATH(CATCPIP.EMAIL) and Unicenter TCPaccess user is CATCPIP, the security administrator could use these RACF commands to permit alter authority for the TCPACCES user ID:

```
ADDSD 'CATCPIP.EMAIL.*' UACC(NONE) OWNER(CATCPIP)
        DATA('TCPACCES MAIL DATA SET ')
PERMIT 'CATCPIP.EMAIL.*' ID(CATCPIP) ACCESS(ALTER)
```

   To avoid data set enqueue conflicts, choose a unique PATH name for every Unicenter TCPaccess address space running at a site.

   - PATH names of CATCPIP.EMAIL and CATCPIP.EMAIL2 would be valid for separate Unicenter TCPaccess address spaces because the second level in the name is unique.
   - The names CATCPIP.EMAIL and CATCPIPS.EMAIL.A are not recommended because the second PATH name is a subset of the first.

8. Protecting packet trace programs from unauthorized use.

   Packet tracing programs must be protected from unauthorized usage. Program T03PTCPE and its alias, TCPEEP, traces packets in and out of the network. User IDs, passwords, and perhaps proprietary installation data, can be seen with the packet trace programs.

   Program control is a mechanism within RACF to protect programs. The following command turns on program control within RACF:

   ```
   SETROPTS WHEN(PROGRAM)
   ```

   The LINKLIB data set should contain programs T03PTCPE and its alias, TCPEEP. To protect these programs from unauthorized usage in library CATCPIP.LINK on VOLSER SYS001 where user SYSUSER can execute these unauthorized programs, issue the following commands:

   ```
   ADDSD 'TCPACCES.LINK' UACC(EXECUTE)
   RDEFINE PROGRAM T03PTCPE
   ADDMEM('CATCPIP.LINK'/SYS001K/PADCHKK) UACC(NONE)
   RDEFINE PROGRAM TCPEEP
   ADDMEM('CATCPIP.LINK'/SYS001/PADCHK) UACC(NONE)
   PERMIT T03PTCPE ID(SYSUSER) ACCESS(EXECUTE)
   PERMIT TCPEEP ID(SYSUER) ACCESS(EXECUTE)
   SETROPTS WHEN(PROGRAM) REFRESH
   ```

9. User validation is required for access to the Unicenter TCPaccess internal debugging services ACTEST and SYSSTAT.

   Users are prompted for a user ID and password when they invoke ACTEST and SYSSTAT. The user ID and password are validated by the security system. If the user ID and password are valid, the security system also checks to see if the user is authorized to access the SYSTRAN resource profile in the AC#CMD resource class. If the user is not authorized for a minimum of read access to the SYSTRAN resource profile, then access to ACTEST or SYSSTAT is denied.

   a. Use the following command to define the Unicenter TCPaccess profile SYSTRAN in resource class AC#CMD:

      ```
      RDEFINE AC#CMD (SYSTRAN) UACC(NONE)
      ```

   b. Use the following command to permit group SYS1 and user USER01 to use the SYSTRAN profile in class AC#CMD:

      ```
      PERMIT SYSTRAN CLASS(AC#CMD) ID(SYS1,USER01) ACCESS(READ)
      ```

      This gives group SYS1 and user USER01 access to the ACTEST and SYSTAT debugging services.

Note: If the RACF profile SYSTRAN does not exist or if the AC#CMD resource class is not defined and activated, user access is automatically allowed to the service.

10. Activate profiles for Unicenter TCPaccess application services.

    You can use command security to limit access to any application with an APPL statement in member APPCFG*xx*. Set the SECURITY parameter to something other than the default of SECURITY(NO) and use the SYSTRAN profile.

    To change the profile or to add security to other commands, use the SECURITY parameter of the APPL statement in member APPCFG*xx*. See the *Customization Guide* for more information.

    Whenever an application defaults to or sets APPL SECURITY(NO) in member APPCFG*xx*, Unicenter TCPaccess allows universal access to the service.

    a. Define a NETSTAT application service in member APPCFG*xx* as follows:

       ```
       APPL NAME(NETSTAT) SECURITY(YES)
       ```

       Provide a valid user ID and password combination to Unicenter TCPaccess that is authorized for access to the NETSTAT profile. SECURITY(YES) tells Unicenter TCPaccess to make command security calls using the service name for the profile name. In this example the NETSTAT profile name is NETSTAT.

    b. Use the following command to define the NETSTAT profile in class AC#CMD:

       ```
       RDEFINE AC#CMD (NETSTAT) UACC(NONE)
       ```

    c. Use the following command to give group SYS1 and user ID USER01 permission to use the NETSTAT profile in class AC#CMD:

       ```
       PERMIT NETSTAT CLASS(AC#CMD) ID(SYS1,USER01) ACCESS(READ)
       ```

    d. Define a NETSTAT application service in member APPCFG*xx* as follows:

       ```
       APPL NAME(NETSTAT) SECURITY(SYSTEM)
       ```

    e. Provide a valid user ID and password combination to Unicenter TCPaccess that has been authorized for access to the SYSTEM profile to use the NETSTAT command.

       In this example the NETSTAT profile name is SYSTEM, as specified on the SECURITY parameter.,

    f. Use the following command to define the SYSTEM profile in class AC#CMD:

       ```
       RDEFINE AC#CMD (SYSTEM) UACC(NONE)
       ```

g. Use the command to permit group SYS1 and user ID USER01 to use the SYSTEM profile in class AC#CMD:

```
PERMIT SYSTEM CLASS(AC#CMD) ID(SYS1,USER01) ACCESS(READ)
```

Note: Any new user ID defined to RACF while the Unicenter TCPaccess address space is actively running, is not allowed access to resources protected by command security (ACTEST SYSTAT) or any protected application services. Once the Unicenter TCPaccess address space is brought down and restarted, the new user ID is allowed access to these services.

RACF: Using the Terminal Security Class Within Unicenter TCPaccess

Unicenter TCPaccess has the ability to pass a terminal ID to RACF during signon attempts. Unicenter TCPaccess passes either the remote IP address or the actual VTAM terminal ID in the Terminal field during signon attempts.

Terminal security customization is an optional feature. Any site that currently does not implement terminal security may skip this step.

To use the terminal security class within Unicenter TCPaccess, follow these steps:

1. Read the *RACF Security Administrator's Guide* (SC23-3726) for information on terminal security.

   Be very careful when activating terminal security for the first time. If done incorrectly, no one will be able to sign on to either MVS or Unicenter TCPaccess.

   Research these sample commands for turning on terminal security:

   ```
   SETROPTS TERMINAL(READ)
   ```

   ```
   SETROPTS CLASSACT(TERMINAL) RACLIST(TERMINAL)
   ```

   Issuing the RACF command SETROPTS TERMINAL(NONE) prevents all undefined terminals from signing on to an address space using terminal security. This is useful in restricting access to Unicenter TCPaccess via IP addresses. Undefined IP addresses are not permitted to sign on to Unicenter TCPaccess.

2. All users that need to access Unicenter TCPaccess through VTAM logon points must be permitted READ access authority to the VTAM Terminal APPL names (not ACBNAMEs). Currently, there is no mechanism within Unicenter TCPaccess to map VTAM LU usage to specific user IDs at the VTAM logon points. You do not know which LU will be allocated at these logon points. The LUs used at VTAM logon points within Unicenter TCPaccess are allocated by ACCPOOL. Do not confuse the LUPOOL capability to map IP address to user IDs with ACCPOOL LU customization.

SAMP member A03ACCES shows VTAM APPL names starting with A03VLT. This member is a model to use or modify for local use.

The RACF security administrator can use the RDEFINE TERMINAL... and the PERMIT RACF commands to designate which users can access Unicenter TCPaccess via VTAM using the A03VLT*xx* VTAM APPLs.

If the RACF system administrator decides to allow access to Unicenter TCPaccess via the VTAM interfaces only to groups SYS1 and ENG, the terminals can be protected as defined in the SAMP member A03ACCES with the following RACF commands:

```
RDEFINE TERMINAL A03VLT* UACC(NONE)

PERMIT A03VLT* CLASS(TERMINAL) ID(SYS1,ENG) ACCESS(READ)
```

The RACF security administrator should check with both the Unicenter TCPaccess and VTAM systems programmers to identify which VTAM LUs are being used by the site for access within Unicenter TCPaccess.

3. All users who want to sign on to Unicenter TCPaccess must have READ access authority to the Unicenter TCPaccess IP address(es) as specified on the IP address parameter for every NETWORK statement in the TCPCFG*xx* member.

   A sample NETWORK statement in member TCPCFG*xx* may begin like this:

   ```
   NETWORK IPADDRESS(138.42.224.15)
   ```

   The security system accepts terminal IDs only in hexadecimal form, so the above IP address must be converted. IP address 138.42.224.15 would use a terminal ID of 8A2AE00F (where 138 = 8A, 42 = 2A, 224 = E0, and 15 = 0F).

   To allow all users at a site to sign on to Unicenter TCPaccess for a default IP address of 138.42.224.15 with terminal 8A2AE00F, issue the following command:

   ```
   RDEFINE TERMINAL 8A2AE00F UACC(READ)
   ```

   If you replace UACC(READ) on the above command with UACC(NONE), the RACF security administrator must use the RACF PERMIT command to allow READ access to all users or groups that need to sign on to Unicenter TCPaccess.

4. Individual users must be permitted to use their own IP addresses. If local user USER01 comes off the local network from host 138.42.224.250, this user must be permitted access to this IP address via terminal 8A2AE0FA. This can be done with the following commands:

   ```
   RDEFINE TERMINAL 8A2AE0FA UACC(NONE)

   PERMIT 8A2AE0FA CLASS(TERMINAL) ID(USER01) ACCESS(READ)
   ```

   A RACF administrator could allow everyone on his local network (138.42.224) access to Unicenter TCPaccess with the following RACF command:

   ```
   RDEFINE TERMINAL 8A2AE0* UACC(READ)
   ```

5. Configure Unicenter TCPaccess to place the terminal ID on all security parameter lists passed to RACF for all signon attempts to Unicenter TCPaccess. If you use the TERMID option on the XSEC parameter of the SECURITY statement in the IJTCFG*xx* member, this happens automatically. By default, Unicenter TCPaccess does not place the terminal ID on any signon call.

   To activate passing terminal IDs on the security parameter list to RACF for an active Unicenter TCPaccess address space, issue the following command under ACTEST:

   ```
   XSEC TERMID ON
   ```

   You can deactivate passing terminal IDs on the security parameter list to RACF for an active Unicenter TCPaccess address space by issuing the following command under ACTEST:

   ```
   XSEC TERMID OFF
   ```

6. To enable signon checking for Telnet users, add the CPASSWORD option to the Telnet related SERVICE statement(s) in the APPCFG*xx* member for Telnet ports (typically, 23,1023).

WARNING! *Activate terminal security checking only after all RACF customization for Unicenter TCPaccess is completed and the RACLIST profiles have been refreshed (SETROPTS REFRESH TERMINAL). RACF terminal security can prevent signon to MVS, as well as Unicenter TCPaccess, if the customization is performed incorrectly.*

SAF Server Access Authorization (SERVAUTH)

Unicenter TCPaccess uses the SERVAUTH Security Access Facility (SAF) class to protect TCP/IP resources from unauthorized access. The use of SERVAUTH is optional so the customization described in this section is optional.

Port Security

Port security allows a Unicenter TCPaccess installation the ability to SAF-authorize port numbers to network applications. This provides the means for an installation to prevent rogue applications from assuming the function of a network application normally using a defined port number (access to ports can also be restricted by job name).

To enable SAF port security, a SAF resource name is specified on the bind security PORTRULE statement. Any network application attempting to bind to the specified port must be running under a userid with SAF access to the resource. The SAF entity name used is of the following form:

```
EZB.PORTACCESS.sysname.tcpname.resname
```

Where EZB.PORTACCESS is a constant, *sysname* is the value of the MVS &SYSNAME system symbol, *tcpname* is the Unicenter TCPaccess started task or job name, and *resname* is the one- to eight- character value specified on the PORTRULE statement.

For example, a port reserved with

```
PORTRULE NUMBER(111) PROTOCOL(TCP) SAF(INSTAPP)
```

on the MVS system named MVSPROD and Unicenter TCPaccess started task named ACCESS would require a SAF entity name of the following form:

Chapter

4

UNIX System Services Support

This chapter provides information necessary to configure Unicenter TCPaccess with IBM UNIX Systems Services (USS) for z/OS. We will review the BPXPRMxx statements needed to operate as a single USS network Physical File System (PFS) or to operate in a USS multiple network PFS environment called Common INET (CINET). There is one network PFS for each TCP/IP stack which is to participate in the USS environment. The network PFS provides an interface between USS and TCP/IP stack transports for a specific domain.

For additional information on configuring and using Unicenter TCPaccess USS Socket Support and IBM UNIX System Services Socket support, refer to the Unicenter TCPaccess Communication Server C/Socket Programmer Reference and the IBM documents regarding Unix System Services Planning.

The following topics are discussed:

- Using the Unicenter TCPaccess PFS Alone – Describes how to set up TCPaccess as a single PFS on the AF_INET domain.
- Configuring Unicenter TCPaccess PFS for Common INET—Discusses the changes required in the MVS UNIX System Services configuration member to support more than one AF_INET physical file system
- BPXPRMxx Configuration for TCPaccess – Describes the statements required in the USS configuration member BPXPRMxx to make TCPaccess known to UNIX System services.
- TCPaccess USS Setup Requirements – Discusses the system requirements to enable TCPaccess to function with UNIX System Services.
- TCPaccess USS PFS Startup – Describes the requirement for placing the TCPaccess PFSLOAD dataset in the OMVS procedure STEPLIB DD.
- Host Name/Address Resolution—Discusses the mechanism for address resolution under UNIX System Services.
- Installation Verification for TCPaccess and USS — Describes the Installation verification process for TCPaccess integration with USS.

Using the Unicenter TCPaccess PFS Alone

This section describes the two statements you must add or change in the USS BPXPRMxx member to configure TCPaccess as a single network PFS supporting the AF_INET domain. These statements are FILESYSTYPE and NETWORK, and are described below:

FILESYSTYPE Statement

Used to identify the Unicenter TCPaccess Physical File System (PFS) to UNIX System Services in the following format:

```
FILESYSTYPE TYPE ( pfs_name )
            ENTRYPOINT ( T010PFSA | module_name )
            PARM ( 'SYSID ( subsys_id )' )
```

TYPE(pfs_name)—Specifies a one to eight character name of the network PFS. This pfs_name will be cross-referenced in a corresponding NETWORK TYPE(pfs_name) statement and keyword. We recommend coding the TCPaccess jobname or started task name for the pfs_name. See NETWORK Statement.

ENTRYPOINT(module_name)—Specifies the TCPaccess load module name. This parameter MUST be specified as "T010PFSA". The module is located in the TCPaccess PFSLOAD dataset, which must be added to the USS startup procedure STEPLIB DD. The USS startup procedure is usually member OMVS in SYS1.PROCLIB.

PARM('SYSID(subsys_id)')—Specifies the four character subsystem id of the TCPaccess stack. The parameter within the outer parentheses must itself be enclosed in single quotes, including the inner parentheses.

Default: ACSS

NETWORK Statement

Used to assign socket domains or address families to the Unicenter TCPaccess PFS in the following format:

```
NETWORK TYPE(pfs_name )
DOMAINNAME(domain_family_name)
DOMAINNUMBER(domain_family_number )
MAXSOCKETS(number_current_sockets )
```

TYPE(pfs_name)—Specifies the pfs_name, and must exactly match the pfs_name as it appears in the FILESYSTYPE statement TYPE keyword for this NETWORK. See FILESYSTYPE , above.

DOMAINNAME(*domain_family_name*)—Specifies a one to eight character symbolic name for the domain. It is recommended to use AF_INET for domain 2, as certain name resolution functions may not work if DOMAINNUMBER(2) is not defined.

DOMAINNUMBER(*domain_family_number*)—Specifies the domain number. This parameter must be set to "2" for domain AF_INET.

MAXSOCKETS(*number_current_sockets*)—Specifies the maximum number of concurrently active sockets for the specified domain. This number should be larger than the expected total number of all concurrent application socket connections. The IBM USS default may be too small, resulting in undesired rejections of connection requests to application servers. It is recommended that this parameter be set to 64000.

Configuring Unicenter TCPaccess PFS for Common INET

This section describes statements you must add or change in the BPXPRMxx member to configure TCPaccess as a member of CINET supporting the AF_INET domain. In other words, multiple network PFSs will be configured together to support a single domain. It is important to note that CINET only functions in the AF_INET domain(2).

In a CINET environment, the BPXPRMxx statement FILESYSTYPE statement is specified differently. As with the single TCPaccess PFS configuration identified in the previous section, there is still only one FILESYSTYPE and NETWORK statement for the AF_INET domain. The difference between the single stack configuration and CINET is the additional SUBFILESYSTYPE statements, one specified for each PFS in the domain.

It is important to note that INADDRANYPORT and INADDRANYCOUNT are only applicable in a CINET environment. When using the CINET environment, port configuration for each corresponding TCPaccess belonging to the domain must be reviewed for port management conflicts. The NETWORK statement for the CINET domain uses two additional keyword parameters. These are INADDRANYPORT and INADDRANYCOUNT and they are used to reserve a set of ports for conditions where USS must control the port assignment across all TCP/IP stacks operating in the CINET environment.

When a USS socket application makes a bind() function call with the port specified as INADDRANY (port=0), USS will assign a port from the reserved set of ports and pass the port number with the bind() function request to each TCP/IP stack configured for CINET. Each stack must allocate the port for use by the socket application. If the port is not available on one of the TCP/IP stacks, the original bind() function request will fail. If any of the TCP/IP stacks has a port assignment configuration that overlaps with ports reserved for USS, then duplicate port assignment errors will occur whenever one of the stacks has already allocated a port that USS wishes to use.

Therefore, it is very important that the ports reserved for CINET do not overlap with any TCP/IP stack port assignment configuration statements.

The TCPaccess statement to be verified is the TCP statement located in the TCPCFGxx configuration member. The TCP statement keyword PORTASGN parameter must specify a range of ports that does not overlap the INADDRANYPORT specification.

For more information on common INET support, read the IBM documents regarding Unix System Services Planning.

The first required statement for CINET is the FILESYSTYPE statement. Only the TYPE and ENTRYPOINT keywords are specified.

FILESYSTYPE Statement

Used to define CINET to UNIX System Services in the following format:

```
FILESYSTYPE TYPE(cinet_name) ENTRYPOINT( module_name )
```

TYPE(*cinet_name*)—Specifies the name of the CINET environment. The name is specified with one to eight characters. This CINET name will be referenced by the TYPE keyword parameter in the associated SUBFILESYSTYPE and NETWORK statements. This parameter should be set to "CINET".

ENTRYPOINT(*module_name*)—Specifies the entry point for the PFS. This parameter *MUST* be specified as "BPXTCINT".

The SUBFILESYSTYPE statement is required to identify each individual PFS to CINET

:

SUBFILESYSTYPE Statement

Used to define the Unicenter TCPaccess PFS to CINET in the following format:

```
SUBFILESYSTYPE NAME (pfs_name)
               TYPE (cinet_name) ENTRYPOINT(module_name)
               PARM ('SYSID(subsys_id)')
```

NAME(*pfs_name*)—Specifies a one to eight character name of the network PFS. This PFS name will be referenced in a corresponding NETWORK statement TYPE(*pfs_name*) keyword. This parameter must be the same as the TCPaccess Jobname or Started Task name.

TYPE(*cinet_name*)—Specifies the name of the CINET environment. The name is specified with one to eight characters. This cinet_name must match the cinet_name specified by the TYPE keyword in the previous FILESYSTYPE statement. These parameters should be set to "CINET".

ENTRYPOINT(*module_name*)—Specifies the TCPaccess load module name. This parameter MUST be specified as "T010PFSA". The module is located in the TCPaccess PFSLOAD dataset which must be added to the USS startup procedure STEPLIB DD. The USS startup procedure is usually member OMVS in SYS1.PROCLIB.

PARM('SYSID(*subsys_id*)')—Specifies the four character Subsystem ID of the TCPaccess stack. The parameter within the outer parentheses must itself be enclosed in single quotes, including the inner parentheses.

The NETWORK statement is required to establish connectivity characteristics:

NETWORK Statement

Used to assign the socket domain or address family, and connection characteristics for CINET in the following format:

```
NETWORK     TYPE(cinet_name)
            DOMAINNAME(domain_family_name)
            DOMAINNUMBER(domain_family_number)
            MAXSOCKETS(number_current_sockets)
            INADDRANYPORT(first_port_number)
            INADDRANYCOUNT(number_ports)
```

TYPE(*cinet_name*)—Specifies the name of the CINET environment. The name is specified with one to eight characters. This cinet_name must match the cinet_name specified by the TYPE keyword in the previous FILESYSTYPE statement. These parameters should be set to "CINET".

DOMAINNAME(*domain_family_name*)—Specifies a one to eight character symbolic name for the domain. It is recommended to use AF_INET for domain 2, as certain name resolution functions may not work if DOMAINNUMBER(2) is not defined.

DOMAINNUMBER(*domain_family_number*)—Specifies the domain number. This parameter must be set to "2" for domain AF_INET.

MAXSOCKETS(*number_current_sockets*)—Specifies the maximum number of concurrently active sockets for the specified domain. This number should be larger than the expected total number of all concurrent application socket connections. The IBM USS default may be too small, resulting in undesired rejections of connection requests to application servers. It is recommended that this parameter be set to 64000.

INADDRANYPORT(*first_port_number*)—Specifies the first port in the set of ports reserved for USS port allocations for socket applications making a bind() function call with a port number set to INADDR_ANY (port 0). This parameter must be specified greater then 1024. The suggested specification for INADDRANYPORT is 60000 with INADDRANYCOUNT specified as 4000. This port range may need to be expanded depending on the number of network applications requiring support for INADDR_ANY.to be expanded depending on the number of network applications requiring support for INADDR_ANY..

INADDR ANYCOUNT(*number_of_ports*)—Specifies the number of ports reserved for USS CINET bind() function processing. This parameter defines the range of ports for INADDRANYPORT. It is suggested to use the number 4000 to begin with.

BPXPRMxx Configuration for TCPaccess

The following examples provide the statements to be included in the USS BPXPRMxx configuration. The examples demonstrate how the statements, keywords and parameters could be coded. In addition to the statements identified in the example, other USS statements may be required for proper USS operation. Refer to the UNIX System Services Planning and the MVS Initialization and Tuning Reference manuals for a detailed explanation of BPXPRMxx statements.

Example 1 Configuration for a single Unicenter TCPaccess PFS

```
FILESYSTYPE TYPE(RUNTCP)
            ENTRYPOINT(T010PFSA)
            PARM('SYSID(ACSS)')
NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
        MAXSOCKETS(64000)
        TYPE(RUNTCP)
```

Example 2 Configuration for multiple Unicenter TCPaccess PFSs on CINET

```
FILESYSTYPE TYPE(CINET) ENTRYPOINT(BPXTCINT)
NETWORK DOMAINNAME(AF_INET)
        DOMAINNUMBER(2)
        INADDRANYPORT(60000)
        INADDRANYCOUNT(4000)
        MAXSOCKETS(64000)
        TYPE(CINET)

SUBFILESYSTYPE NAME(RUNTCP)
               PARM('SYSID(ACSS)')
               TYPE(CINET)
               ENTRYPOINT(T010PFSA)
               DEFAULT
SUBFILESYSTYPE NAME(RUNTCP1)
               PARM('SYSID(ACS1)')
               TYPE(CINET)
               ENTRYPOINT(T010PFSA)
SUBFILESYSTYPE NAME(RUNTCP2)
               PARM('SYSID(ACS2)')
               TYPE(CINET)
               ENTRYPOINT(T010PFSA)
```

TCPaccess USS Setup Requirements

In order for TCPaccess to function properly with USS, the following steps MUST be implemented.

1. The TCPaccess address space must be defined to your security system with a valid OMVS security segment.
2. If two or more TCPaccess stacks are running in one LPAR, a SYSUNIQ statement must be added to each stack's TCPCFGxx configuration.
3. If one or more TCPaccess stacks are configured in USS CINET, then each stack's TCPCFGxx must contain PORTASGN statements specifying ports that do not overlap with the BPXPRMxx NETWORK INADDRANYPORT and INADDRANYCOUNT specification.

Note: Refer to the TCPaccess Customization Guide for details regarding configuring the TCPCFGxx SYSUNIQ and TCP PORTASGN statements identified above.

TCPaccess USS PFS Startup

The z/OS OMVS cataloged procedure is used to startup the USS kernel and the TCPaccess PFS. During the installation of TCPaccess, the PFSLOAD data set was created and the TCPaccess PFS and other USS based applications are placed into this dataset. The PFSLOAD data set must be added to the STEPLIB DD statement in the OMVS cataloged startup procedure. One may put the PFSLOAD data set in the linklist, but this is not recommended. The PFSLOAD data set must be APF authorized. When OMVS is started and the TCPaccess PFS successfully initializes, the following message will be displayed on the system console:

T01OE004I Connection To OpenEdition Established - Provider *pfs_name*

Host Name/Address Resolution

If you will be using Unicenter TCPaccess UNIX System Services sockets, the method used to resolve host names and addresses is different than that used for the Unicenter TCPaccess socket API. OpenEdition MVS Version 1.2 uses the LE/370 Version 1.3 and 1.4 runtime libraries to perform certain socket related functions such as gethostbyname(), and getprotobyname(). The LE/370 runtime library (RTL) reads specific MVS data sets to map services to names and to obtain domain name resolution configuration information.

The LE/370 Version 1.5 runtime library uses members in the /etc directory under OMVS to perform these functions. This is similar to UNIX configurations.

See the Unicenter TCPaccess Communications Server C/Socket Programmer Reference for a complete description of the method and the data set members you need to perform these socket functions.

Defining a SYSUNIQ statement in your TCPCFGxx may also be necessary, as noted above.

Installation Verification for TCPaccess and USS

This section describes the Installation verification process for TCPaccess integration with USS. The procedure requires editing the sample IVPUSSJ JCL located in the SAMP installation data set and submitting the JCL. The first step "SETAFF" will create an affinity between TCPaccess and the IVP application invoked in the following RUNIVP procedure. The affinity only occurs within a USS CINET environment, otherwise the SETAFF step can be deleted, commented out, or the SETAFF return code can be ignored. If TCPaccess is not configured in CINET, then SETAFF will return with RC=8. Under this condition, the RC=8 can be ignored. If TCPaccess IS configured with CINET, a SETAFF RC=8 means the SETAFF PARM= parameter does not match the PFS name for TCPaccess as defined in BPXPRMxx configuration. If this is the case, make the two specifications match and re-submit. The return code for the RUNIVP GO step must always be zero, although the port assignments identified in the job output may be different. If a single TCPaccess PFS is configured, the port assignments come from the TCPaccess TCPCFGxx TCP PORTASGN statement. If CINET is configured, the port assignments come from the BPXPRMxx NETWORK INADDRANYPORT and INADDRANYCOUNT statement. If more than one TCPaccess PFS is configured for CINET, then repeat the test for each PFS.

The following steps are required to run the IVPUSSJ test procedure:

1. Add a proper job card to IVPUSSJ.
2. Change the PARM=RUNJOB statement to match the USS PFS name and TCPaccess Started task or Job name.
3. Change INFILE= to the name of the TCPaccess installation data set name for SAMP.
4. Submit the JCL and review the output.

Chapter 5

User Exits

This chapter provides information about writing exit routines for Unicenter TCPaccess. It includes these sections:

- The Exits—Defines data areas, macro instructions, and coding conventions and restrictions that apply to all user exit routines in Unicenter TCPaccess
- Converting from TCPX0001 and TCPX0002—Describes how to use the SYNRCVD and SENDSYN exit points in Unicenter TCPaccess 5.2 to control inbound and outbound TCP connection requests if you have exits TCPX0001 and TCPX0002 coded in prior releases of Unicenter TCPaccess

The Unicenter TCPaccess exit facility lets users write exit routines to handle certain specialized requirements within their installation.

User Exit Points

Various user exit points are defined within Unicenter TCPaccess to allow the product to be customized. You can configure multiple exit programs for the various exit points. An exit program can communicate with itself across various exit point invocations by establishing an exit context. Exit programs can issue messages, accept or reject various requests, and change or reroute messages. Exit points are defined in message services, at various points in the TCP/IP stack, and in FTP.

The following table lists defined user exit points. These exits are configured in the IJTCFG*xx* member of the PARM data set. See the *Customization Guide* for details on configuring the exit points.

Exit Point	When Invoked	Function
FTPCMND	An FTP command is received	The exit can reject the command
FTPLOGIN	An FTP login is received	The exit can reject the request
FTPRSRCE	An FTP command has been received which will cause a dataset allocation	The exit can reject the request
INIT	Startup	Initialize the exit environment
LOG	A message is formatted	The exit can change the message text, reroute the message, or suppress the message
RAWRECV	A RAW IP datagram is received	The exit can reject the datagram
RAWSEND	A RAW send request is received	The exit can reject the request
RAWSOCK	A RAW socket is created	The exit can reject the request
SENDSYN	A TCP syn packet is about to be sent	The exit can reject the request
SMFEXIT	An SMF record is about to be written	The exit can reject the writing of the record
SYNRCVD	A TCP syn packet is received	The exit can reject the connection
TCPBIND	A TCP bind request is received	The exit can reject the request
TCPCLOSE	A TCP connection is closed	Information only
TCPESTAB	A new TCP connection is established	Information only
TERM	Shutdown	Terminate the exit environment
UDPBIND	A UDP bind request is received	The exit can reject the request
UDPRECV	A UDP datagram is received	The exit can reject the datagram

Exit Point	When Invoked	Function
UDPSEND	A UDP send request is received	The exit can reject the request
VTAMBIND	When a BIND RU is received by the TN3270E server	Allow or reject the BIND request

Parameters

Except where noted in the following discussion, parameter lists and the data areas they point to should be left unchanged by the exit program. Changes to other fields are ignored and will not be made effective.

Exit Point ID

Each exit is passed a parameter list pointed to by R1, which includes a fullword identifying the exit point. The first word of the parameter list will be one of the following:

F'0'	INIT exit
F'1'	TERM exit
F'2'	LOG exit
F'3'	TCPBIND exit
F'4'	SYNRCVD exit
F'5'	SENDSYN exit
F'6'	TCPESTAB exit
F'7'	TCPCLOSE exit
F'8'	UDPBIND exit
F'9'	UDPSEND exit
F'10'	UDPRECV exit
F'11'	RAWSOCK exit
F'12'	RAWSEND exit
F'13'	RAWRECV exit
F'14'	FTPLOGIN exit
F'15'	FTPRSRCE exit
F'16'	FTPCMND exit

F'21'	VTAMBIND exit
F'22'	SMFEXIT exit

Issuing Messages from Exits

Each exit except the LOG exit is passed the address of a routine that can be invoked in order to write a message to the log or to the operator.

When calling this routine, the following must be provided:

R00	Must contain the value of R13 on entry to the exit program.
R01	Must point to a one-byte message type, followed by 80 bytes of message text. For a description of the message types, see the *Customization Guide*.
R13	Must point to a standard 72-byte register save area.
R14	Must contain the return address.

The message is prefixed by the exit facility with a standard Unicenter TCPaccess message ID (T00EX004) and the exit program name.

Exit Context

At each exit point except INIT, the program is passed a fullword of context. This context word is provided by the program at the INIT exit point, and can be used by the exit program to communicate across exit points.

Return Codes

Two return codes are defined for exits:

- **Return Code 1:** Returned by the exit in R15 when returning to the exit facility. This return code determines whether the next configured exit program is called for this exit point, if multiple exits are configured. This return code should normally be set to zero to allow subsequent exit programs to be invoked.

  A non-zero value causes the exit facility to bypass calling any subsequent exit programs configured at that exit point. (Return code 1 does not apply to the INIT and TERM exits. Register 15 should be set to zero when returning from these exit points.)

- **Return Code 2:** Is part of the parameter list provided on entry to the exit. The exit can set this return code, when appropriate, to reject the current request. For instance, the SYNRCVD exit can set a non-zero value in this word to reject the connection request. If multiple exit programs are configured at an exit point, the Return Code 2 value is passed from one exit program to the next.

  The value in Return Code 2 after the last exit is called is the value that is returned to the caller of the exit facility (such as TCP). An exit program, therefore, should not change the Return Code 2 value, and, in particular, should not change it from a non-zero to a zero value, unless there is a compelling reason.

Care should be used when setting return codes and when configuring exits. The exit programs are invoked in the same order in which they are configured. A subsequent program can change the Return Code 2 setting from an earlier exit program. An earlier exit program can prevent a subsequent exit program from being called by using Return Code 1.

Exit Work Area

With the exception of the INIT exit point, if requested, the exit facility provides a work area to the exit program upon each invocation. Request the work area with the EWASIZE parameter of the EXIT statement in member IJTCFG*xx*, or with a parameter returned by the program at the INIT exit point (the program parameter takes precedence).

The Exit Work Area (EWA) addressed is passed to the exit program at each exit point as the fifth parameter, following the address of the message-writing routine.

The following entry is added to each exit parameter list table:

+16	04	var	Address of the Exit Work Area (or zero).

The EWA size can be from 1 to 65532 bytes. It is taken from pooled storage. The smallest pool that satisfies the requested size will be used. Pool usage can be monitored via the IFS POOL command.

One of the following pools will be used:

Pool Name	EWA Size
256B	1-252
512B	253-508
01KB	509-1020
04KB	1021-4092
08KB	4093-8188
16KB	8189-65532

Note: These figures are provided to enable the exit program designer to make efficient use of the EWA buffers. It is important that the exit use only the size requested on the EWALENGTH parameter, or by the INIT exit. The exit facility monitors the exit program's use of the EWA, and will force an ABEND if an overrun is detected, *even if space remains in the buffer*.

Exit Recovery Routine

The exit program can supply the address of a recovery routine via a parameter returned at the INIT exit point. The exit facility calls this routine in the event of an ABEND in the exit program. The recovery routine is called in the same mode as the abending exit program, and is passed the System Diagnostic Work Area (SDWA) address and the EWA address (if any). Since the exit's recovery routine is called after the system's Recovery and Termination Manager (RTM) has finished processing the abend, it should perform only local cleanup functions. Any updates to the SDWA are ignored. The recovery routine should **not** attempt to free the SDWA, since this is done by the exit facility.

Recovery Exit

Exit Point: When an abend has occurred in an exit program.

Function: Perform cleanup associated with the exit program.

Dispatchable Unit: Identical with the abending exit program.

Register contents are shown in the following table.

Register	Contents on Entry
R00	SDWA (if processing under an SRB, a copy of the SDWA)
R01	Exit Work Area address (if applicable)
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Note: On return from the recovery exit, R13 must be restored.

Exit Parameter List Mapping Macro—T00DEXPL

A new macro, T00DEXPL, is supplied to map the parameter lists to the exit program at the various exit points. The macro is distributed in the SAMP library.

Using the IEFUSI Sample Exit

When a single Unicenter TCPaccess region is to service many application users, it may require virtual storage beyond the default provided in most installations. Specifying a private area REGION size greater than 16 MB, however, can cause storage allocation problems for system resources below the 16 MB line. In these instances, you may find it necessary to implement a user exit, such as IEFUSI, to ensure that adequate region values are supplied for Unicenter TCPaccess operation. Source for a sample IEFUSI exit is provided in the TCPSAMP distribution data set.

Note: This is a sample only. The region values should be modified to fit your installation's requirements.

The Exits

This section describes the individual exits.

There is a sample EXIT program in the HLQ.SAMP data set.

INIT Exit

Exit Point: Unicenter TCPaccess startup.

Function: The exit is called synchronously at startup, during parsing of the configuration statements in the IJTCFG00 configuration member. You can use it to create an exit context that is passed to the exit program at subsequent exit points. It can also issue messages.

Addressing Mode: 31

Dispatchable Unit: Task mode (TCB)

Restriction: The exit should not block indefinitely.

Register Contents at Entry:

00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Zero

INIT Exit Parameters Passed.

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'0'.
+04	04	--	Exit context. This word contains zeros on input. The INIT exit can place a word of context in this word. The context is then passed to the exit program at other exit points.
+08	04	--	Return code 2. This should be set to zero.
+12	04	--	Address of the message routine. Note, since the log is not allocated when this exit is called, the message type should be one that will be written by WTO. The LOG exit point can be used to ensure that the message written (T01EX004) is routed to the console.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	04	Address of a word in which the exit may define which exit points will be driven (except the TERM exit, which is always driven).

Offset	Parm Length	Data Length	Description
			This word can be built by ORing the exit point ID flag values for the exits that are to be driven. Exit point flags are defined as follows: X'80000000' LOG exit X'40000000' SMFEXIT exit X'00800000' TCPBIND exit X'00400000' SYNRCVD exit X'00200000' SENDSYN exit X'00100000' TCPESTAB exit X'00080000' TCPCLOSE exit X'00040000' UDPBIND exit X'00020000' UDPSEND exit X'00010000' UDPRECV exit X'00008000' RAWSOCK exit X'00004000' RAWSEND exit X'00002000' RAWRECV exit X'00000080' FTPLOGIN exit X'00000040' FTPRSRCE exit X'00000020' FTPCMND exit X'00000010' VTAMBIND exit For instance, to drive the TCPESTAB, TCPCLOSE, and FTPLOGIN exits, set the value of this word to X'00180080'. **Note:** This request may be overridden by the configuration. See the discussion of the EXIT statement in the *Customization Guide*.

Offset	Parm Length	Data Length	Description
+24	04	Var	Address of the PARM string from the EXIT configuration statement. **Note:** This area is released following the exit point. If the exit program wants to save this string, it must get storage and make a copy.
+28	04	04	Address of a fullword containing the length of the PARM string from the EXIT configuration statement.
+32	04	04	The address of the four-byte Unicenter TCPaccess subsystem ID.
+36	04	04	The address of a fullword area in which the exit program may put the size of the EWA to get.
+40	04	04	The address of a fullword area in which the exit program may put the address of a recovery routine to be called in the event the exit program abends at a subsequent entry point.
+44	04	04	Four-byte product version in C'0v0r' format. For example, Release 5.2 would be x'F0F5F0F2'
+48	04	32	MF subtype mask. Each bit in the mask corresponds to an SMF record subtype 0-255. If the SMF exit point is requested, the exit can turn on those bits corresponding to the SMF record subtypes for which it wants to gain control. The SMF exit point is called for these subtypes, in addition to any SMF subtypes configured on the SMF statement in IJTCFG*xx*. Turning on a bit only causes the exit point to be driven; it does not effect whether the record is written to the SMF data set.

SMF exit

Exit point: When an SMF record is about to be written.

Function: Allow or suppress writing the record.

Addressing Mode: 31

Dispatchable Unit: Task mode or SRB mode.

Restrictions: Normal restrictions for SRB-mode processing. No SVCs may be issued.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

SMFXIT Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'22'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be set to one of the following: 00 - Write the record 04 - Do not write the record.
+12	04	--	Address of the message routine.
+16	04	--	Address of the exit work area
+20	04	08	Address of the SMF record

TERM Exit

Exit Point: Unicenter TCPaccess shutdown.

Function: The exit is called synchronously at shutdown. It can be used to terminate the exit environment and clean up.

Addressing Mode: 31

Dispatchable Unit: Task mode (TCB)

Restrictions: The exit should not block indefinitely.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Zero

TERM Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'1'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be left unchanged.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero)

LOG Exit

Exit Point: When a message has been formatted and is ready to be written.

Function – The exit can change the message text, reroute the message, or suppress the message.

Addressing Mode: 31

Dispatchable Unit: Task mode PC routine or SRB

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

LOG	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

LOG Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'2'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be left unchanged. To suppress the message, set the message routing code to zero (see below).
+12	04	--	Zeroes.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	var	Address of the message text. The text may be changed by the exit.
+24	04	04	Address of a word containing the message buffer length. **Note:** This value should not be changed; the message cannot be made any longer than this length. To shorten the message, pad to the right with blanks.
+28	04	04	Address of a word containing a message routing code: X'00000000' Do not issue message X'00000004' Write message to log X'00000008' X'0000000C' Write to log and console. This word may be changed by the exit to change the routing for the message. To suppress the message, set this word to zero.

VTAMBIND Exit

Exit point: When a BIND RU is received by the TN3270E server.

Function: Allow or reject the BIND request.

Addressing Mode: 31

Dispatchable Unit: SRB mode (the VTAM SCIP exit).

Restrictions: Normal restrictions for SRB-mode processing. No SVCs may be issued.

Register Contents at Entry:

R00	Exit point Id
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

VTAMBIND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'21'.
+04	04	--	Exit context.
+08	04	--	Return code 2. This should be set to one of the following: 00 – Accept the BIND 04 – Reject the BIND.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet
+24	04	08	Address of the remote host AF_Inet
+28	04	04	Address of the User ID.
+32	04	04	Address of the TCP session number.
+36	04	08	Address of the SLU name.
+40	04	08	Address of the PLU name.
+44	04	08	Address of the requested application.
+48	04	36	Address of the first 36 bytes of the BIND image (mapped by ISTDBIND).

Stack Exits

This section describes the stack exits. These are exit points in TCP, UDP, and RAW IP processing. These exits can be called in SRB mode or in cross-memory PC mode, and must not issue any SVCs. No assumptions should be made about the cross-memory environment. Home, primary, and secondary address spaces may all differ.

In most cases, these exits can reject a request. All can issue a message, and all have access to the exit context.

The parameter list in each case is similar—in each case, the addresses of the local host address and the remote host address are provided. This is eight-bytes in standard AF_Inet form:

Offset	Length	Field
00	02	Address Family
02	02	Port number (hexadecimal)
04	04	IP address (hexadecimal)

Not all of these fields are filled in for every exit point. For instance, at the bind exit point, only the local port number is filled in.

Each stack exit is passed the address of a 24-byte user identification area. This area identifies the cross-memory TLI, UNIX System Services, or IUCV user. The format is as follows:

Offset	Length	Field
00	08	Jobname
08	08	Stepname
16	08	Procstepname

TCPBIND Exit

Exit Point: When a TCP bind request is received.

Function: This exit can accept or reject the request.

Addressing Mode: 31.

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

TCPBIND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'3'.
+04	04	--	Exit context.
+08	04	--	Return code 2.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

SYNRCVD Exit

Exit point: When a TCP connection request (syn packet) is received.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

SYNRCVD Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'4'.
+04	04	--	Exit context.
+08	04	--	Return code 2.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

SENDSYN Exit

Exit Point: When a TCP connection request (syn packet) is about to be sent to a remote host.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

SENDSYN Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'5'.
+04	04	--	Exit context.
+08	04	--	Return code 2.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

TCPESTAB Exit

Exit Point: When a new TCP connection is fully established.

Function: This exit point is intended for information only. The exit cannot reject the connection at this point.

Addressing Mode: 31

Dispatchable Unit: Task mode or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

TCPESTAB Exit Parameters Passed:

Offset	Parm length	Data length	Description
+00	04	--	Exit point ID. This word contains F'6'.
+04	04	--	Exit context.
+08	04	--	Return code 2.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

Register Contents at Entry:

R00	Exit point
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined

TCPCLOSE Exit Parameters Passed:

Offset	Parm length	Data length	Description
+00	04	--	Exit point ID. This word contains F'7' .
+04	04	--	Exit context.
+08	04	--	Return code 2.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

UDPBIND Exit

Exit Point: When a UDP bind request is received.

Function: This exit can accept or reject the request.

Addressing Mode: 31.

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

UDPBIND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'8'.
+04	04	--	Exit context.
+08	04	--	Return code 2.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

UDPSEND Exit

Exit Point: When a UDP datagram is about to be sent.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

UDPSEND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'9'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to abort the send.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

UDPRECV Exit

Exit Point: When a UDP datagram has been received.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Rest

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

UDPRECV Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'10'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to reject the datagram.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

RAWSOCK Exit

Exit Point: When a RAW socket request was received.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

RAWSOCK Exit Parameters Passed :

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'11'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to reject the request.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

RAWSEND Exit

Exit Point: When a RAW IP datagram is about to be sent.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

RAWSEND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'12'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to abort the send.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

Note: If you need additional information, refer to the mapping macro in the SAMP library.

RAWRECV Exit

Exit Point: When a RAW IP datagram has been received.

Function: This exit can accept or reject the datagram.

Addressing Mode: 31

Dispatchable Unit: Task mode PC or SRB mode.

Restrictions: The exit should not block execution. The exit must not issue any SVC requests.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

RAWRECV Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'13'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to reject the datagram.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	24	Address of the User Identification.

FTP Exits

Following are the FTP exit points. These exits are invoked in task mode, but **must not** block execution on the TCB.

These exits can be used to issue a message or reject an access request.

Note: The SMFEXIT should be used in place of an FTP termination exit. The SMF exit is called just before an SMF record is written for an FTP session that is ending. See SMF exit for more information.

FTPCMND Exit

Exit Point: When an FTP command is received by the FTP server.

Function: At this exit point, the exit program can accept or reject the command. If the command is rejected, an FTP error reply will be sent.

Addressing Mode: 31.

Dispatchable Unit: Task mode.

Restrictions: The exit program must not block execution.

Register Contents at Entry:

R00	Exit point
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

FTPCMND Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'16'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to reject the command.
+12	04	--	Address of the message routine.
+16	04	Variable	Address of the Exit Work Area
+20	04	08	Address of the local host AF_Inet value.
+24	04	08	Address of the remote host AF_Inet value.
+28	04	08	Address of the User ID.
+32	04	08	Address of the eight-byte user ID of the user logged in.
+36	04	08	Address of the eight-byte command received by the FTP server.
+40	04	80	Address of an 80-byte area in which the exit program can optionally place the halfword length (maximum 78 bytes) of the reply text, followed by the text. If the command is rejected by the exit, and the reply text length is non-zero upon return from the exit, the FTP server appends the text provided to an FTP error reply. Otherwise, the reply will be "500 xxxx command rejected by user exit".

FTPLOGIN Exit

Exit Point: When an FTP user is logging in (when the PASS or ACCT command is received). The exit is called before the SAF router is called to log the user on.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode.

Restrictions: The exit must not block execution.

Register Contents at Entry:

R00	Exit point
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

FTPLOGIN Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'14'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to reject the login.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04	08	Address of the remote host AF_Inet.
+28	04	08	Address of the User ID.

FTPRSRCE Exit

Exit Point: When an FTP user requested an operation that will result in a data set resource allocation. This exit is called before the SAF router is invoked to verify the user's access authority.

Function: This exit can accept or reject the request.

Addressing Mode: 31

Dispatchable Unit: Task mode.

Restrictions: The exit must not block execution.

Register Contents at Entry:

R00	Exit point ID
R01	Parameter list address
R02-R12	Zeros
R13	Save area address
R14	Return address
R15	Entry point address

Register Contents on Return:

R00-R12	Undefined
R13	Restored
R14	Undefined
R15	Return Code 1

FTPRSRCE Exit Parameters Passed:

Offset	Parm Length	Data Length	Description
+00	04	--	Exit point ID. This word contains F'15'.
+04	04	--	Exit context.
+08	04	--	Return code 2. The exit can set this word to a non-zero value to reject the request.
+12	04	--	Address of the message routine.
+16	04	var	Address of the Exit Work Area (or zero).
+20	04	08	Address of the local host AF_Inet.
+24	04 04	52 1024	**MVS:** Address of the 44-byte dataset name, followed by an eight-byte member name (or blanks). **HFS:** Address of the file name (1024 bytes, padded on the right with blanks).
+28	04	52	Address of the 44-byte dataset name, followed by an eight-byte member name (or blanks).
+32	04	06	Address of the volume serial number, or blanks.
+36	04	01	Address of a one-byte field identifying the authorization request type: X'40' Read X'20' Write X'10' Alter
+40	04	02	Address of a two-byte field identifying the FTP operation type: X'8000' HELP command

Offset	Parm Length	Data Length	Description	
			X'4000'	STAT command (Statistics)
			X'2000'	STOR command (Store)
			X'1000'	APPE command (Append)
			X'0800'	RETR command (Retrieve)
			X'2400'	STOU command (Store Unique)
			X'0200'	LIST command (Directory list)
			X'0100'	NLST command (Directory list)
			X'0080'	RNTO command (Rename)
			X'0040'	DELE command (Delete)
			X'0020'	Special PDS directory allocation for LIST/NLST.
			X'0010'	MKD command (Make Directory)
+44	04	var	Address of the ACEE. The ACEE is mapped by macro IHAACEE. Its length is in the ACEELEN field.	

Note: See the mapping macro in the SAMP library for additional information.

Converting from TCPX0001 and TCPX0002

Users of prior releases of Unicenter TCPaccess who coded exits TCPX0001 and TCPX0002 to control inbound and outbound TCP connection requests, can use the SYNRCVD and SENDSYN exit points in this release of Unicenter TCPaccess to accomplish the same purpose. The exit programs will have to be modified to conform to the new exit facility.

The following is a brief guide to help you modifying exits TCPX0001 and TCPS0002.

Programming Limitations

The SYNRCVD and SENDSYN exits are called in either SRB mode or cross-memory stacking PC mode, and are subject to the restrictions imposed by those environments. Supervisor calls (SVCs) are not allowed.

Exit Parameter List

The XITDEPL macro is not supported. The exit parameter lists vary among exit points. Use the T00DEXPL macro to map the parameter lists.

Program Entry

The XITMENT macro is not supported. Instead, use standard linkage conventions for saving registers and establishing addressability. General register 13 contains the address of an 18-fullword register save area on entry to all exit points. Registers 14 and 15 contain the return address and entry point address of the exit program. The exit program can use all registers; it is not necessary to maintain the contents of any register during execution.

Program Exit

The XITMRET macro is not supported. Use standard linkage conventions to return to the exit facility. It is not necessary to restore any registers except register 13 before return to the exit facility. Place the return code in register 15 before returning.

Obtaining Storage

The XITMFUNC GETMEM and FREEMEM services are not supported. Use STORAGE OBTAIN/RELEASE to obtain and release storage. Do **not** use GETMAIN/FREEMAIN, as these macros generate SVC calls.

Writing Messages

The XITMFUNC WRITELOG function is not supported. Instead, the address of a message service routine is passed to the exit as part of the parameter list.

To issue a message, call this routine using standard linkage conventions. Provide in register 1 the address of an 81-byte field in the form:

```
CL1'message_severity',CL80'message_text'
```

See the chapter "Overview of Messages" in the *Prefixed Messages Guide* for a list of the message severities.

Tracing

The XITMFUNC TRACE function is not supported. A trace entry is written at entry to and return from each exit program. No facility exists for exit programs to write trace entries to the Unicenter TCPaccess internal trace table.

Setting Return Values

The XITMSET macro is not supported. Where appropriate, the exit parameter lists can be updated directly by the exit program. It is important to note that, unlike TCPX0001 and TCPX0002, multiple exit programs can be run at each exit point. The value returned in general register 15 (return code 1) determines whether subsequent exit(s) are called at that entry point. The value returned in the return code field in the parameter list (return code 2) determines whether the request is allowed or rejected. Set this field to a non-zero value to cause the request to be rejected.

Note: If the return code 2 value is non-zero on entry to the exit, it indicates that another exit program at that exit point determined to reject the request. Setting the return code 2 field from a non-zero value to a zero effectively reverses the decision of a previous exit. Use care in setting the return code 2 value.

Exit Naming Conventions

There is no restriction placed on the name of the exit programs.

Dependencies Between Exits

As with TCPX0001 and TCPX0002, exit programs should not assume that an earlier exit point was called, or that it was successful. The exit context word is provided at each exit point to allow the exit program to communicate across exit points.

Appendix

A

Editing Tools For Installation

This chapter describes the edits you must make before installation and how to use the tools necessary for these editions.

It covers the following subjects:

- Setting Up the SMP Environment—Describes how to allocate common load data sets
- Making Global Changes with the ISPF Editor—Describes how to use the ISPF editor to edit the ALLOCSMP job
- Updating the TCPNAMES ISPF Edit CLIST—Describes how to edit the TCPNAMES member in the CNTL data set to globally change strings used by the ALLOC*xxx* job streams

Setting Up the SMP Environment

The first time you install Unicenter TCPaccess, you must set up the SMP environment. To do this, you must edit some of the members to allocate common load data sets.

There are two ways to allocate the common data sets:

- Making Global Changes with the ISPF Editor
- Updating the TCPNAMES ISPF Edit Clist

Globally edit the following symbols using values appropriate for your system configuration:

HOLDCL	DSTINDX	DSTUNIT
DSTVOL	LNKINDX	SMPINDX
SMPUNIT	SMPVOL	TLBUNIT
TLBVOL	TRGINDX	TRGUNIT
TRGVOL		

All Unicenter TCPaccess software products require that you establish a common environment for your installation.

The process includes these tasks:

- Allocating and initializing the SMP and common data sets for Unicenter TCPaccess
- Allocating product-dependent data sets for optional products

Making Global Changes with the ISPF Editor

This is the format of the global change command if you are using the ISPF editor to edit the ALLOCSMP job:

```
c 'old_string' 'new_string' all
```

To change the string SMPINDX to TCPACSS, enter this command:

```
c 'SMPINDX' 'TCPACSS' all
```

If you are changing the HOLDCL symbol to an asterisk in order to use the same SYSOUT class as specified on the JOB statement MSGCLASS parameter, make sure the asterisk is enclosed in single quotes in your global change command, as shown below:

```
c 'HOLDCL' '*' all
```

Important! *Do not modify job ALLOCSMP to use a predefined Consolidated Software Inventory (CSI) in which other software products are installed. That is, Unicenter TCPaccess software products must be installed in their own target and distribution zones.*

Note: Expect return code zero from each step of ALLOCSMP.

Updating the TCPNAMES ISPF Edit CLIST

TCPNAMES is an ISPF Edit CLIST in the CNTL data set. TCPNAMES globally changes all the strings used by the ALLOC*xxx* job streams. TCPNAMES inserts a jobcard and updates the job with your local variables.

The TCPNAMES Clist requires ISPF Version 2 or higher. If the TCPNAMES CLIST is used as distributed, all data sets must have the same high-level qualifier except for the LINK data set, which is prefixed by SYS1. plus the high-level qualifier used for the other data sets.

1. Edit the JOBCARD member in the CNTL data set, changing the JOB statement to match your site's requirements.
2. Copy member TCPNAMES to a fixed length record format CLIST data set that is in your TSO SYSPROC DD concatenation.
3. As you edit the ALLOC*xxx* and other installation members, update the variables in the ALLOC*xxx* job with the parameters passed through this TCPNAMES command:

```
TCPNAMES high_level disk_vol disk_unit tape_vol tape_unit
```

TCPNAMES	Is the command to use to edit the TCPNAMES CLIST.
high_level	Indicates the data set high-level qualifier.
disk_vol	Specifies the disk volume where the data sets are to be created.
disk_unit	Indicates the disk unit type of the volume.
tape_vol	Specifies the volume serial number of the installation tape.
tape_unit	Indicates the address of the tape unit to which the installation tape is assigned.
Sms_clas	Specifies the SMS storage class for PDSE libraries. (Alternatively, you can modify the change statements within TCPNAMES which will allow you to execute the CLIST without any parameters.)

Usage Notes:

Always run the CLIST TCPNAMES from the command line of the member for every job prior to submitting the job.

If the high-level qualifier you are using is TCPACSS, the disk volume serial number is MVS001, the disk unit type is 3390, the tape volume serial number is TCPACSS, and the tape unit is TAPE, then use this primary line command to update variables:

```
TCPNAMES TCPACSS.Vxxx MVS001 3390 TAPE
```

This changes all the strings that need to be changed in the job and replaces the JOB statement with the one in the JOBCARD member.

Note: To run the ALLOC*xxx* jobs, it is not necessary to substitute variables for the *tape_vol* or *tape_unit* parameters. They are used in INSTSMPE.

Appendix

B

Configuring for Cisco Routers

This appendix provides information to help you with the setup and configuration of Unicenter TCPaccess with Cisco 7000 and 7500 routers. It also includes the steps necessary to configure the software for use with GateD Fault Tolerant and Virtual IP Addressing (VIPA).

The following sections are included.

- IOGEN Information—Describes the configuration statements to define your Cisco 7000/7500 series router to MVS
- Configuring the Interface—Describes the configuration changes to define your Cisco 7000/7500 series router to work with Unicenter TCPaccess
- Configuring the Router—Describes the configuration changes to define your Cisco 7000/7500 series router to work with Unicenter TCPaccess
- Fault Tolerant Considerations—Describes the configuration changes to enable the GateD Fault Tolerant feature of Unicenter TCPaccess
- RIP/OSPF Changes for the CIP Router—Describes the configuration changes to make your Cisco 7000/7500 series router work in the GateD Fault Tolerant environment with Unicenter TCPaccess
- GateD Fault Tolerant With VIPA—Describes the configuration changes to enable the VIPA feature to work with GateD
- Example—Provides examples of some of the configurations described in this document

Communicating with Cisco Routers

Unicenter TCPaccess communicates with Cisco 7000 and 7500 series routers using the Common Link Access to Workstation (CLAW) channel protocol through either an Enterprise Systems Connection (ESCON) or parallel channel connected to a CIP card that is installed in the router.

Once a physical connection is established, MVS must be configured to recognize the CIP and to define a series of subchannels on which the CLAW protocol is used to communicate between Unicenter TCPaccess and the Cisco 7000/7500 series router. Adding statements to your MVS input/output configuration program (IOCP) is described in IOGEN Information.

In addition to MVS configuration, both Unicenter TCPaccess and the Cisco 7000/7500 router must also be configured. The configuration for Unicenter TCPaccess is performed by adding statements to your TCP configuration member as described in the Configuring the Interface. This configuration defines the association between MVS and Unicenter TCPaccess, giving it access to the physical device.

The configuration for the Cisco 7000/7500 router is performed by issuing configuration commands to the router as described in the Configuring the Router. These commands provide the final configuration necessary to complete the communications path between Unicenter TCPaccess and the Cisco 7000/7500 router.

This appendix includes information for the following configurations:

- Single Channel Interface Processor (CIP) interface
- Multiple CIP Interfaces
- Multiple CIP interfaces with GateD Fault Tolerant
- Multiple CIP interfaces with GateD Fault Tolerant and VIPA

The steps provided in this appendix assume the following:

- Cisco 7000 or 7500 series routers are installed and integrated into the current network
- CIP cards are already installed, but the MVS input/output generation (IOGEN) has not been completed
- Unicenter TCPaccess is installed and set up to work in loopback

Additional References

For additional information, read the following documents:

- *Unicenter TCPaccess Communications Server Customization Guide*
- *Bridging and IBM Networking Configuration Guide*
- *Bridging and IBM Networking Command Reference*
- *Network Protocols Command Reference, Part 1*
- *Network Protocols Configuration Guide, Part 1*

IOGEN Information

This section describes the configuration statements necessary to define Cisco 7000 and 7500 series routers to MVS.

Before you install Unicenter TCPaccess, you must define the hardware interfaces needed to support the product. The I/O and IOCP generations define the devices used by Unicenter TCPaccess to access the network.

Note: Unicenter TCPaccess communicates with a CIP using the CLAW channel protocol. Each CLAW connection requires two adjacent subchannel addresses.

Parallel Channel-Attached Routers

This section describes the MVS definition instructions for Cisco 7000 and 7500 routers that are connected to the mainframe via parallel channel CIPs. It defines all parallel channel interfaces for Unicenter TCPaccess running the CLAW protocol.

MVSCP for Parallel Channel CIPs

Use the following statement to define the parallel channel-attached Cisco 7000 and 7500 series router for system generation for the Multiple Virtual System Configuration Program (MVSCP):

```
IODEVICE UNIT=CTC,ADDRESS=(aaa, n)
```

aaa	Base address.
n	Number of subchannel addresses to be defined.

Refer to the appropriate IBM system generation manual for more information.

IOCP for Parallel Channel CIPs

Use the following statements to define the parallel channel-attached Cisco 7000 and 7500 series router for IOCP generation:

```
CHPID     PATH=(pp),TYPE=CTC

CNTLUNIT CUNUMBER=number,PATH=(pp),UNIT=3088,
          UNITADD=(aa,n),PROTOCOL=pr

IODEVICE CUNUMBER=number,ADDRESS=(aaa,2)
          UNIT=CTC,UNITADD=aa
```

pp	Channel path address.
number	Unit number.
aaa or aa	Base unit address.
n	Number of subchannel addresses to be defined.
pr	One of these values: S 3.0 MB data streaming mode S4 4.5 MB data streaming mode

ESCON-Attached Routers

This section describes the MVS definition instructions for Cisco 7000 and 7500 series routers attached to the mainframe through ESCON, but not under the control of an Enterprise System Connection Director (ESCD) service. If you are defining the device under the control of an ESCD, refer to the appropriate IBM documentation.

MVSCP for ESCON CIPs

Use the following statement to define the ESCON-attached Cisco 7000 and 7500 series router for MVS system generation:

```
IODEVICE UNIT=SCTC,ADDRESS=(aaa,n)
```

aaa	Base address.
n	Number of subchannel addresses to be defined. Refer to the appropriate IBM system generation manual for more information.

IOCP for ESCON CIPs

Use the following statements to define the ESCON channel-attached Cisco 7000 and 7500 series router for IOCP generation:

```
CHPID    PATH=(pp),TYPE=CNC

CNTLUNIT    CUNUMBER=number,PATH=(pp),UNIT=SCTC,
    UNITADD=(aa,n),CUADD=c
IODEVICE    CUNUMBER=number,ADDRESS=(aaa,2),
    UNIT=SCTC,UNITADD=aa
```

pp	Channel path address.
number	Unit number.
aaa or aa	Base unit address.
n	Number of subchannel addresses to be defined. Refer to the appropriate IBM system generation manual for more information.
c	Control Unit Address; Can be 0 - F.

Configuring the Interface

This section describes the configuration changes you must make to define Cisco 7000 and 7500 series routers to work with Unicenter TCPaccess.

Defining the CIP Interface

You must make following changes to the TCPCFG*xx* member and the Cisco 7000 and 7500 series routers to define a CIP interface to Unicenter TCPaccess.

Note: TCPCFG*xx* is the main TCP/IP configuration member for Unicenter TCPaccess that controls communication with the network.

The TCPCFG*xx* member statements are described in the *Customization Guide*.

The MEDIA, NETWORK, and CLAW statements control access to the MVS host. Minimal updates are required in order to establish communication.

These are the steps necessary to configure Unicenter TCPaccess:

1. Define a MEDIA statement in the TCPCFG*xx* member. See the MEDIA Statement.
2. Define a NETWORK statement in the TCPCFG*xx* member. See the NETWORK Statement.
3. Define a CLAW statement in the TCPCFG*xx* member. See CLAW Statement.
4. Add the CLAW command at the Cisco router configuration prompt for the appropriate CIP. See the CLAW Command.

MEDIA Statement

The MEDIA statement defines a physical medium for Unicenter TCPaccess. You must define a CLAW network medium for Unicenter TCPaccess to communicate to a channel interface on a Cisco 7000 or Cisco 7500 series router. You must also specify a name to associate with this media.

If you intend to use the TCP Assist feature, you must also code ASSIST on the MEDIA statement. The default is NOASSIST.

The MEDIA statement is described in the "Network Configuration" chapter of the *Customization Guide*.

NETWORK Statement

The NETWORK statement defines the IP addressing of the media.

You must define the host IP address on the NETWORK statement. This IP address corresponds to the CLAW definition for the channel interface in the Cisco 7000 or Cisco 7500 series router.

Note: The NETWORK statement must follow the MEDIA statement or the MEDIANAME keyword must be coded to reference the desired MEDIA statement.

The DEST parameter specifies the remote IP address for point-to-point links. SUBNET specifies the subnet mask in standard dot notation (for example, 255.255.255.0).

```
NETWORK IPADDRESS ( ip_address )
    DEST ( destination )
    [ MEDIANAME ( media_name ) ]
    [ METRIC ( metric ) ]
    [ NETMASK ( net_mask ) ]
    [ SUBNETMASK ( subnet_mask ) ]
```

The NETWORK statement is described in the "Network Configuration" chapter of the *Customization Guide*.

CLAW Statement

The CLAW statement specifies configuration parameters for an interface running the CLAW protocol. You must code a CLAW statement with the starting device subchannel address for the Cisco 7000 or 7500 series router in the DEVADDR parameter of this statement. This starting subchannel address corresponds to the CLAW definition for the channel interface of the Cisco router. Additionally, the HOSTNAME and WSNAME parameters on this statement must reflect the host name and workstation name set by the CLAW definition in the router.

If you intend to use the CLAW Packing feature, you must also code PACKED on the CLAW statement. The default is UNPACKED. The CLAW definition for the channel interface in the Cisco router must also be set up for the PACKED feature (see Configuring the Router).

Note: Verify that your router supports and is set up for CLAW packing. If you code PACKED and your router does not support, or is not set up correctly for, this feature, the following message displays:

```
T01LL194E Device dev_name : packetizing feature is not supported.
```

This indicates that the CLAW link is unable to connect; it does not specify the reason for the failure. If your router does not support CLAW packing, do not code PACKED on the CLAW statement.

If the router is set up for a PACKED channel and you do not configure Unicenter TCPaccess for CLAW packing, this message displays:

```
T01LL198# Device dev_name : the CLAW statement does not reflect the controller
configuration PACKED option
```

This message also indicates that the CLAW link is unable to connect. If your router is set up for the CLAW packing feature, then you must code PACKED on the CLAW statement.

Note: The CLAW statement must follow the MEDIA statement or the MEDIANAME keyword must be coded to reference the desired MEDIA statement.

```
CLAW DEVADDR ( ccuu )
    [ BUFSIZE ( 1024 | 2048 | 3072 | 4096 | 8192 | 12288 | 16384 | 20480 | 24576 |
28672 |
        32768 | 36864 | 40960 | 45056 | 49152 | 53248 | 57334 | 61440 | 65535) ]
    [ CHARSET ( charset ) ]
    [ HOSTNAME ( hostname ) ]
    [ IBUF ( bufsize each ) ]
    [ MEDIANAME ( name ) ]
    [ OBUF ( outputbuffercount ) ]
    [ PACKED | UNPACKED ]
    [ RESTART ( restarttime ) ]
    [ SINGLENOOP | DOUBLENOOP ]
    [ START | NOSTART | AUTOSTART | NOAUTOSTART ]
    [ WSNAME ( workstation_name ) ]
```

The CLAW statement is described in the "Network Configuration" chapter of the *Customization Guide*.

Defining Multiple CIP Interfaces

If you have more than one CIP interface, you must define the following statements in the TCPCFG*xx* member for each interface and add the corresponding CLAW command as described in the CLAW Command section:

- MEDIA Statement
- NETWORK Statement
- CLAW Statement

Although there is no relationship between NETWORK and driver statements (CLAW statements), you must consider the organization of these statements under the MEDIA statement(s) for point-to-point type links such as CLAW. Cisco Claw devices have a one-to-one correspondence between the host IP address and a subchannel address pair. Once you establish a MEDIA and NETWORK statement pair, the CLAW statement of the device corresponding to the NETWORK IP address in the Cisco router must follow immediately or the MEDIANAME keyword must be coded. For ease of organization and readability, it is recommended that you code the MEDIA, NETWORK, and CLAW statements, in that order.

Note: If you use fault tolerant with your Cisco routers, you must define at least two MEDIA, NETWORK, and CLAW statements that correspond to separate CIP interfaces.

Configuring the Router

This section describes the configuration changes necessary to make your Cisco 7000 and 7500 series routers to work with Unicenter TCPaccess.

Configuring the CIP Interface

The following sections describe the configuration commands necessary to complete the communications between Unicenter TCPaccess and the Cisco 7000/7500 router. See your Cisco IOS Configuration documentation for more details on how to configure your Cisco 7000/7500 router.

In addition to CLAW configuration commands specific to a particular CLAW connection, each CIP must also receive some basic configuration. This includes a subnet and IP address for each CIP interface, as well as other basic interface configuration commands that are desirable or required for the CIP to operate correctly.

CIP interface configuration commands are to be entered at the interface configuration prompt for the corresponding physical channel connection.

For more details on configuring the CIP, refer to *Bridging and IBM Networking Configuration Guide* for your version of Unicenter TCPaccess.

IP ADDRESS Command

Assigns an IP address and associated subnet mask for a given channel interface. The IP address chosen should be within the same subnet as defined by the NETWORK statement in the TCPCFG*xx* member. Additionally, the mask specified on the IP ADDRESS command must match that specified in the NETWORK statement.

```
ip address address mask
```

For more information on the IP ADDRESS command, see the "IP Commands" chapter of the *Network Protocols Command Reference, Part 1*.

CLAW Command

Specifies the configuration Unicenter TCPaccess needs to communicate with the CIP. It provides configuration information necessary to communicate through MVS and to Unicenter TCPaccess.

The parameters associated with communicating through MVS include the channel path and device address that correspond to the CTLUNIT and IODEVICE statements (described in IOGEN Information) in your IOCP.

```
claw path dev_addr ip_addr host_name dev_name host_app ws_app
[ broadcast ]
```

path This hexadecimal value specifies the data path and consists of two digits for the physical connection, one digit for the control unit logical address, and one digit for the channel logical address.

For configuring the CLAW statement in Unicenter TCPaccess, use the form *XXHC*, where:

- *XX* For ESCON point-to-point or Bus and Tag, this is 01; for switch point-to-point configuration, it is the ESCD port number that the host channel is plugged into.
- *H* For ESCON, this specifies the host partition number if channel path identifier (CHPID) is shared; for Bus and Tag, or non-shared ESCON CHPIDs, it is zero.
- *C* For ESCON-defined control units, this matches the CUADD parameter of the CNTLUNIT macro in the IOCP.

dev_addr Unit address associated with the control unit number and path as specified in the host IOCP file.

In this case, it corresponds to the UNITADD field of the CNTLUNIT macro of the IOCP.

ip_addr IP address in the HOME statement of the host TCP/IP application configuration file.

In this case, it must match the IP address specified in the corresponding TCPCFG*xx* member NETWORK statement

host_name Host name in the device statement in the host TCP/IP application configuration file.

In this case, it must match the HOSTNAME(*name*) parameter of the corresponding TCPCFG*xx* member CLAW statement

dev_name CLAW workstation name in the device statement in the host TCP/IP application configuration file. In this case, it must match the WSNAME(*name*) parameter of the corresponding TCPCFG*xx* member CLAW statement.

This is an optional parameter and should be coded only if broadcast traffic is to be sent to the mainframe (in other words, the gateway dæ mon (GateD) is running on the host).

host_app Name of the host application for the CLAW IP link.

It will be one of the following:

- TCPIP For normal channel mode
- PACKED For PACKED channel mode

ws_app Name of the workstation application for the CLAW IP link. It will be one of the following:

- TCPIP For normal channel mode
- PACKED For PACKED channel mode

broadcast Enable broadcast processing for the subchannel.

For more information on the CLAW command, see *Bridging and IBM Networking Command Reference*.

channel-protocol Command

Defines the data rate for a parallel channel interface.

The value for this command corresponds to the PROTOCOL=*pr* parameter on the CNTLUNIT statement in the IOCP input deck (see IOCP for Parallel Channel CIPs).

Note: For ESCON-attached CIPs, you need not add the channel-protocol command.

```
channel-protocol [s | s4]
```

Other Suggested Commands

The following commands are suggested. For more details on their purpose refer to *Network Protocols Command Reference, Part 1*. Additional information can also be found in *Bridging and IBM Networking Configuration Guide*.

```
ip route-cache same-interface
no ip redirects
```

Fault Tolerant Considerations

This section describes the configuration changes you must make to enable the GateD Fault Tolerant feature of Unicenter TCPaccess.

The GateD routing protocol lets Unicenter TCPaccess perform some of the functions of a router in a multi-homed environment. Specify configuration parameters for GateD in the GTDCFG*xx* member.

To use Fault Tolerant GateD you must have at least two CIP interfaces and you must have an equal number of NETWORK and CLAW statements defined in the TCPCFG*xx* member. If you did not define a MEDIA statement for LOOPBACK, then you should also have the same number of MEDIA statements as NETWORK and CLAW statements.

Use the following steps to configure Unicenter TCPaccess for GateD Fault Tolerant:

1. Create a GTDCFG*xx* member.
2. Activate GateD by specifying GATED on the IP statement in member TCPCFG*xx*. See GateD Interface.
3. Configure the routing protocol on the Cisco 7000 or 7500 router. See RIP/OSPF Changes for the CIP Router.
4. Remove the DEST parameter on all of the NETWORK statements.

Note: Leaving the DEST parameter coded on the NETWORK statement while using GateD can cause unpredictable results.

These statements and members are described in the "Network Configuration" chapter of the *Customization Guide*.

To use fault tolerant facilities with Unicenter TCPaccess, you must have at least two CIP interfaces. Within this configuration both the CIP and the MVS host could run the recommended open shortest path first (OSPF) or the Routing Information Protocol (RIP) routing protocol. If correctly configured, this ensures timely recovery in the event of a network outage.

GateD Interface

The IP statement controls the operation of the Internet layer. To activate GateD, you must define an IP statement with the GATED parameter specifying the GTDCFG*xx* member. You must also code FORWARD to allow hosts on one local interface to forward to hosts on another local interface.

IP Statement

```
IP [ FORWARD | NOFORWARD ]
     [ GATED ( gated_config ) | NOGATED ]
     [ REASSEMBLYTIMEOUT ( timeout ) ]
     [ TIMETOLIVE ( number ) ]
     [ TYPEOFSERVICE ( number ) ]
     [ MAXVIPA( number )]
```

Configuring GateD

Use the GTDCFG*xx* member to specify configuration parameters for the GateD routing protocol.

The GateD configuration member is very different from most other configuration files for Unicenter TCPaccess. It consists of a sequence of statements, each terminated by a semi-colon (;). Statements are composed of tokens separated by white space (any combination of blanks, tabs, and new lines). Comments use the C style comment, which begins with a "/*" and ends with "*/".

GateD relies heavily on functions and features native to UNIX operating systems. These functions and features are emulated on MVS by Unicenter TCPaccess or by the SAS/C runtime library.

Note: References to the UNIX kernel in this section refer to Unicenter TCPaccess and not the MVS operating system.

For information about various function routines generally native to UNIX, refer to the following:

- *Unicenter TCPaccess Communications Server C/Socket Programmer' Reference*
- *Unicenter TCPaccess Communications Server RPC/XDR Programmer's Reference*
- *SAS/C Library Reference*, Second Edition, Volumes 1 and 2
- The man pages on UNIX. man pages are UNIX system dependent and may not be the same under all UNIX implementations.

For information about initial configuration, read the "Network Customization" chapter in the *Customization Guide*.

Traceoptions Statement

Controls tracing options for GateD.

```
traceoptions traceoption [ traceoption [ ...] ]
    [ except traceoption [ traceoption [... ] ] ];
```

- If none is the only option specified, tracing is turned off.
- If except is specified, flags listed before it are turned on and flags listed after it are turned off. Use this to turn on all but a few flags.

Interfaces Statement

Lets you specify interface options in the GTDCFG*xx* member.

```
interfaces {
    options [ strictifs ] [ scaninterval time ] ;
    interface interface_list interface_options ;
    define address [ broadcast broadaddr | pointopoint lcladdr ]
        [ netmask netmask ] [multicast];
    } ;
```

Options Statement

Allows specification of global GateD options.

```
options [ option_list ];
```

OSPF Protocol Configuration

The ospf statement lets you configure GateD for the OSPF protocol. If the OSPF clause is not specified, the default is ospf off.

OSPF is the recommended protocol.

```
ospf yes | no | on | off
[ {
    [ defaults
        {
            preference preference ;
            cost cost ;
            tag [ tag | as [ as_tag ] ] ;
            type 1|2 ;
        } ] ;
    [ exportlimit routes ; ]
    [ exportinterval time ; ]
    [ traceoptions traceoptions ; ]
    [ monitorauthkey authkey ; ]
    [ area area | backbone
        {
            authtype 0 | 1 | none | simple ;
            stub [ cost cost ];
            networks { network [ mask mask ] ; } ;
            stubhosts { host cost cost ; } ;
            interface interface [ cost cost ]
            {
                [ enable | disable ] ;
                retransmitinterval time ;
                transitdelay time ;
                priority priority ;
                hellointerval time ;
                routerdeadinterval time ;
                authkey auth_key ;
```

```
            } ;
            interface interface nonbroadcast [ cost cost ]
            {
                pollinterval time ;
                routers { gateway [ eligible ] ... } ;
                [ enable | disable ] ;
                retransmitinterval time ;
                transitdelay time ;
                priority priority ;
                hellointerval time ;
                routerdeadinterval time ;
                authkey auth_key ;
             } ;
        } ; ]
    virtuallink neighborid routerid transitarea area
    {
        [ enable | disable ] ;
        retransmitinterval time ;
        transitdelay time ;
        priority priority ;
        hellointerval time ;
        routerdeadinterval time ;
        authkey auth_key ;
    } ;   . . .
} ] ;
```

RIP Protocol

This is an example of how to configure GateD for RIP (Routing Information Protocol). If the RIP clause is not specified, the default is rip on.

```
rip yes | no | on | off [
{
    broadcast;
    nobroadcast;
    nocheckzero;
    preference preference;
    defaultmetric metric;
    interface interface_list [noripin] [noripout]
        [metricin metric] [metricout metric]
        [version 1]|[version 2 broadcast]
        [authentication [none|password]];
...
    trustedgateways gateway_list;
    sourcegateways gateway_list;
    traceoptions traceoptions;
} ] ;
```

RIP/OSPF Changes for the CIP Router

This section describes the configuration changes to make your Cisco 7000/7500 series router work in the GateD Fault Tolerant environment with Unicenter TCPaccess. One change that is independent of whether you plan to run OSPF or RIP on the mainframe is that the broadcast parameter must be specified on the CLAW statement defining each host link used for propagating routing updates. All other parameters are specific to the routing protocol and are described in this section.

Configuring for OSPF

This section gives a brief description of configuring OSPF for your Cisco 7000/7500 series router. For more detail, or to configure your router for OSPF beyond what is needed for Fault Tolerance CIP support, read the *Network Protocols Configuration Guide*, from Cisco, for your version of IOS.

router ospf Command

The router ospf command enables OSPF routing on your router. More specific OSPF commands follow.

```
router ospf process_id
```

This router ospf command will be followed by more specific commands describing your OSPF network. These include the network area command, the passive-interface command, and several others, depending on the specifics of your network.

network area Command

Defines which interfaces are to participate in OSPF routing and the OSPF area that they are in.

```
network address wildcard_mask area area_id
```

passive-interface Command

Suppresses the sending of hello packets through the specified interface. This is often used when running multiple routing protocols on the same router to keep the hello packets from being sent across the wrong interfaces.

```
passive-interface type number
```

If you plan to run multiple routing protocols in the same router, you should also consider using the DEFAULT-INFORMATION ORIGINATE and REDISTRIBUTE routing commands. These are described in the *Network Protocols Configuration Guide*, from Cisco, for your version of IOS. OSPF commands also need to be added to the CIP interfaces of your Cisco 7000/7500 routers.

ip ospf network Command

Configures the OSPF network type for this interface. For the CIP, this should always be non-broadcast.

```
ip ospf network non-broadcast
```

ip ospf hello-interval Command

Specifies the length of time, in seconds, between the OSPF hello packets sent out of this interface.

```
ip ospf hello-interval seconds
```

ip ospf dead-interval Command

Specifies the length of time, in seconds, that the hello packets for a device must not have been seen before its neighbors declare the router down.

```
ip ospf dead-interval seconds
```

Configuring for RIP

This section gives a brief description of configuring RIP for your Cisco 7000/7500 series router.

router rip Command

Enables RIP routing on your router. Specific RIP commands follow.

```
router rip
```

network Command

The network command associates a network with a RIP routing process.

```
network network_number
```

version Command

Tells the router the default version of RIP packets to send and receive from the router. This can be overridden on the interface.

```
version 1/2
```

On each interface supporting RIP, you can tell the router the version of RIP supported. This allows multiple levels to be run on the router at the same time.

ip rip send version Command

Configures an interface to send either RIP1, RIP2, or both types of update packets.

```
ip rip send version 1/2/12
```

ip rip receive version Command

Configures an interface to receive either RIP1, RIP2, or both types of update packets.

```
ip rip send version 1/2/12
```

As with OSPF, if you are planning to run multiple routing protocols in the router, consider the redistribute and passive-interface routing commands. For more information about these commands, refer to the *Network Protocols Configuration Guide,* from Cisco, for your version of Unicenter TCPaccess.

GateD Fault Tolerant with VIPA

This section describes the configuration changes you must make to enable the VIPA feature to work with GateD.

In addition to the previous configuration, installations can make use of the VIPA facility to provide complete and constant access to their host data. This configuration does require that one or more separate subnets be defined to the CIP and in the TCPCFG*xx* member. Since these subnets are virtual, if a route to a real interface goes down, you do not lose access to the host when using the virtual addressing described here.

At this point, you should have a fully configured multihomed Unicenter TCPaccess system with GateD Fault Tolerant.

These are the steps necessary to configure Unicenter TCPaccess for GateD Fault Tolerant:

- Define a MEDIA Statement and a NETWORK Statement in the TCPCFG*xx* member for the static VIPA.

  The VIPA must be on an unused subnet and it must follow the MEDIA statement for the VIRTUAL IP address.

- Define one or more VIPANET statements in the TCPCFG*xx* member for the application dynamic VIPA subnets.

  See the Defining Application Dynamic VIPA Subnets—VIPANET Statement section of the *Customization Guide*.

- Add the vr0 and dv interface in the GateD configuration member.

  In the GTDCFG*xx* member that you are using for GateD Fault Tolerant, code the following:

```
interfaces
{
    interface vr0 passive;
    interface dv passive;
};
```

  When using the recommended OSPF protocol, you should also add vr0 and dv interface definitions to the OSPF backbone.

```
interface vr0 nonbroadcast
interface dv nonbroadcast
```

  When using the RIP protocol, code the vr0 interface and the dv interface with the noripout option.

```
rip yes
{
    interface vr0 noripout ;
    interface dv noripout ;
} ;
```

Note: Always remove the DEST parameter on all TCPCFG*xx* NETWORK statements or unpredictable results may occur.

Example

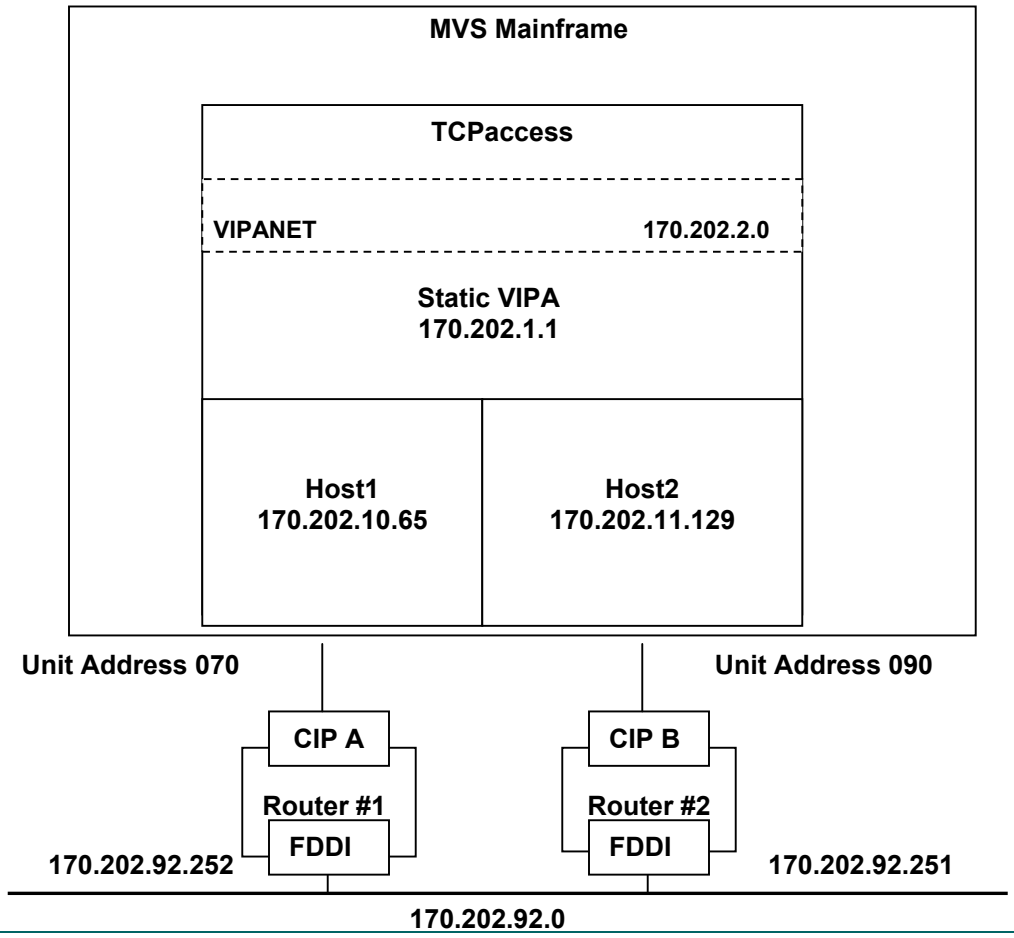
This example shows the features of VIPA and GateD with OSPF, providing a fault tolerant environment. Two Cisco 7000 series routers with CIPs provide the network-to-host connections.

For this example, the following are true:

- The CIP cards are ESCON-attached; they are not under the control of an ESCD service
- Addressing is set to Class B
- The 070 channel address range is defined on CHPID '10'
- The 090 channel address range is defined on CHPID '20'

System Diagram

The following diagram shows the TCP Fault Tolerant example with VIPA and GateD running OSPF using two CISCO 7000 series routers with CIPs described in this section.



VIPANET

```
VIPANET IPADDRESS(170.202.2.0)
    SUBNETMASK(255.255.255.0)
```

VIPA

```
MEDIA VIRTUAL
    MTU(4096)
    NAME(VIRTUAL)
NETWORK IPPADDRESS(170.202.1.1)
    SUBNET(255.255.255.0)
```

CIP A

```
MEDIA CLAW
    NAME(CLAW1)
    MTU(4096)
    MSSDEF(4096)
    MSSOPT(ALWAYS)
    ASSIST
NETWORK IPADDRESS(170.202.10.65)
    SUBNET(255.255.255.0)
    MEDIANAME(CLAW1)
CLAW DEVADDR(070)
    BUFSIZE(4096)
    IBUF(26)
    OBUF(26)
    HOSTNAME(HOSTTCPA)
    WSNAME(CIPTCPA)
    PACKED
    MEDIANAME(CLAW1)
```

CIP B

```
MEDIA CLAW
    NAME(CLAW2)
    MTU(4096)
    MSSDEF(4096)
    MSSOPT(ALWAYS)
    ASSIST
NETWORK IPADDRESS(170.202.11.129)
    SUBNET(255.255.255.0)
    MEDIANAME(CLAW2)
CLAW DEVADDR(090)
    BUFSIZE(4096)
    8BUF(26)
    OBUF(26)
    HOSTNAME(HOSTTCPB)
    WSNAME(CIPTCPB)
    PACKED
    MEDIANAME(CLAW2)
```

GTDCFGxx Configuration Example

This is an example of GTDCFG*xx* configuration shown in the previous system diagram:

```
traceoptions general mark protocol update;
options noresolv;
interfaces
    {
        interface vr0 passive;
        interface dv passive;
    };
routerid 170.202.1.1;
ospf yes
{
    backbone
    {
        networks
        {
            170.202.1.0 mask 255.255.255.0 ;
            170.202.2.0 mask 255.255.255.0 ;
            170.202.10.0 mask 255.255.255.0 ;
            170.202.11.0 mask 255.255.255.0 ;
            170.202.0.0 mask 255.255.0.0 ;
        };
        authtype 0;
        interface vr0 nonbroadcast cost 1
        {
            retransmitinterval 5;
            hellointerval 6;
            pollinterval 6;
            routerdeadinterval 24;
        };
        interface dv nonbroadcast cost 1
        {
            retransmitinterval 5;
            hellointerval 6;
            pollinterval 6;
            routerdeadinterval 24;
        };

        interface cl0 nonbroadcast cost 1
        {
            retransmitinterval 5;
            hellointerval 6;
            pollinterval 6;
            routerdeadinterval 24;
            routers {170.202.10.126 eligible ; };
        };
        interface cl1 nonbroadcast cost 1
        {
            retransmitinterval 5;
            hellointerval 6;
            pollinterval 6;
            routerdeadinterval 24;
            routers {170.202.11.190 eligible ; };
        };
    };
};
```

Cisco CIP Configuration

These are examples of the CIP configuration.

CIP A Configuration

```
CIPA
!
version 11.2
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname COMPANY-CIPA
!
boot system flash slot0:rsp-k-mz_260-4.bin
enable password 7 1315991059EA062B25
!
interface Fddi2/0
    ip address 170.202.92.252 255.255.255.0
    no ip redirects
    no ip mroute-cache
!
interface Channel3/0
    ip address 170.202.10.126 255.255.255.0
    no ip redirects
    ip ospf network non-broadcast
    ip ospf hello-interval 6
    no keepalive
```

Connectivity to Router

```
    claw 0100 70 170.202.10.65 HOSTTCPA CIPTCPA PACKED PACKED broadcast
    !
```

Set up OSPF Globally

```
router ospf 1
    network 170.202.0.0 0.0.255.255 area 0
    network 190.202.0.0
    default-metric 56 2000 255 255 4096
    neighbor 170.202.10.65 priority 1 poll-interval 6
!
ip domain-name company.com
!
line con 0
    login
line aux 0
    login
    transport input all
line vty 0 4
    login
!
end
```

CIP B Configuration

```
CIPB
!
! No configuration change since last restart
!
version 11.2
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname COMPANY-CIPB
!
boot system flash slot0:rsp-k-mz-260-4.bin
enable password 7 131549101AE1972B25
!
interface Fddi0/0
    ip address 170.202.92.251 255.255.255.0
    no ip redirects
!
interface Channel1/0
    ip address 170.202.11.190 255.255.255.0
    no ip redirects
    ip ospf network non-broadcast
    ip ospf hello-interval 6
     no keepalive
```

Connectivity to Router

```
claw 0100 90 170.202.11.129 HOSTTCPB CIPTCPB PACKED PACKED broadcast
!
!
```

Set Up OSPF Globally

```
router ospf 1
    network 170.202.0.0 0.0.255.255 area 0
    network 170.202.0.0
     default-metric 56 2000 255 255 4096
    neighbor 170.202.11.129 priority 1 poll-interval 6
!
ip domain-name company.com
!
line con 0
    login
line aux 0
    login
    transport input all
line vty 0 4
    login
!
end
```

Appendix

C

Using a CDLC Driver

Unicenter TCPaccess customers can use a 3745 front-end processor (FEP) to attach channel interfaces directly to an IP network. The channel interface in this situation is configured in the Network Control Program (NCP) as a native IP element. The IP channel interface does not fall under control of VTAM.

Unicenter TCPaccess uses the IBM CDLC protocol to implement support for the IP channel attachment capability. The 3745 FEP appears as a single device, operating in half-duplex mode. The CDLC protocol uses a single subchannel in half-duplex mode.

IP datagrams are passed from an MVS-based IP application over bus-and-tag or ESCON channels to a 3745 FEP. To Unicenter TCPaccess, the channel-attached NCP running native IP support looks like a channel-attached router.

The 3745 FEP cannot share the same bus-and-tag interface between SNA and IP traffic to the mainframe. The native IP attachment requires a dedicated 3745 channel adapter. If you are using ESCON with the 3746-900 frame, IP and SNA traffic can flow over the same channel connection if you create a separate NCP link station for the IP traffic.

Related References

The following IBM references provide additional information about the 3745 IP-Over-Channel feature:

- G023601 *WSC Flash*
- Item FN00276 in IBMLINK
- SG24-2592 *NCP Version 7 Release 3 New Functions*
- http://www.raleigh.ibm.com/tcm/tcmprod.html

System and Maintenance-Level Requirements

Verify that your hardware and software are at the current levels and that all necessary maintenance has been applied.

This CDLC driver support assumes that you have either an Ethernet or a Token-Ring adapter installed in your FEP. The following table lists the IBM software and hardware requirements. If you do not meet these levels you must check with IBM.

Software / Hardware Component	Minimum Release Level	Additional Requirements, including APARs and PTFs
ACF/NCP	Version 7 Release 3	IR30290, IR30569, IR30374, IR30430
ACF/SSP	Version 4 Release 3	IR30035, IR30113, IR30588, IR30729, IR30732
ACF/VTAM	Version 3 Release 3	OY64723
3745		Dedicated type 6 or type 7 channel adapter for the IP traffic
3746-900		Microcode EC D22510K (ECA 142)
3746-950		The 3746-950 has its own control program. Configuration is done through a GUI, not through an NCP system generation.

Callable System Services Library

Users must have the Callable System Services library, SYS1.CSSLIB, available for the installation. Modules from this library are linked with Unicenter TCPaccess for CDLC driver support. This library is distributed with MVS/ESA.

Note: If you do not have this library available, CDLC driver support will not function properly.

Index

3

A

B

C

D

E

F

G

H

I

L

M

N

O

P

R

S

T

U

V

V