VAX P.S.I.
(Packetnet System Interface)
Management Guide

Order No. AA-HS82B-TE

SUPERSESSION/UPDATE INFORMATION: This is a revised manual

OPERATING SYSTEM AND VERSION: VAX/VMS V5.0

SOFTWARE VERSION: VAX PSI V4.2

digital™

First Edition, May 1986
Updated August 1987
Revised June 1988

The READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

This manual was produced by the Wide Area Communications Environment group in Reading, England.

Contents

2 Configuring VAX PSI

3 Testing the Configuration of VAX PSI

Part II: PSI Security

4 Introduction to PSI Security

5 How PSI Security Works

6 How to Use PSI Security

B The VAX PSI Installation Checkout Procedure

C Flow Control Parameter Negotiation in VAX PSI

Index

Figures

Tables

How to Use This Manual

Purpose of This Manual

This manual explains how to use VAX PSI to communicate with remote systems connected to a Packet Switching Data Network (PSDN). This manual uses the term PSDN to refer to any public or private packet switching network that DIGITAL supports. For further details of supported networks, refer to the *Software Product Description* (SPD) for VAX PSI.

Intended Audience

This manual is intended for network managers who have the task of managing VAX PSI on a VMS system, to monitor and control outgoing access to PSDNs and also incoming access from PSDNs.

The manual assumes that network managers understand and have experience of the following:

- Data communications
- Packet Switching
- Local Area Networks
- Wide Area Networks
- VMS Operating Systems
- The System Management function on VAX and MicroVAX computers
- DECnet
- DIGITAL's PSI product

In addition, the manual assumes that network managers have read the *P.S.I. Introduction.*

Structure of This Manual

The manual is divided into four parts and covers the following topics:

- General Management - This part of the manual describes how to set up and configure VAX PSI on your system using the Network Control Program commands directly relating to VAX PSI. It also describes how to use the Configuration Test Program to test the installation and full configuration of VAX PSI on your system.
- Security - This part of the manual describes the PSI Security utility and shows how to control both access to your system from PSDNs and access to PSDNs by local users.
- Accounting - This part of the manual describes the PSI Accounting utility and shows how to produce new files or reports using the data in one or more previously recorded copies of the PSI accounting log file.
- Reference Information - This part of the manual contains information relating to the task of managing VAX PSI on your system.

Associated Manuals

The other manuals in the VAX PSI documentation set are:

P.S.I. Introduction - This manual introduces PSDNs, the X.25 and X.29 interfaces, and the VAX PSI software. READ THIS MANUAL BEFORE READING THE OTHER VAX PSI MANUALS.

VAX P.S.I. Installation Procedures - This manual explains how to install and test the installation of VAX PSI or VAX PSI Access.

VAX P.S.I. X.25 Programmer's Guide - This manual explains how to use VAX PSI to communicate with remote systems.

VAX P.S.I. X.29 Programmer's Guide - This manual introduces the X.29 facilities and the X.29 terminal driver.

VAX P.S.I. PAD and MAIL Utilities - This manual explains:

- How to use the VAX PSI PAD utility, which provides a VMS-based PAD as an alternative to the PSDN PAD.
- How to use VMS Mail through PSI.

VAX P.S.I. Problem Solving Guide - This manual describes a general procedure for solving problems and describes how to perform traces, loopback testing and device dumps.

Public Network Information manual - This manual contains information specific to each PSDN supported by VAX PSI.

The user is also assumed to have knowledge of the VMS manual set; in particular, the following manuals:

VMS Networking Manual and *VMS Network Control Program Manual* - These manuals describe DIGITAL networking and give an overview of how to configure, control, and monitor the VAX PSI product.

Guide to VMS System Security

Guide to Maintaining a VMS System

VMS Accounting Utility Manual

VMS Authorize Utility Manual

VMS DCL Dictionary

For information on the rest of the VMS manual set, see the *Overview of VMS Documentation.*

Manual Conventions

<*xxx*>	This one- to three-character symbol indicates that you press a key on the terminal; for example: <RET> indicates the RETURN key <DEL> indicates the DELETE key <ESC> indicates the ESCAPE key
<CTRL/*x*>	This symbol indicates that you press the CTRL key at the same time as you press another key; for example, <CTRL/C>, <CTRL/Y>, and so on.
Italics	indicate variable information.
Red print	indicates commands and data that you enter.

I

General Management Procedures

This part of the manual consists of three chapters:

- Chapter 1 - Summarizes the tasks that you have to perform as the VAX PSI system/network manager, and points to other parts of the manual for more details.
- Chapter 2 - Describes how you configure your system using the Network Control Program.
- Chapter 3 - Describes how you run the Configuration Test Program to test the configuration of VAX PSI on your system.

1

VAX PSI Management Tasks

As the manager of a VAX PSI system, you are responsible for the following tasks:

- Pre-installation preparation
- Adding destinations to the basic system produced during installation
- Setting up Data Link Mapping
- Placing security restrictions on the system
- Making use of the accounting facilities
- Managing the newer VAX PSI features such as a Combination system and ISO 8208 networks
- Migrating from the current system to a larger system, or modifying the current system

The following sections outline these tasks.

1.1 Pre-installation Preparation

The following subsections explain what information you need and what actions you must perform before starting to install the VAX PSI system.

1.1.1 Line Information

Decide which line you will use as the PSI line. You need to know:

- The Control Status Register (CSR) address
- The vector address
- The UNIBUS Adapter (UBA) number of the line interface

Obtain this information from the site management guide maintained by DIGITAL Field Service for your configuration.

1.1.2 Subscription to a Local PSDN

Make sure you subscribe to the appropriate PSDNs. You need to:

1. Contact the PSDN authorities to get a line to a DCE and an approved modem.
2. Make sure you have the correct cabling for your line interface.
3. Get permission from the PSDN authorities to connect your VAX PSI system to the PSDN. Ideally, you should subscribe to the local PSDN well before you expect to install VAX PSI, because the subscription procedure can take a few months.

The PSDN authorities allocate a DTE address to your system. The format of the address varies depending on which PSDN you subscribe to, but usually the DTE address is 12 digits long. (See the *Public Network Information* manual for details of the DTE address format for your PSDN.)

The PSDN authorities also allocate ranges of Logical Channel Numbers (LCNs) for use by the DTE. These are 12-bit decimal numbers that define the channel each virtual circuit uses. Network authorities usually group these channels by the type of virtual circuit. For example, PSS uses LCNs in the range 1 to 511 for PVCs, 512 to 1023 for incoming calls, and so on. (See the *Public Network Information* manual for details of the logical channel ranges for your PSDN.)

1.1.2.1 Optional Facilities

When you subscribe to a PSDN, you can decide whether or not to use a range of optional facilities. You must be aware of what facilities you have subscribed to, before you install VAX PSI. If you have agreed to use different default and maximum packet and window sizes from those set as defaults by the network authorities, you must specify these with Network Control Program (NCP) commands (see Appendix C).

Refer to the *VMS Networking Manual* and the *VMS Network Control Program Manual* for further information.

1.1.2.2 Remote DTE Addresses

If you plan to use Data Link Mapping (DLM), make sure you know the full DTE addresses of all the remote DTEs to which your system will connect. These must be entered in the correct format and you will usually need to include a subaddress. Refer to Section 2.5 for details.

1.1.3 Setting Up the Line

Before starting to set up the line to the DTE, make sure that the line hardware is ready for use. In particular, check that:

- The modem works. (Check this by asking the PSDN to run a remote test on the modem.)
- The cables between the modem and the line interface are correctly connected.
- The PSDN authorities have commissioned your port on the DCE.

1.2 Installing the VAX PSI System

Installation should be performed by a DIGITAL Software Specialist as described in the *VAX P.S.I. Installation Procedures.*

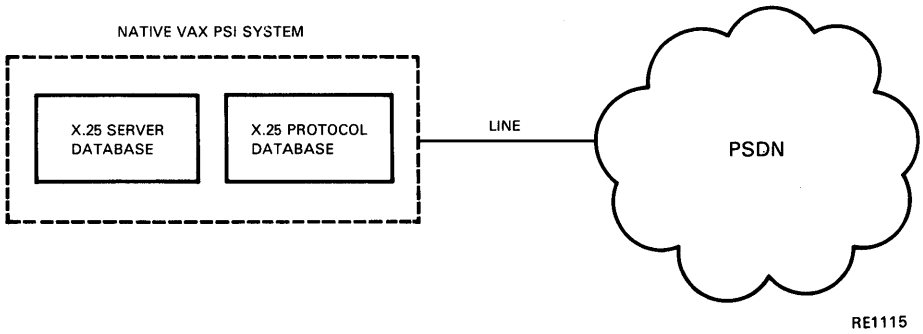
When VAX PSI is installed for the first time, a basic system of at least one DTE connected to at least one PSDN is produced. (Or if the system is VAX PSI Access, the connection is to a Multi-host system.)

1.3 Setting Up Destinations

When VAX PSI has been installed, you need to define destinations for incoming calls to VAX PSI. The type of destinations you set up varies according to whether you have a Native mode, Multi-host mode or Access mode system.

1.3.1 Setting Up Destinations in a Native Mode System

Figure 1–1 A Native Mode System



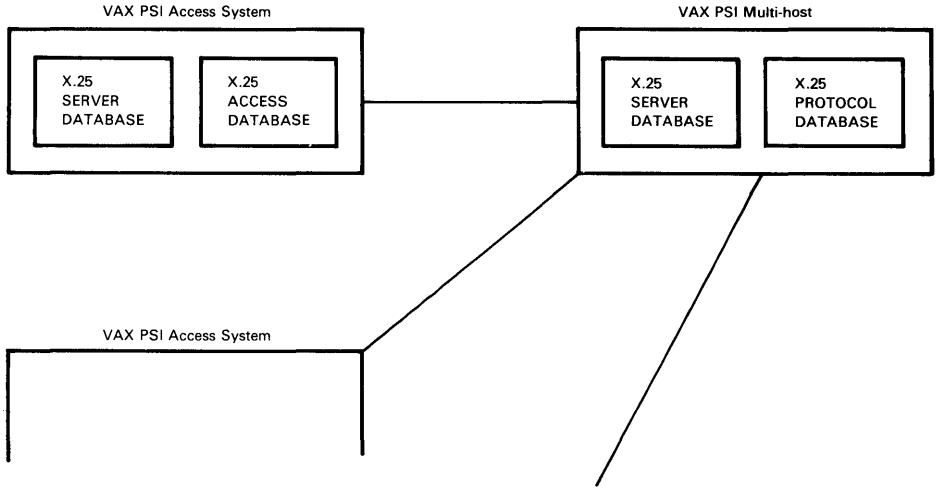
After a Native mode system has been installed you need to add destinations to the X25-SERVER database. Destinations are used to receive incoming calls to the VAX PSI software from remote DTEs. If a call matches a destination, then a process associated with the destination is activated.

There will usually be several destinations defined in the X25-SERVER database. Each destination has the job of matching with different types of call or calls from different DTEs.

Refer to Section 2.3.4 for details of Network Control Program (NCP) commands for setting up destinations.

1.3.2 Setting Up Destinations in a Multi-host Mode System

Figure 1–2 A Multi-host Mode System



When a Multi-host system has been installed, you need to add destinations to the X25-SERVER database:

1. For calls intended for the Multi-host system itself.
2. For calls intended for the Access systems that use your Multi-host system as a gateway.

Details are given in Section 2.3.

1.3.3 Setting up Destinations in an Access System

When an Access system has been installed, you need to add destinations to the X25-SERVER database. (You do this as you would on a Native system.)

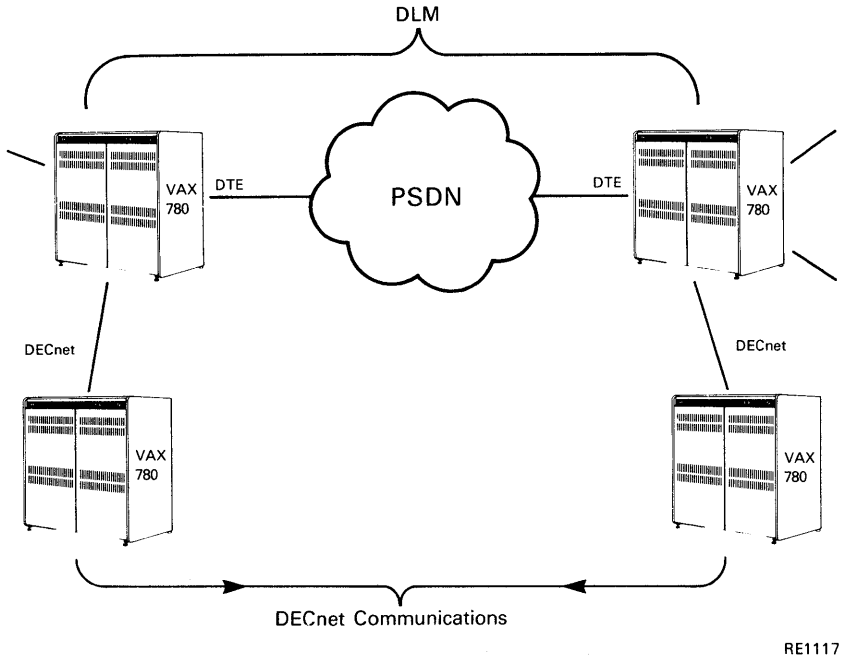
You also need to specify, in the X25-ACCESS database, which Connector system is going to be the gateway to a PSDN.

The Network Control Program commands for this are described in Chapter 2.

1.4 Setting Up DLM (Data Link Mapping)

DLM allows DECnet users to reach remote DECnet nodes across a PSDN. The PSDN part of the route is not visible to the DECnet users because VAX PSI manages the DLM circuit on their behalf. This means that when users want to send data to a remote DECnet node through the local PSDN, they can send the data as if the remote DECnet node were on their local DECnet network.

Figure 1–3 Data Link Mapping



The DECnet Routing module recognizes that the data has to go through a PSDN and selects a DLM circuit to carry the data. The DLM circuit passes the DECnet data to the VAX PSI software for conversion to X.25 packets. This is done by including X.25 protocol information with the DECnet data. VAX PSI then sends the packets to the remote DTE, which removes the X.25 protocol information before sending the DECnet data on to its destination.

1.4.1 Setting Up DLM Circuits

If your system is to provide DLM, you need to set up virtual circuits specifically for DLM. For any DLM circuit set up between two DTEs, one DTE makes the call by setting the outgoing circuit on. At this DTE the circuit is an outgoing DLM circuit. The DTE at the other end of the circuit accepts the call if the call's subaddress is within the range defined for DLM calls at that DTE, and if the circuit (here an incoming DLM circuit) is set to ON by the system manager (with an NCP command).

The NCP commands you use to set up DLM circuits are described in Section 2.5.

1.5 Using the Security System

When a basic VAX PSI system has been installed, you have the option of either using PSI Security or leaving the security system inactive.

1.5.1 Activating the Security System

You activate PSI Security when you use PSIAUTHORIZE to add data to any of the security databases, or when you run the command file PSI_SECURITY.COM.

NOTE

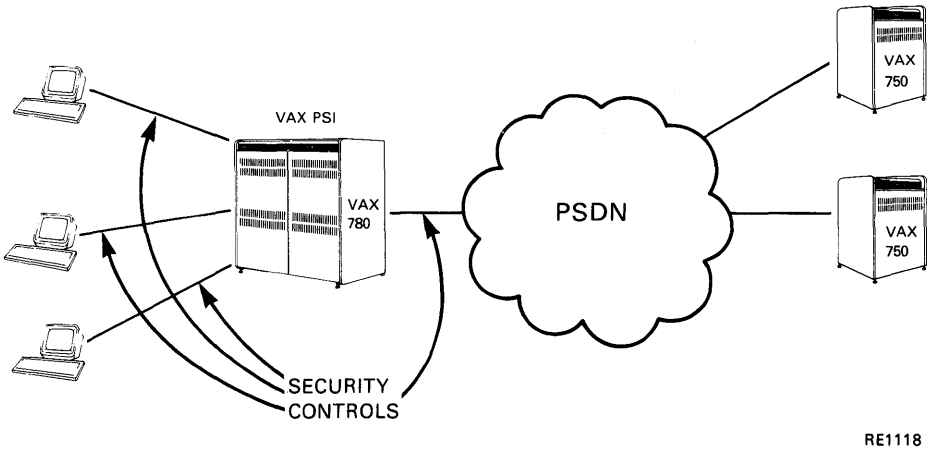
All incoming and outgoing calls are allowed until PSI Security is activated.

PSI Security is described fully in Part II of this manual (Chapters 4 to 7).

1.5.2 Allowing Outgoing Calls

One of the major tasks in managing the security system is deciding which local users you are going to allow to connect to specified remote DTEs. Probably the main consideration here is cost. This is because PSDNs are available worldwide, and you could incur large operating costs by allowing access rights indiscriminately.

Figure 1–4 PSI Security Controls



1.5.3 Allowing Incoming Calls

As well as deciding which users are allowed to connect to remote DTEs, you have to decide which remote DTEs are allowed to access your system, and the way in which these remote DTEs can use your system. For example, you could specify that certain remote DTEs are allowed to use your system only for electronic mail or file transfer.

1.6 Using the Accounting System

When VAX PSI is used, you can optionally record details of the way in which it is used. This may be so that you can charge users for X.25 resources or it may simply be for performance records. This information may also help you decide what security restrictions to use. You can record the following data:

- Data to allow you to calculate the cost of any call.
- Data to allow you to determine who was using the network at any given time (including the remote DTE involved).

  Note that for calls handled internally (for example, incoming X.29 calls), the data refers to PSI. See Section 9.4.3 for more information.
- All calls and attempts to call (including failed outgoing access) and all access to Permanent Virtual Circuits (PVCs), on Multi-host, Native mode, and Access VAX PSI systems.

When a set of accounting records have been produced, you can run the VAX PSI accounting utility (PSIACCOUNTING) to produce a report or alternatively you can process the records with your own program. A sample user accounting program is shown in Appendix A. Details of generating the accounting records and the accounting record formats are also included in this program. You can use the program as provided or adapt it to suit your own requirements.

PSI Accounting is described fully in Part III of this manual (Chapters 8 to 11).

1.7 Diagnosing Problems on VAX PSI

The tools available for diagnosing problems on VAX PSI are as follows:

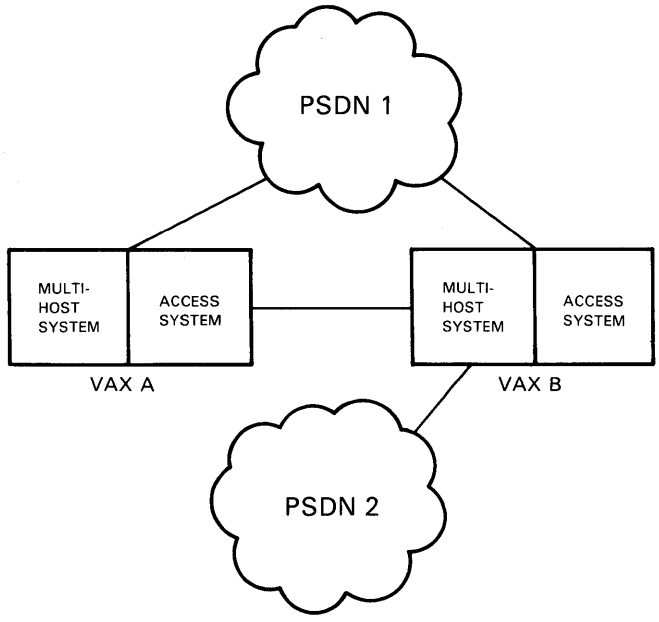
- Event Logs - which provide a record of protocol errors, circuits that are set up, and so on.
- The VAX PSI Trace utility, which monitors and collects data passing through VAX PSI that can be analyzed later.
- Loopback tests - over X.25 and DECnet lines.
- The KMS/KMV dump analyzer, which checks link level (level 2) data that has passed through the KMS or KMV device.

Refer to the *VAX P.S.I. Problem Solving Guide* for further information on problem diagnosis and how to use the tools mentioned above.

1.8 Managing a Combination System

A Combination system can be thought of as two separate VAX PSI systems, one Multihost mode and the other Access, both running on the same VAX computer.

Figure 1–5 A VAX PSI Combination System



Combination nodes provide additional flexibility as they operate either as Connector or Access nodes:

- Operating as a Connector node, a Combination node provides direct access for local users and applications to the PSDNs connected to it, and indirect access to these PSDNs for users and applications on the Access nodes connected to it.
- Operating as an Access node, a Combination node provides indirect access for local users and applications to other PSDNs via other Connector nodes.

If one of your VAX PSI systems is not working (for example, if the connection to the PSDN is down) then the applications that use VAX PSI could be switched to the other VAX PSI system. This need only involve changing the network default, PSI$NETWORK, to another system.

NOTE

1. If you have a Combination system, you should ensure that names in the configuration databases are unique across both the Access and Multi-host systems. This is particularly important

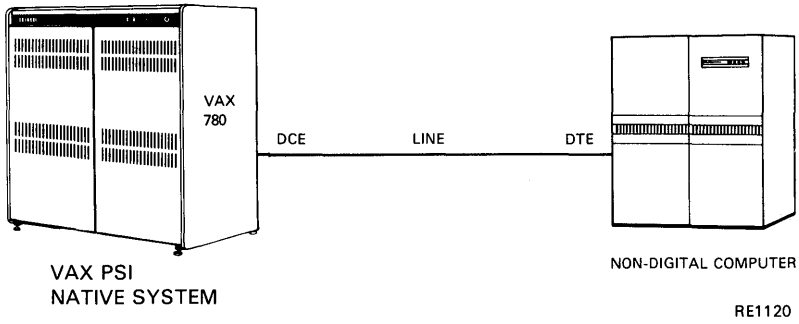
if you upgrade from an Access system to a Combination system.

2. When you are building a Combination system, the Multi-host software must be installed before the Access software.

1.9 Managing an ISO 8208 Network

One of the features of VAX PSI is its ability to act as a DCE rather than a DTE. This feature is useful for communicating with non-DIGITAL computers that use the X.25 protocol.

Figure 1–6 VAX PSI Acting as a DCE



The VAX PSI DCE uses the ISO 8208 protocol (a more closely defined version of CCITT X.25). Information about the ISO 8208 network is provided in the *Public Network Information* manual.

Refer to Section 2.6 for the NCP commands to set up VAX PSI as a DCE.

2
Configuring VAX PSI

When you have successfully installed VAX PSI for the first time, you will have built a basic system. This system can consist of one or more DTEs connecting to one or more networks, but often is just one DTE connecting to one PSDN. This chapter gives an overview of how you can add to a basic system.

The chapter starts by summarizing Network Control Program commands that you can use to add to your system. It then explains exactly which NCP commands were used during installation, and goes on to explain how you can make additions to your system.

2.1 Network Control Program (NCP) Commands

The Network Control Program (NCP) is a VMS utility program that is used to configure, monitor and test your network. NCP is used to control DECnet operations as well as those of VAX PSI. Full details of NCP are given in the *VMS Networking Manual* and *VMS Network Control Program Manual* in the VMS documentation set. However, this section summarizes how to use NCP to control VAX PSI.

At the heart of network management is the configuration database. The configuration database is a map of the physical and logical components of the network. It contains DTEs, lines, circuits, objects and modules. NCP commands are used to modify or display these network components.

At each node there are two copies of the configuration database available, the permanent database and the volatile database.

When you start the system the permanent database is the same as the volatile database. The permanent database provides the initial values for the volatile database. Because there are two databases, when you need to make changes to the configuration of your network, you can change the volatile database without losing the original configuration.

NCP commands are in three main groups, as shown in the following table.

Table 2–1 Network Control Program (NCP) Commands

NCP Command	Function	Database
SET net-component	Creating/Modifying	Volatile Database
DEFINE net-component	Creating/Modifying	Permanent Database
CLEAR net-component	Deleting	Volatile Database
PURGE net-component	Deleting	Permanent Database
SHOW net-component	Displaying	Volatile Database
LIST net-component	Displaying	Permanent Database

These commands act on network components in the appropriate database. The network components relevant to VAX PSI are:

- MODULE
- LINE
- CIRCUIT
- OBJECT

2.1.1 MODULE Commands

2.1.1.1 SET MODULE and DEFINE MODULE Commands

Both the SET MODULE and DEFINE MODULE commands have similar actions. The difference is that SET MODULE acts on the volatile database and DEFINE MODULE acts on the permanent database. The SET/DEFINE MODULE commands are as follows:

- SET/DEFINE MODULE X25-ACCESS - Used to associate the host node (your VAX PSI Access system) with one or more Connector nodes (VAX PSI Multi-host systems).
- SET/DEFINE MODULE X25-PROTOCOL - Used to create or specify a PSDN, a DTE, or a group of DTEs. Separate commands are used for each type of operation.
- SET/DEFINE MODULE X25-SERVER - Used to create or modify the parameters of the X.25 call handler.
- SET/DEFINE MODULE X29-SERVER - Used to create or modify the parameters of the X.29 call handler.

2.1.1.2 CLEAR MODULE and PURGE MODULE Commands

These commands have the opposite effect to SET MODULE and DEFINE MODULE. The command CLEAR MODULE acts on the volatile database and PURGE MODULE acts on the permanent database. The CLEAR/PURGE MODULE commands are:

- CLEAR/PURGE MODULE X25-ACCESS
- CLEAR/PURGE MODULE X25-PROTOCOL
- CLEAR/PURGE MODULE X25-SERVER
- CLEAR/PURGE MODULE X29-SERVER

2.1.1.3 SHOW MODULE and LIST MODULE Commands

These commands are used to display information. SHOW MODULE displays information from the volatile database. LIST MODULE displays information from the permanent database. The SHOW/LIST MODULE commands are:

- SHOW/LIST X25-ACCESS
- SHOW/LIST X25-PROTOCOL
- SHOW/LIST X25-SERVER
- SHOW/LIST X29-SERVER

2.1.2 LINE Commands

There are two LINE commands, SET LINE and DEFINE LINE, which act on the volatile and permanent databases respectively.

The SET/DEFINE LINE commands identify a line and specify parameters for the line, such as the maximum size of a frame, the maximum number of unacknowledged frames, the maximum number of retransmissions of a frame and the protocol (for example, LAPB, LAPBE or ETHERNET). The LINE commands are:

- SET/DEFINE LINE
- CLEAR/PURGE LINE
- SHOW/LIST LINE

2.1.3 CIRCUIT Commands

The SET/DEFINE CIRCUIT commands identify a circuit and specify parameters for the circuit, such as the channel number and DTE for Permanent Virtual Circuits, the maximum packet size and the maximum window size. The CIRCUIT commands are:

- SET/DEFINE CIRCUIT
- CLEAR/PURGE CIRCUIT
- SHOW/LIST CIRCUIT

2.1.4 OBJECT Commands

The SET/DEFINE OBJECT commands identify and specify parameters for a process that is activated by an incoming call. The parameters are the user account to be used by incoming calls, the password of the account and the name of a command file to activate a user program. The OBJECT commands are:

- SET/DEFINE OBJECT
- CLEAR/PURGE OBJECT
- SHOW/LIST OBJECT

2.2 Configuration Work Performed Automatically During Installation

When you install VAX PSI, you run the PSI_SET_UP command procedure that performs NCP commands for you. The following tasks are performed by PSI_SET_UP, as required:

- Definition of the connections to PSDNs. The NCP commands that the procedure performs differ, depending on whether you are installing a Native or an Access system.
- Changing a Native system into a Multi-host system.
- Starting the communications device or devices.

2.2.1 Defining the Configuration Database for Native Mode VAX PSI

The PSI_SET_UP procedure performs the following tasks:

- Deleting the existing contents of the permanent database by executing the following NCP command:

  ```
  PURGE MODULE X25-PROTOCOL KNOWN NETWORKS ALL
  ```

  which deletes all DTEs, groups and parameters for all networks from the permanent database.

- Defining modules as follows:

  ```
  DEFINE MODULE X25-PROTOCOL NETWORK network-id -
                             PROFILE profile-id
  ```

  which establishes the existence of a network with the name *network-id* in the permanent database. The profile identifies the PSDN to which the network will connect.

  ```
  DEFINE MODULE X29-SERVER STATE ON
  ```

  which allows the system to handle X.29 calls

  ```
  DEFINE LINE line-id PROTOCOL LAPB NETWORK network-id
                                    STATE ON
  ```

  which establishes the existence of a line (within the specified network) that uses the LAPB protocol in the permanent database.

  ```
  DEFINE MODULE X25-PROTOCOL DTE dte-address -
                             NETWORK network-id -
                             CHANNELS channel-list -
                             LINE line-id STATE ON
  ```

  which defines a DTE which receives information on the channels specified over line, line-id.

The command procedure repeats the above commands so that you can define more networks, DTEs or lines.

2.2.2 Defining Routes for Multi-host VAX PSI

The PSI_SET_UP procedure defines routes for incoming calls intended for host computers (running VAX PSI Access). Each destination is defined using the following command:

```
DEFINE MODULE X25-SERVER -
              DESTINATION dest-name -
              NODE dest-node -
              ACCOUNT account-name -
              CALL MASK call-mask -
              CALL VALUE call-value -
              GROUP cug-name -
              NUMBER dte-address -
              PASSWORD password -
              SUBADDRESS sub-address-range -
              USER user-id -
              PRIORITY dest-priority -
              OBJECT object-id
```

(There are additional parameters available on the DEFINE MODULE X25-SERVER command; refer to the *VMS Network Control Program Manual* for the complete list.) When incoming calls arrive from remote DTEs, VAX PSI matches them against destinations. The parameters that are compared are:

- Remote DTE address
- Local DTE subaddress
- User group name
- User data . . . .

If the incoming call matches more than one destination, the destination with the highest priority is taken as the match.

When a successful match has been obtained, VAX PSI forwards the call to the appropriate host (VAX PSI Access system). You define the host that receives the call with the NODE parameter.

Destinations are described further in Section 2.3.

2.2.3 Defining the Configuration Database for VAX PSI Access

The PSI_SET_UP procedure does the following:

1. Optionally deletes the existing contents of the permanent database - as shown in Section 2.2.1, except that the X25-ACCESS module is specified instead of the X25-PROTOCOL module.
2. Defines an Access system (or systems) in the permanent database, as follows:

```
DEFINE MODULE X25-ACCESS -
              NETWORK network-name -
              NODE node-name -
              USER user-name -
              PASSWORD password
```

2.2.4 Starting the Communications Devices for VAX PSI

The PSI_SET_UP procedure starts VAX PSI devices and optionally defines a destination for PSI MAIL. The NCP commands for defining the destination for PSI MAIL are as follows:

```
DEFINE OBJECT psi_mail NUMBER 0 -
                       FILE sys$system:psi_mail.com -
                       USER user-name -
                       PASSWORD password

DEFINE MODULE X25-SERVER DESTINATION psi_mail -
                         OBJECT psi_mail -
                         CALL MASK ffffffffffffffff -
                         ffffffffffffffff -
                         CALL VALUE ff00000056332e3 -
                         0204d41494c2d3131
```

Here, VAX PSI applies the mask to data in an incoming call. If the result is equal to the call value, this is a successful match and PSI_MAIL.COM starts the PSI MAIL process.

2.3 Setting Up Destinations

When a remote DTE tries to set up a call to your local DTE, both DTEs supply information to identify the destination that will accept the call. Local information is obtained from a destination database and an object database. You set up both databases with NCP commands.

There are two types of destination:

1. Local destinations - those for calls that are dealt with on the local system, where the local system is Native, Multi-host or Access.

2. Access destinations - those for calls that are forwarded to an Access system (from a Multi-host system).

2.3.1 Local Destinations

Each local destination in the destination database points to an object in the object database. Each object points to a command file which controls the start of a process for the incoming call.

2.3.2 Access Destinations

Access destinations on the Multi-host node do not have a corresponding object in the object database. All access destinations use object 36 (which is VAX PSI Access).

2.3.3 How Destinations are Matched

The destination database contains the identifying information for the local DTE. The remote DTE passes identifying information with the call. The information may include the DTE local subaddress, a closed user group name, values in the user data field, and the called address extension facility. When the call arrives, the VAX PSI software tries to find a match between the information contained in the incoming call and one of the destinations in the destination database.

If the incoming call has additional information other than call user data, which is not defined for the destination with the closest match, the VAX PSI software passes the call to this destination. If there is no match between the identifying information and a destination in the destination database, the VAX PSI software rejects the call.

If the identifying information matches more than one destination, the VAX PSI software allocates the call to the destination with the highest priority. When the VAX PSI software has allocated the call to a destination, either the call is passed to an Access system or the object associated with the destination is used to start up the user process.

2.3.3.1 Destination Checking

You can use the CALL MASK and CALL VALUE parameters in the User Data Field to specify a call mask and a call value. A call mask and a call value are used to test the incoming call to see if the call can be passed to the destination.

If you specify these values, VAX PSI extracts the Call User Data from the incoming call request and performs a logical AND operation between this data and the call mask. The result of this operation is compared with the call value. If the fields match, the incoming call is passed to the destination.

For example, the following command indicates that destination JOE will accept only incoming calls that contain value 11 in their User Data Fields:

```
NCP>SET MODULE X25-SERVER DESTINATION JOE CALL VALUE 11 -
_ CALL MASK FF ...
```

To specify call mask and call value, use stings of 2 to 32 hexadecimal digits for the two parameter values.

2.3.4 NCP Commands for Setting Up Destinations

2.3.4.1 Setting Up Local Destinations

Set up local destinations using NCP commands as shown in the following example:

```
NCP>DEFINE MODULE X25-SERVER DESTINATION ONE -
_ SUBADDRESS 1-10 OBJECT OBJONE PRIORITY 1
NCP>DEFINE MODULE X25-SERVER DESTINATION TWO -
_ SUBADDRESS 11-15 OBJECT OBJTWO PRIORITY 2
```

where ONE and TWO are destination names. The destinations in this example use subaddress ranges as the criteria for accepting incoming calls. Other criteria could be closed user group names, call data, the called address extension facility or a combination of these.

2.3.4.2 Setting Up Access Destinations

Set up Access destinations as shown in the following example:

```
NCP>DEFINE MODULE X25-SERVER -
                  DESTINATION THREE -
                  SUBADDRESS 16-30 -
                  OBJECT 36 -
                  NODE ARTHUR
```

where THREE is a destination name. This is a destination on a Multi-host system that forwards calls to an Access system on node ARTHUR. Object 36 is the DECnet object for VAX PSI Access.

When the call arrives at ARTHUR, the call will be matched against local destinations, as described in Section 2.3.1.

2.3.5 Setting Up Objects

Set up objects using NCP commands as shown in the following example:

```
NCP>DEFINE OBJECT OBJONE FILE OBJSTUP.COM -
_ USER JOE PASSWORD JOSEPH NUMBER 0

NCP>DEFINE OBJECT OBJTWO FILE OBJTWO.COM -
_ USER FRED PASSWORD FREDDY NUMBER 0
```

This example sets up the objects for the destinations in the previous example. When you set up objects, define a file containing a command procedure to run the image for the user process. Specify a user name and password to define the account under which the file is to run.

The default file specification for objects used by VAX PSI has a device and directory name derived from the logical name SYS$LOGIN and an extension of .COM.

2.3.6 Process Names

When VAX PSI creates a process, the process name is derived from the object name set up by the system manager. See the *VMS Networking Manual* for a full description of setting up object names.

A process name is as follows:

name-n

where *name* is the object name, and *n* is a unique number assigned by the VAX PSI software. For example, process names can be:

```
MAIL-92
XFER-1034
```

A process name can be up to 15 characters long. Where the object name plus *n* is is greater than 15 characters, the object name is truncated. For example, if the object name is LONGOBJECTNAME, and *n* is 20375, the resulting process name is:

```
LONGOBJEC-20375
```

2.3.7 Incoming X.29 Calls

Incoming X.29 calls are handled in a similar manner to X.25 calls. If you log in on a PSDN PAD, the calls will automatically be treated as X.29 calls. You can specify destinations for X.29 calls with NCP commands, in the same way as for X.25 calls. For example:

```
NCP>DEFINE MODULE X29-SERVER DESTINATION X29CALLS -
_ SENDING ADDRESS 234567890123 -
_ PRIORITY 20
```

The above X.29 destination handles X.29 calls from DTE 234567890123. Make sure the X29-SERVER module is turned on. Use an NCP command as follows:

```
NCP>DEFINE MODULE X29-SERVER STATE ON
```

Note, however, that if no object is specified in the destination, LOGINOUT is invoked to log in the user as a pseudo-terminal.

Define X.29 destinations and set the X.29 call handler on if you require X.29 support, as in the following example:

```
NCP>DEFINE MODULE X29-SERVER DESTINATION BUNNY -
_ PRIORITY 50 -
_ SUBADDRESS 10-30
NCP>DEFINE MODULE X29-SERVER DESTINATION RABBIT -
_ PRIORITY 20 -
_ SENDING ADDRESS 234567890312
NCP>DEFINE MODULE X29-SERVER STATE ON
```

2.3.7.1 X.29 Destination Checking

Optionally, you can use the CALL MASK and CALL VALUE parameters to indicate that a destination will only accept incoming X.29 calls. The CCITT (Comite Consultatif International Telegraphique et Telephonique) recommends that you use a call value of 01 for incoming X.29 calls.

For example,

```
NCP>SET MODULE X29-SERVER DESTINATION JIM CALL VALUE 01 -
_ CALL MASK FF ...
```

indicates that only incoming X.29 calls that contain the value 01 in their User Data Fields will be passed to destination JOE.

You can use these two parameters to further identify X.29 calls when the terminal user first connects from a PAD. To do so, specify a destination for the X29-SERVER that recognizes a value entered as call data when connecting to the PAD.

For example,

```
NCP>SET MODULE X29-SERVER DESTINATION JAYNE CALL VALUE 0000000041 -
_ CALL MASK 00000000FF ...
```

indicates that the destination JAYNE will accept only incoming X.29 calls that specify a call data value of A (41 is the hexadecimal value for A). Note that in this example, the value 01 in the User Data Field has been masked out.

2.4 Setting Up the X25-ACCESS Database

On each of the DECnet-VAX host systems, set up the X25-ACCESS database using NCP commands as shown in the following example:

```
NCP>DEFINE MODULE X25-ACCESS NETWORK PSS1 -
  NODE ROBIN
NCP>DEFINE MODULE X25-ACCESS NETWORK PSS2 -
  NODE THRUSH
NCP>DEFINE MODULE X25-ACCESS NETWORK TELENET -
_ NODE LARK
```

In this example, ROBIN, THRUSH and LARK are Multi-host VAX PSI nodes, where nodes ROBIN and THRUSH are both connected to PSS, and LARK is connected to TELENET.

Where there is more than one connection to a network, also set up a default network in the system logical name table. See the *VAX P.S.I. Installation Procedures* for details.

2.5 Setting Up DLM Circuits

To set up DLM circuits, you use different NCP commands for incoming and outgoing circuits, as described in the following sections.

2.5.1 Setting Up Incoming DLM Circuits

PSI identifies an incoming call as a DLM call by its subaddress. You must define a range of subaddresses to be used by DLM. Use an NCP command as shown in the following example:

```
NCP>DEFINE EXECUTOR SUBADDRESSES 1-5
```

In this example, the local DTE receiving incoming calls with subaddresses in the range 1 to 5 will treat them as DLM calls. Set up an incoming DLM circuit for this DTE using an NCP command as shown in the following example:

```
NCP>DEFINE CIRCUIT X25-INC STATE ON OWNER EXECUTOR -
_ USAGE INCOMING TYPE X25
```

Note that you do not specify the subaddress range when you are defining the incoming circuit, but you must specify it when defining the executor node.

Note also that you may reserve a DLM circuit for handling incoming calls from a particular DTE by specifying the NUMBER parameter in the NCP command:

```
NCP>DEFINE CIRCUIT X25-INC STATE ON OWNER EXECUTOR -
_ USAGE INCOMING TYPE X25 NUMBER 234567890321
```

2.5.2 Setting Up Outgoing DLM Circuits

When setting up a DLM circuit to send calls, you need to know the range of subaddresses that the remote DTE will accept. Set up an outgoing DLM circuit using an NCP command as shown in the following example:

```
NCP>DEFINE CIRCUIT X25-OUT STATE ON OWNER EXECUTOR -
_ USAGE OUTGOING TYPE X25 NUMBER 2345678901231
```

In this example the DTE sets up an outgoing DLM circuit to call a DTE with an address of 234567890123 and a subaddress of 1. The incoming DLM circuit set up in the example without the number parameter above could accept this call because the subaddress is within the DLM range of 1 to 5.

2.5.2.1 Other DLM Circuit Parameters

Do not alter the default values set for the MAXIMUM RECALLS and RECALL TIMER parameters without considering the effect this action might have. If you change the value of either of these parameters, DLM will continue trying to make the connection until it reaches the limits you have set. The local PSDN may not allow you to do this, because repeated calls generate a lot of unnecessary network traffic. The local PSDN may therefore take steps to stop your DTE transmitting.

When you set a value for the HELLO TIMER parameter, the LISTEN TIMER parameter is automatically set to twice that value. DECnet uses these timers to check on the state of the circuit. If you set these timers, your DLM circuit remains active sending hello messages, even if there is no other traffic on the circuit. This could prove expensive.

For DLM circuits, it is better to set the HELLO TIMER value (and hence the LISTEN TIMER value) to its maximum so that the charges you incur are minimized when the circuit is inactive. Note that if you increase the HELLO TIMER value, the DECnet routing module will take longer to detect a circuit failure.

2.6 Setting Up VAX PSI as a DCE

To set up VAX PSI as a DCE, you have to specify the following in the configuration database:

1. That the network is an ISO 8208 network, as defined in the ISO 8208 profile.

   For example,

   ```
   DEFINE MODULE X25-PROTOCOL NETWORK NONDEC -
                              PROFILE ISO8208
   ```

   This specifies a network name, NONDEC which has the ISO 8208 profile.

2. That the line is to have a DCE interface. For example,

   ```
   DEFINE LINE DMF-0 NETWORK NONDEC -
                     INTERFACE DCE -
                     STATE ON
   ```

   This specifies that the line connects to the network NONDEC as a DCE.

3. That the DTE will have a DCE interface. For example,

   ```
   DEFINE MODULE X25-PROTOCOL DTE 123456789012
                              NETWORK NONDEC
                              LINE DMF-0
                              CHANNELS 100-1
                              INTERFACE DCE
   ```

   This specifies that DTE 123456789012 will act as a DCE. Specify the channels in decreasing order (100-1, for example) to comply with International Standard 8208.

   When VAX PSI acts as a DCE, it reverses the channels list you specify. CHANNELS 100-1 becomes CHANNELS 1-100, for example. Thus, you can specify the same channels list on both systems, with low risk of call collisions.

2.6.1 A Negotiated Interface

VAX PSI also supports negotiation of the interface type at the packet level, as described in International Standard 8208. This specifies that two systems can decide between themselves which one of them will act as a DTE and which one will act as a DCE.

To use this capability specify:

```
DEFINE MODULE X25-PROTOCOL
               DTE ......
               NETWORK ......
               LINE ......
               CHANNELS ......
               INTERFACE NEGOTIATED
```

Note that you have to specify INTERFACE DTE or INTERFACE DCE when you define the line.

2.6.2 Setting Up Destinations for a DCE

When VAX PSI acts as a DCE, it is effectively a DTE with added DCE functionality. So any destinations you set up should be similar to those for a regular DTE. Remember, when using VAX PSI system as a DCE, you are likely to be communicating with a Non-DIGITAL computer, and therefore destinations need not be as selective as for communication with a number of remote DTEs.

3

Testing the Configuration of VAX PSI

3.1 Introduction

This chapter explains when and how to use the Configuration Test Program (CTP) to check the installation and full configuration of your VAX PSI system. This chapter also explains when to use the Installation Checkout Procedure (ICP) to check the installation and basic configuration of your system.

3.2 When to Use the Configuration Test Program

The CTP is a new feature for VAX PSI V4.2, and determines whether you have installed and configured the VAX PSI or VAX PSI Access software on your system correctly.

NOTE

- Use the CTP if you have VAX PSI V4.2 installed and running on your system AND on any remote system you want to use for the configuration test.
- Use the old ICP if you have an earlier version of VAX PSI installed and running on your system, OR if the remote system you want to use for the configuration test has an earlier version of VAX PSI or RSX PSI installed and running. See Appendix B for more information about the ICP.

The Configuration Test Program can be used to:

- Check the installation and basic configuration of VAX PSI on your system after you have installed and set up the software for the first time (as described in the *VAX P.S.I. Installation Procedures* manual). Note that you can also use the Installation Checkout Procedure to do this.
- Check the installation and full configuration of VAX PSI on your system after you have added to or modified the full configuration of VAX PSI on your system (as described in Chapter 2 of this manual).

The CTP is used to confirm the network configuration of your VAX PSI system by testing the access points to your PSDN. It is not a problem solving tool; however it can be used to confirm that the network configuration of your system is correct if a problem occurs when you are using VAX PSI.

For more information about the problems that may occur when you are using VAX PSI, and how to solve these problems, see the *VAX P.S.I. Problem Solving Guide*.

3.3 The Configuration Test Program

The CTP is a program designed specifically as a network configuration testing tool. The CTP tests the access points to PSDNs from your system and the routes to remote systems. The CTP acts as a test send program that activates a test receive program to check that the configuration on your VAX PSI system can set up a call and transfer data. Testing consists of accessing a circuit, testing the interrupt procedure, transferring data and de-accessing the circuit.

3.3.1 Loopback and Remote System Testing

You can run the CTP either in loopback from your system through the PSDN and back to your system, or across the PSDN to a remote system.

To run the CTP in loopback, you must have VAX PSI installed, configured and running on your system. Loopback testing is not possible for PVCs and for PSDNs which do not allow virtual circuits to be switched back to the local DTE. Also, it is not possible to run the CTP in loopback to test point to point connections to other VAX PSI systems.

To run the CTP from your system to a remote system, both systems must have VAX PSI installed, configured and running. (Note that testing to a remote system is not possible if the remote system does not have VAX PSI V4.2 installed, configured and running.)

See Section 3.3.2 for information about when to run loopback and remote system testing.

3.3.2 When to Use Loopback or Remote System Testing

Use loopback testing if you simply want to test access to a PSDN, or if a remote system running VAX PSI V4.2 is not available. Note that your PSDN must allow loopback from the network to your system. Loopback testing cannot be done over PVCs and point to point links using ISO 8208 profiles.

Use remote system testing if you want to test access to a remote node with VAX PSI V4.2 installed and running. Note that PVCs cannot be tested using this method if the CTP is set up only as a network object on the remote system.

3.4 What the Configuration Test Program Does

The CTP tests the network configuration that you have set up on your system by checking that the access points to a PSDN are operational. There are three types of access point that can be checked by the CTP:

- Locally configured DTEs for Native, Multi-host or Combination nodes.

  These local DTEs are configured in the DTE database of the VAX PSI Configuration Database on the local node. The local DTEs are checked by making a virtual call across the PSDN to transfer data with a remote system, either in loopback to the local node or to a remote system.

- Configured PVCs for Native, Multi-host and Combination nodes.

  These PVCs are configured in the circuit database of the VAX PSI Configuration Database. A PVC provides a permanent logical association between a local DTE associated with the local node and a remote DTE associated with a remote system. The PVCs are checked by transferring data across the PVC from the local node to the remote system.

- Network configurations for Access nodes.

  These network configurations are defined in the X25-ACCESS Module of the configuration database on the Access node. The network configurations are checked by making a virtual call across the PSDN to transfer data with a remote system, via a Connector node and associated local DTE.

3.5 Setting Up the Configuration Test Program

The CTP can be run in one of two operating modes:

- Send/Receive Mode

  In this mode, the CTP is run interactively. The program initiates testing of the configuration and creates a process to handle incoming test virtual calls.

- Receive Only Mode

  In this mode, the CTP is run either interactively, or as a network object. The program creates a process to handle incoming test virtual calls and to handle incoming tests on configured PVCs.

To run the test in loopback, the CTP must be run interactively in Send/Receive Mode.

To run the test to a remote system, the CTP must be run interactively in Send/Receive Mode on the local or host node, and in Receive mode only on the remote system. This can be done on the remote system interactively, or by setting up the program as a network object. Note that PVCs can only be tested by running the program interactively.

3.6 How to Run the Configuration Test Program

This section explains how to set up and run the CTP for different configurations of VAX PSI, both in loopback and to remote systems.

3.6.1 Required Privileges and Quotas

To run the CTP, you need NETMBX, TMPMBX, WORLD, CMKRNL, DETACH and SYSPRV privileges.

In addition, you need the following quotas:

- ASTLM - Asynchronous system trap limit = 100
- BIOLM - Buffered I/O limit = 100
- BYTLM - Buffered I/O byte limit = 40000
- DIOLM - Direct I/O limit = 100
- TQELM - Timer queue limit = 30

3.6.2 Preparing to Run the CTP

Before executing the CTP, check the following conditions:

1. VAX PSI is installed and configured on your system.
2. VAX PSI is up and running on your system.
3. If you are running the CTP to a remote node, you must also ensure that:
   - The remote node has VAX PSI V4.2 installed, configured and running.
   - You have the remote DTE and subaddress available for the remote node.
   - The remote node is running the CTP in receive mode.

In addition, you should check that the DTEs you want to use for the test are up and running, and that the remote node destination parameters are correct.

Checking that a DTE is up and running

For a Native or Multi-host system, to check that a DTE is up and running, use the following command:

```
NCP>SHOW MODULE X25-PROTOCOL KNOWN DTES
```

For an Access system, use the following command:

```
NCP>TELL node-id SHOW MODULE X25-PROTOCOL KNOWN DTES
```

where *node-id* is the name of the Connector node.

If a DTE is up, it appears in the summary display produced by the command with a state and substate of ON-RUNNING.

If no DTEs are up, wait for two minutes before issuing the command again. If the display shows that there are still no DTEs up, refer to the *VAX P.S.I. Problem Solving Guide*.

Checking the remote node destination parameters

If you are checking a Native or Multi-host system, make sure that the remote node destination parameters are correct by issuing the following command:

```
NCP> SHOW MODULE X25-SERVER DESTINATION remote-node-id
```

where *remote-node-id* is name of the remote node.

For Access systems, issue the following command:

```
NCP> TELL node-id SHOW MODULE X25-SERVER DESTINATION remote-node-id
```

where *node-id* is the name of the Connector node, and *remote-node-id* is the name of the remote node.

Check the displayed details against the information supplied in the *VAX P.S.I. Installation Procedures* and amend as necessary.

3.6.3 Running the CTP Interactively

To run the CTP interactively and to start the CTP, enter the following command at the DCL prompt:

```
$ RUN SYS$TEST:PSI$CTP
```

The program starts and displays general information about the CTP, and the conditions that must be satisfied before the CTP can be run. (Press <RET> to continue reading the information displayed.)

At the end of this information, the program asks if you wish to continue running the CTP. Press <RET> to continue, and the program then asks which operating mode you wish to use.

Enter one of the options, either Send/Receive Mode or Receive Only Mode, as follows:

- Select Send/Receive Mode to test your local DTEs, access networks or PVCs. This is the default mode.
- Select Receive Only Mode to set up your system for incoming virtual call tests from a remote system.

3.6.3.1 Running the CTP in Send/Receive Mode

Run the CTP in Send/Receive Mode to test your local DTEs, access networks and PVCs. In this mode the program initiates testing of the configuration and creates a process to handle incoming test virtual calls.

Testing the Local DTE Configurations

If your system is a Native, Multi-host or Combination node, and has local DTEs configured, you can test the local DTEs either in loopback, or to a remote system.

NOTE

By default, the program is set up in loopback, and uses the address for each local DTE you select as the default for loopback testing to that local DTE. You can specify a default remote DTE address for testing to a remote system by entering the address when prompted.

The program displays information about each local DTE configured for your system. Select the local DTEs you wish to test by pressing <RET>. Note that only local DTEs in the "ON" state can be tested. When prompted, select either the default remote DTE address displayed, or enter a new remote DTE address.

When you have selected the local DTEs and have entered the relevant information for each, the testing begins. There is a pause while the CTP tests the local DTEs.

After the testing, the program displays a table of the local DTEs with an indication under the *Result* column showing the success or failure of each test. Press <RET> for an analysis display for each test failure.

Note the reason for each failure, and refer to Section 3.7 for more information.

Testing the PVCs

If your system is a Native, Multi-host or Combination node and has configured PVCs, the program displays information about each PVC configured for your system. Select the PVCs you wish to test.

When you have selected the PVCs, the testing begins. There is a pause while the CTP tests the PVCs by transferring data across each PVC to the remote systems.

After the test has completed, a table of the PVCs is displayed with an indication under the *Result* column of the success or failure of the test for each PVC. You are prompted for the analysis display for each test failure.

Note the reason for each test failure, and refer to Section 3.7 for more information.

Testing Access Networks

If your node is an Access or Combination node, and has access networks configured, you can test the access networks to a remote system. You can enter a default remote DTE address for testing to the remote system when prompted.

The program displays information about each access network configured for your system. Select the access networks you wish to test by pressing <RET>, and by selecting either the default DTE address you have chosen, or by entering another remote DTE address.

When you have selected the access networks and have entered the relevant information for each, the testing begins. There is a pause while the CTP tests the access networks.

After the testing, the program displays a table of the access networks with an indication under the *Result* column showing the success or failure of each access network test. Press <RET> for an analysis display for each test failure.

Note the reason for each test failure, and refer to Section 3.7 for more information.

3.6.3.2 Running the CTP in Receive Only Mode

Run the CTP in Receive Only Mode on your system to test incoming virtual calls from a remote system. In this mode, the CTP creates a process to handle incoming test virtual calls and to handle incoming tests on configured PVCs. Note that the remote system must be running the CTP in Send/Receive Mode.

Setting Up SVCs

When you select Receive Only Mode, the CTP enables the receiver program for SVCs to handle the incoming test virtual calls from the remote system.

Setting Up PVCs as Test Receivers

You can set up any configured PVCs as test receivers by selecting the circuits when prompted by the CTP.

After you have selected the PVCs you wish to test, the CTP displays a summary of the selected circuits with an indication that they have been set up correctly. Your system is now ready to receive incoming tests from the remote system.

When the tests are complete, you can run the CTP again or exit the program.

3.6.4 Running the CTP as a Network Object

This section describes how to run the CTP as a network object to handle incoming test virtual calls only. The CTP can be set up in this way on any system for loopback testing through a PSDN, or on the local system for testing across a PSDN from a remote system.

To set up the CTP as a network object, enter the following command at the DCL prompt:

```
$ @SYS$TEST:PSI$CTP_ADD_NETOBJ.COM
```

This command procedure sets up the CTP as a network object in the permanent and volatile databases.

If you want to run the CTP as a network object all the time, you can add this command to the PSI startup command procedure, STARTPSI.COM immediately after the $exit: label.

To remove the CTP as a network object, enter the following command at the DCL prompt:

```
$ @SYS$TEST:PSI$CTP_REM_NETOBJ.COM
```

This command procedure removes the CTP as a network object in the permanent and volatile databases.

3.7 CTP Failure Reasons

3.7.1 Send/Receive Mode Failures

When you run a CTP in Send/Receive Mode, it either passes or fails. If the CTP passes, the routes and access points to the PSDN are configured correctly. If it fails, a reason for the failure is displayed.

Table 3–1 provides information about CTP failures. Note the reason and cause of any failure and refer to the *VAX P.S.I. Problem Solving Guide* if necessary.

Table 3–1 Reasons for CTP Send/Receive Mode Failure

Reason for Failure	Cause of Failure	Notes
QIO failed	Call cleared unexpectedly	1. Remote system not set up in receive mode 2. Configuration incorrect 3. The DTE or Gateway link has gone down
	The circuit RESET unexpectedly	1. Remote system not set up in receive mode (PVCs only) 2. The network initiated a reset
	The DTE or Gateway link is down, or has gone down	
	The PVC has already been accessed	
	Invalid remote DTE address supplied	

Table 3–1 (Cont.) Reasons for CTP Send/Receive Mode Failure

Reason for Failure	Cause of Failure	Notes
Unsolicited mbx message	Call cleared unexpectedly	1. Remote system not set up in receive mode 2. Configuration incorrect 3. The DTE or Gateway link is down, or has gone down
	The circuit RESET unexpectedly	1. Remote system not set up in receive mode (PVCs only) 2. The network initiated a reset
	Unexpected data received	Data corruption between sender and receiver
	Unexpected interrupt received	Data corruption between sender and receiver
	The DTE or Gateway link is down, or has gone down	
Test timeout expired	Receiver not responding	
	Data lost by network	
Read/Write data mismatch	Data corrupted between sender and receiver	

3.7.2 Receive Only Mode Failures

When you run the CTP in Receive Only Mode and set up PVCs as receivers, the PVC set up may fail. Table 3–2 provides information about PVC receiver failures.

Table 3–2 Reasons for CTP Receive Only Mode Failure

Reason for Failure	Cause of Failure	Notes
Failed to set up receiver	PVC already accessed	Deaccess PVC before running the CTP again in Receive Only Mode

II

PSI Security

This part of the manual consists of four chapters:

- Chapter 4 - Provides an introduction to PSI Security and describes the concepts behind the security system.
- Chapter 5 - Describes how PSI Security works, and gives some examples to show how the security databases are used.
- Chapter 6 - Describes how you use PSI Security to set up the security databases on your system.
- Chapter 7 - Contains a command reference section for the PSI Security utility, PSIAUTHORIZE.

4

Introduction to PSI Security

4.1 Security in Packet Switching Data Networks

Packet Switching Data Networks (PSDNs) have become very wide spread throughout the world and new networks are being introduced all the time. Most of these are public networks, and they are interconnected via gateways. For this reason, it is realistic to talk about a worldwide public data network with thousands of connected systems and many more remote terminals using public or private PADs.

If you connect your system to a public PSDN in this global networking environment, all other systems connected to the global network have the potential to connect to your system. Also, users on your system have the potential to connect to all other systems in the global network.

This potential to connect to all of these other systems has obvious advantages. However, there is an increasing need to be able to control access to and from systems connected to public PSDNs.

As more companies and individuals start to use public PSDNs for data communications, increasing amounts of sensitive information are transmitted over these networks. In addition, more users are granted access to the systems that store and handle confidential information. Because of this, companies and individuals are becoming more concerned about the security of their systems in a public PSDN environment.

Some public PSDNs provide optional security facilities such as the use of passwords, Closed User Groups (CUGs) and incoming only and outgoing only DTEs. For more information about the security facilities offered by your public PSDN, see your PSDN documentation or consult your PSDN authority.

Security in a private PSDN is usually easier to achieve because the network and the systems connected to it are controlled by the authority that owns the network. However, some private PSDNs have gateways to public PSDNs. This means that, from a security point of view, the systems connected to the private PSDN are as exposed to potential security breaches as if they were connected directly to public PSDNs.

There are three main areas where network security may be breached:

- **Physical** - can unauthorized users gain access to the network?
- **Transmission** - how safe is data when in transit across the network?
- **System** - how safe is the data stored on the individual systems connected to the network?

Physical Security

The physical security of the network is the responsibility of the network authority. For public data networks, the authority is usually the local PTT and, in such cases, the PTT allocates and controls the physical access to the network. For private PSDNs, the network authority is usually the owner of the network, such as a large company or organization. Often, other organizations may be permitted to use these private networks. In such cases, the physical security of the private network often becomes the focus for network security.

Transmission Security

The security of data during transmission across a network depends also on the physical security of the network, and on the security of the network systems that handle and transmit the data. Users need to be sure that data is sent to the specified DTE and cannot be intercepted by unauthorized users.

The way in which PSDNs operate enhances data security during transmission because the data is sent in packets. Each packet is independently routed across the network to the destination. This makes the interception of complete data by physical means very difficult.

Some systems connected to PSDNs also use data encryption software or PSDN supplied encryption devices to increase the security of their data when transmitted across the network. The software or the devices must be installed at both the sending and receiving systems to ensure that the transmitted data can be recovered by the receiving system.

NOTE

PSI Security is NOT a form of data encryption software, nor does it include any data encryption facilities.

System Security

The security of a system connected to a PSDN is the responsibility of the company or individual running the system. System security covers the protection of the system against the observation, use, theft or corruption by unauthorized users of the data stored on that system.

The problem of system security increases when certain outside users are allowed access via a PSDN to certain areas of the system. For large systems connected to public PSDNs, the security controls need to provide at least the basic mechanisms to control access to the system and its data. A lack of security at a system level can undermine security at a network level, no matter how secure the network itself may be.

4.2 What is PSI Security?

PSI Security is a facility that lets you control the use of VAX PSI on your system. You can think of PSI Security as an additional form of system security, designed to help you implement network security. Using PSI Security, you can control access:

- To your system, for remote DTEs that want to make incoming calls to users and destinations on your system.
- From your system, for users and processes that want to make outgoing calls to remote DTEs.

You can decide which remote DTEs are allowed to make incoming calls to your system, and the applications that these remote DTEs are allowed to use. For example, you could restrict certain remote DTEs to using the PSI Mail utility, while allowing others to use an X.25 utility written for your system.

You can decide also which local users are allowed to make outgoing calls to particular remote DTEs. This facility allows you to control access to the PSDNs connected to your system, and to keep network operating costs to a minimum. PSDNs are available worldwide, and you could incur large operating costs if you allow users to make outgoing calls at will.

For example, within your company, you could allow all users on your system to make outgoing calls to other company locations in the same country. Outgoing calls to company locations in other countries and to other organizations could be restricted to specific users only.

When you first install VAX PSI on your system, you have the option of either using the security system or leaving it inactive. By default, the security system is inactive.

If PSI Security is inactive, users only require NETMBX privilege or the rights identifier PSI$X25_USER to use VAX PSI to send and receive calls across the PSDN. Until PSI Security is activated, there is no full control over incoming and outgoing calls across the PSDN.

If you have selected to use the security system, you must edit the PSI_SECURITY.COM command file in SYS$MANAGER to include the VAX PSI Security commands you require.

NOTE

1. All incoming and outgoing calls are allowed by VAX PSI until the security system is activated.
2. The security system is activated by the command procedure STARTPSI.COM each time the system is rebooted.
3. The PSI Security databases, created when the security system is activated, are volatile.

You activate PSI Security either by running the PSI_SECURITY.COM command file you have created, or when you add security data to the PSI Security databases by using the PSIAUTHORIZE utility. Once you have set up PSI Security correctly, your system is protected.

The level of protection for your system depends on the security controls you have set up. You can modify the level of protection at any time by using the PSIAUTHORIZE utility to modify the data in the security databases.

NOTE

The security databases you set up using PSIAUTHORIZE are volatile. You MUST edit the PSI_SECURITY.COM command file in SYS$MANAGER to include the PSI Security commands you have entered using the PSIAUTHORIZE utility.

When you reboot your system, the command procedure STARTPSI.COM activates PSI Security by running PSI_SECURITY.COM to reset the security databases.

4.3 VAX PSI Privilege Checks

You can define two optional rights identifiers that can be used to provide basic security on the use of VAX PSI. These identifiers are:

- PSI$X25_USER - this identifier must be owned by a user or process that wants to make or accept calls using VAX PSI.
- PSI$DECLNAME - this identifier must be owned by a process that wants to declare itself as an X.25 network process.

Note that these rights identifiers provide only basic security controls. You must set up PSI Security to provide full security for your system.

See Section 5.1.2 for more information about the rights identifiers PSI$X25_USER and PSI$DECLNAME and the privilege checks performed by PSI Security.

4.4 VAX PSI Configurations

When setting up and using PSI Security, it is important to note the configuration of VAX PSI on your system. The configuration of VAX PSI determines the security facilities applicable to your system.

There are four configurations of VAX PSI:

- Native mode
- Multi-host mode
- Access mode
- Combination mode

Native Mode

Native mode provides direct access to one or more PSDNs for users and applications on a single VAX/VMS system with VAX PSI installed.

Multi-host Mode

Multi-host mode provides direct access to one or more PSDNs for users and applications on a VAX/VMS node with VAX PSI installed, and indirect access for other VAX/VMS systems. The node with VAX PSI installed is a Multi-host node, and serves as a gateway to the PSDN. The other systems are Access nodes and must have VAX PSI Access installed. All systems must have DECnet installed and running.

Note that because a node with VAX PSI installed in Multi-host mode connects directly to one or more PSDNs, it is known also as a Connector node. See Section 4.4.1 for more information about PSI Security and Connector nodes.

Access Mode

Access mode provides a VAX/VMS host node with indirect access to one or more PSDNs via a Connector node. The host node must have VAX PSI Access installed, and all systems must have DECnet installed and running.

Combination Mode

Combination mode combines the capabilities of both Multi-host and Access modes. A Combination node has both VAX PSI and VAX PSI Access installed, and provides both direct and indirect access to one or more PSDNs for users and applications on the local node and other VAX/VMS host nodes. All nodes must have DECnet installed and running.

For more information about VAX PSI configurations, see the *P.S.I. Introduction* and the *VMS Networking Manual*. For information about setting up and configuring VAX PSI and VAX PSI Access, see Chapter 2.

4.4.1 PSI Security and Connector Nodes

A Multi-host node with VAX PSI installed and which connects directly with one or more PSDNs is known also as a Connector node.

DIGITAL's X25router communications product is also referred to as a Connector node. This product offers similar facilities to VAX PSI in Multi-host mode, and provide indirect access to one or more PSDNs for one or more VAX PSI Access nodes.

Note, however, that the X25router does not support the PSI Security facility. If you have an X25router installed, you should ensure that PSI Security is set up for each node running VAX PSI Access on your DECnet network.

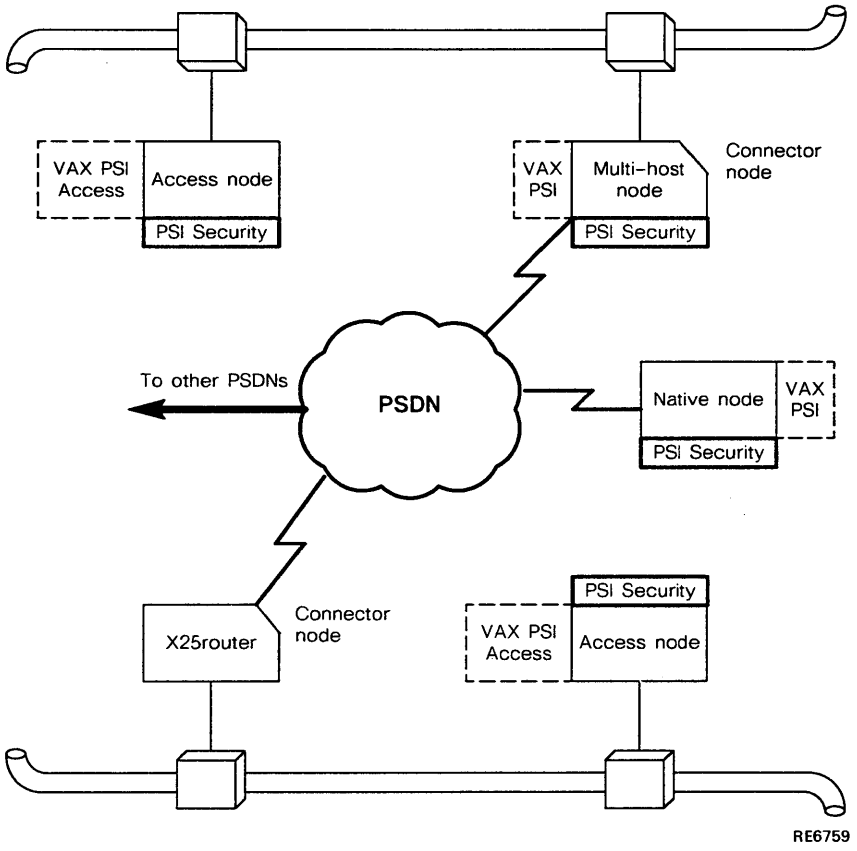
NOTE

Multi-host node is a term used to refer to a node running VAX PSI in Multi-host mode, and which therefore supports PSI Security.

Connector node is a generic term used to refer to any system which connects directly to a PSDN. Such a system may or may not support PSI Security. For example, a VAX PSI Multi-host node supports PSI Security, while an X25router does not. However, as PSI Security is supported on VAX PSI Access nodes, you can implement PSI Security on Access nodes in your network.

Figure 4–1 shows the various configurations of VAX PSI and their access to a PSDN. In this figure, PSI Security is set up on all nodes except the X25router.

Figure 4–1 VAX PSI Configurations and Access to PSDNs



For Native and Multi-host nodes, a local DTE is associated directly with the Native or Multi-host node, and provides direct access to a PSDN. The local DTE provides direct incoming and outgoing access to and from the PSDN for the Native or Multi-host node.

For Access nodes, a local DTE is associated with a Connector node via which the Access node has indirect access to a PSDN. The Connector node and local DTE provide indirect incoming and outgoing access to and from the PSDN for the Access node. The Connector node can be a Multi-host node running VAX PSI or an X25router.

4.5 The Concepts Behind PSI Security

This section introduces the concepts behind PSI Security. See the appropriate sections of Chapter 5 for more information about the concepts described.

4.5.1 Agents, Objects and Security Controls

PSI Security uses the concept of agents that act on objects. Agents have associated rights, while objects are protected by access lists. There are several different types of agent and object. Each type has an associated security database.

Agent databases contain a list of agents and their rights identifiers. Object databases contain a list of objects and an Access Control List (ACL) for each object. The concepts of rights identifiers and Access Control Lists as security mechanisms are also used by VMS Security.

Rights identifiers are defined by you, and are used by PSI Security as keys to objects. You protect the access to an object using rights identifiers which you grant to agents selectively. You can think of rights identifiers as labels which you can define to specify groups of agents, or to describe the processes or access actions that you want to associated with a particular agent or group of agents.

ACLs are created by you, and are used by PSI Security to determine the type of access permitted to specified objects. An ACL contains the rights identifiers and access actions associated with a particular object. Access actions are defined in PSI Security, and determine the type of access and characteristics of a call.

You can think of an *agent* as a person with a key (the *rights identifier*), who wants to open (the *access action*) a door (the *object*). If the person has the correct key, he can open the door. In this example, the door lock acts as a security mechanism to determine whether the key will open the door. PSI Security attempts to match the rights identifiers specified in the agent and object databases to check that an agent is allowed access to the specified objects. In other words, an agent can gain access to an object if it has the correct rights identifiers and associated access actions.

See Table 4–1 for examples of agents and objects for incoming and outgoing calls.

Table 4–1 Summary of Agents and Objects for Incoming and Outgoing Calls

Node	Type of Call	Agent	Objects
NATIVE	Incoming	REMOTE DTE	LOCAL DTE and DESTINATION
	Outgoing	USER (or PROCESS)	REMOTE DTE and LOCAL DTE
MULTI-HOST	Incoming	REMOTE DTE	LOCAL DTE and DESTINATION
	Outgoing	USER (or PROCESS) or ACCESS NODE	REMOTE DTE and LOCAL DTE
ACCESS	Incoming	REMOTE DTE	LOCAL DTE and DESTINATION
	Outgoing	USER (or PROCESS)	REMOTE DTE

NOTE

For outgoing calls, PSI Security on an Access node only checks access to a remote DTE. No Local DTE check is made for outgoing calls.

See Section 4.5.2 and Chapter 5 for more information about rights identifiers and ACLs, and how they are used by PSI Security.

4.5.2 Rights Identifiers

Rights identifiers in PSI Security are names and attributes that you can create to protect the objects associated with your system. You grant rights identifiers to specific groups of agents, or individual agents, to control the way they make use of these objects.

You can use rights identifiers to:

- Specify groups of agents or individual agents
- Specify the access rights associated with a group of agents, or an individual agent, to objects on your system.

To protect objects, you create rights identifiers, and specify these rights identifiers with associated access actions in an Access Control List (ACL) for each object. Thus, by creating an ACL for an object, containing rights identifiers and associated access actions, you can allow access to the object by granting these rights identifiers to groups or individual agents. For more information about ACLs, see Section 4.5.3.

Note that it is usual to create a rights identifier to protect an individual object. This rights identifier is specified with associated access actions in an ACL entry (ACE) created to protect that object. By granting the rights identifier to a group of agents, each member of the group will have the same access to the object.

For example, a single rights identifier can be created to protect a local DTE. An ACL can be created containing this rights identifier and associated access actions, to define the access to the local DTE. By granting the rights identifier to a particular group of agents, each individual member of the group can access the local DTE, according to the access actions specified in the ACL.

Rights identifiers can also be created to allow access to an object for an individual agent. However, where possible, rights identifiers should be granted to groups of agents rather than individual agents, so that fewer rights identifiers are needed to implement PSI Security.

See Section 5.1 for more information about rights identifiers.

4.5.3 Access Control Lists (ACLs) and Access Control List Entries (ACEs)

Access Control Lists (ACLs) are lists containing the rights identifiers and access actions associated with objects on your system. As you may need to create several rights identifiers to specify the different access actions you want to associate with an object, you can create a list with several entries.

Each entry in the list is called an Access Control List Entry, or ACE. An ACE defines the access actions that are to be associated with an object for agents possessing the rights identifier named in the ACE. Note that an ACE controls the access to an object for an agent, based on the agent's rights identifiers. An agent can gain access to an object only if you have granted it the rights identifiers specified in the ACL for the object.

See Section 5.3 for information about ACL structure and the ACL matching algorithm.

4.5.4 The Agent Rights Databases

The Agent Rights Databases contain a list of agents and the rights identifiers held by those agents. An agent that holds a rights identifier also specified in an ACL for an object can gain access to that object depending on the access actions specified in the ACL.

An agent is the initiator of a call, and can be one of the following:

- For incoming calls - a **remote DTE**
- For outgoing calls - a **local user**, **process** or **Access node**

  An Access node can be an agent for outgoing calls via a Connector node.

Note that the Agent Rights Databases are volatile; each time your system is rebooted, the command procedure STARTPSI.COM activates PSI Security by running the command file PSI_SECURITY.COM to set up the PSI Security databases again.

There are three Agent Rights Databases used by PSI Security:

- The User Rights Database

- The Access Node Rights Database
- The Remote DTE Rights Database

When you create a rights identifier, it is added to the System Rights Database and copied into the appropriate Agent Rights Database. You then grant the rights identifier to the required agents in the appropriate Agent Rights Database.

Note that the System Rights Database is permanent; you must remove a rights identifier from the System Rights Database using either the VMS AUTHORIZE or PSIAUTHORIZE utility to remove it from your system.

The Agent Rights Databases are illustrated in Figure 4–2. See Chapters 5 and 6 for more information about how the Agent Rights Databases work and how to set up and manage these databases for your system.

The User Rights Database

The User Rights Database contains a list of the users defined on your system, and the rights identifiers held by those users for outgoing calls. All nodes have a User Rights Database.

Note that on each node, the list of users in the User Rights Database is the same list as defined in the System Rights Database. You can set up security to allow outgoing access from your system for specified users only, and to stop outgoing calls from unspecified users. PSI Security obtains the user information it requires from the System Rights Database. Processes that want to call remote DTEs use the rights identifiers held by the user who owns the process.

For more information about the User Rights Database, see Section 5.5.1. For information about the commands used to set up and manage the User Rights Database, see Section 6.4.1.

The Remote DTE Rights Database

The Remote DTE Rights Database contains a list of remote DTEs and the rights identifiers granted to those remote DTEs for incoming calls. All nodes have a Remote DTE Rights Database.

Note that your node may have access to one or more PSDNs and therefore many remote DTEs. You can set up security to allow incoming access to your system only from known remote DTEs, and to stop incoming access from unknown remote DTEs.

For more information about the Remote DTE Rights Database, see Section 5.5.2. For information about the commands used to set up and manage the Remote DTE Rights Database, see Section 6.4.1.

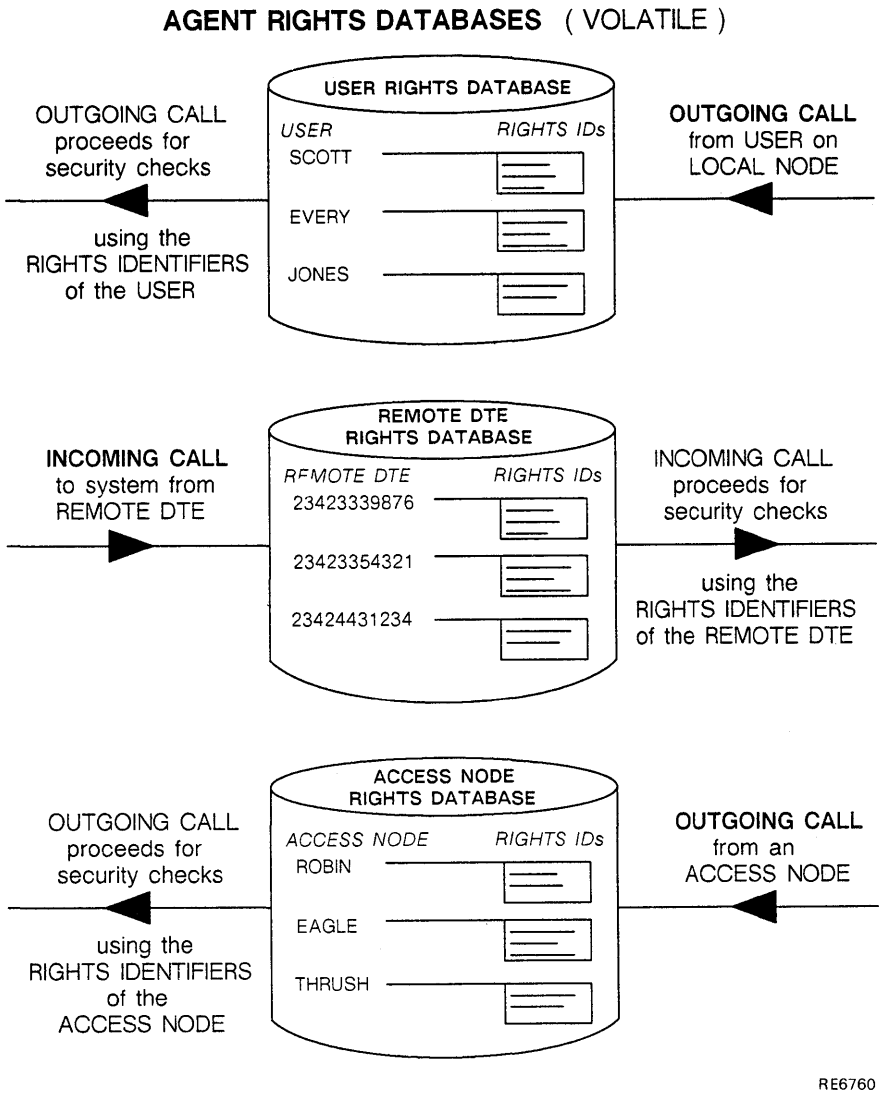
The Access Node Rights Database

The Access Node Rights Database contains a list of the Access nodes, and the rights identifiers associated with those Access nodes, that are allowed to make outgoing calls to remote DTEs via your Multi-host node. Only nodes operating in Multi-host mode have an Access Node Rights Database.

Note that if you set up PSI Security to allow an Access node to make outgoing calls via your Multi-host node, all users on that Access node can make outgoing calls. PSI Security must be set up on each Access node to define which users and processes on that node can make outgoing calls.

For more information about the Access Node Rights Database, see Section 5.5.3. For information about the commands used to set up and manage the Access Node Rights Database, see Section 6.4.1.

Figure 4–2 The Agent Rights Databases



4.5.5 The Object Access Control Databases

The Object Access Control Databases contain a list of objects and the ACLs that specify the rights identifiers and access actions associated with those objects.

Objects in PSI Security can be the following:

- For incoming calls - a **local DTE** and, optionally, an **X.25 destination**
- For outgoing calls - a **local DTE** and a **remote DTE**

You create ACLs to protect the objects on your system. Each ACL entry contains a rights identifier and associated access actions. If you grant a rights identifier that is contained in an object ACL entry to an agent, the agent can gain access to that object depending on the access actions specified in the entry.

There are three Object Access Control Databases used by PSI Security:

- The Local DTE Access Control Database
- The Remote DTE Access Control Database
- The Destination Access Control Database

Note that the Object Access Control Databases are volatile; each time your system is rebooted, the command procedure STARTPSI.COM activates PSI Security by running the command file PSI_SECURITY.COM to set up the PSI Security databases again.

The Object Access Control Databases are illustrated in Figure 4–3. See Chapters 5 and 6 for more information about how the Object Access Control Databases work, and how to set up and manage these databases for your system.

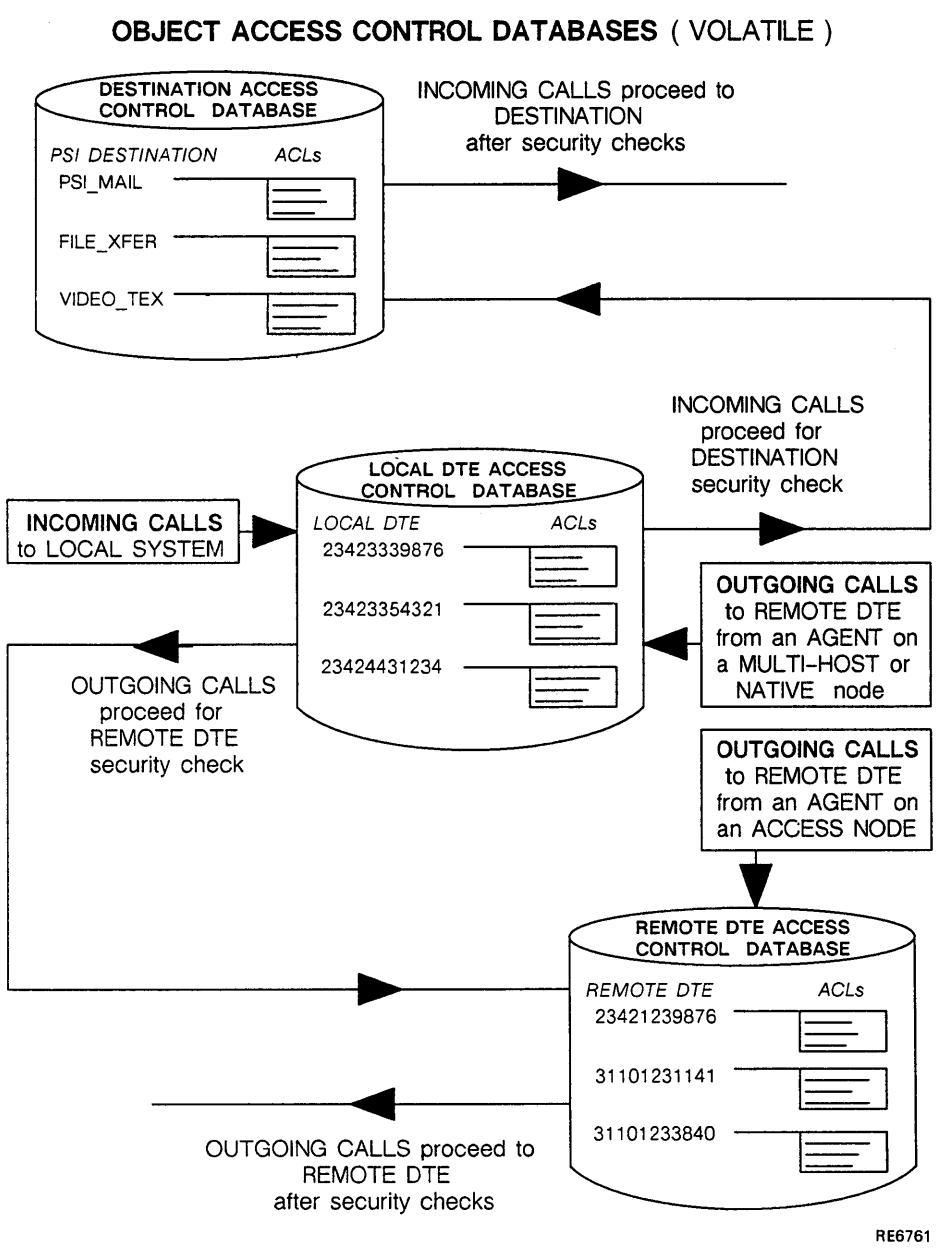
The Local DTE Access Control Database

The Local DTE Access Control Database contains a list of local DTEs and ACLs. The ACLs specify the rights identifiers and access actions associated with those local DTEs for incoming and outgoing calls. All nodes have a Local DTE Access Control Database. Note, however, that PSI Security on an Access node does not check outgoing access to a local DTE; it only checks outgoing access to a remote DTE. If you create an ACL entry for a local DTE for outgoing calls from your Access node, it has no effect.

For more information about how the different configurations of VAX PSI use the Local DTE Access Control Database, see Section 5.6.1.

Note that your node may have access to one or more local DTEs connected to one or more PSDNs. You can set up security controls for each local DTE. You can do this for each local DTE individually, or for all local DTEs collectively. See Sections 6.4.2.1 and 6.4.2.4 for information about the commands to use.

Figure 4–3 The Object Access Control Databases



The Remote DTE Access Control Database

The Remote DTE Access Control Database contains a list of remote DTEs and ACLs. The ACLs specify the rights identifiers and the access actions associated with those remote DTEs for outgoing calls. All nodes have a Remote DTE Access Control Database. For more information about how VAX PSI uses the Remote DTE Access Control Database for outgoing calls, see Section 5.6.2.

Note that your node may have the potential to access many remote DTEs via one or more PSDNs. You can set up security controls to allow outgoing access from your system only to known remote DTEs, and to stop outgoing access to unknown remote DTEs. See Section 6.4.2.2 for information about the commands to use.

The Destination Access Control Database

The Destination Access Control Database contains a list of destinations and ACLs. The ACLs specify the rights identifiers and access actions associated with these destinations for incoming calls (if a destination is provided). All nodes have a Destination Access Control Database.

The list of destinations in the Destination Access Control Database contains:

- A list of destinations that you have defined in the X25-SERVER database using NCP.
- A list of declared destinations that have been created by processes that have declared themselves as network processes by issuing IO$_ACPCONTROL QIOs.

See Sections 4.5.7 and 4.5.8 for more information about destinations and declared destinations.

For more information about how VAX PSI uses the Destination Access Control Database for incoming calls, see Section 5.6.3.

Note that security checks on destinations for incoming calls are optional. Although there must be an X25-SERVER destination for an incoming call, security does not have to be set up for the destination for the incoming call to proceed. You can set up security controls for destinations in addition to security controls for the local DTEs to which you have access. See Section 6.4.2.3 for information about the commands to use.

4.5.6 Remote DTE Trees

Remote DTE entries in the security databases are identified by their DTE address and are grouped according to the network with which they are associated. Each network has its own remote DTE tree structure, with entries that branch out from the top level entry that identifies the network. This top level entry is called the Data Network Identifier Code (DNIC). PSI Security searches the database for the remote DTE entry with an address that is the closest match to that specified.

See Section 5.4 for more information about remote DTE trees, and the remote DTE matching algorithm.

4.5.7 VAX PSI Destinations

You must ensure that the destination names you enter when using the PSI Security commands match the internal destination names in the X25-SERVER database. The X25-SERVER database defines the processes that are the destinations for incoming calls, so that incoming calls are directed to the appropriate process or object.

NOTE

1. PSI Security checks on destinations are optional.
2. Incoming calls which do not match a destination are cleared.
3. Only X.25 destinations are included in the X25-SERVER database; X.29 destinations cannot be protected individually.
4. Any new destinations created in the X25-SERVER database are unprotected.

The X25-SERVER database contains entries that identify the destinations on your system, and define the parameters that determine whether a destination can accept incoming calls from VAX PSI. The type of destination set up depends on the configuration of VAX PSI on your system. Note that destinations are defined in the X25-SERVER database using NCP commands.

When you configure a Multi-host node, you define Access nodes as destinations in the X25-SERVER database on your node using NCP. Usually, an incoming call to an Access node via a Multi-host supplies a destination subaddress in the incoming call packet. Alternatively, a matching mechanism recognizes an Access node as a destination using the call data supplied in the incoming call packet. The Multi-host uses this information to route the call to the appropriate Access node on your DECnet network.

See Section 2.3 for information about setting up destinations on your system for different configurations of VAX PSI.

4.5.8 VAX PSI Declared Destinations

Declared destinations can be specified and protected by ACLs in the Destination Access Control Database before a process declares itself a network process by issuing an IO$_ ACPCONTROL QIO. When the IO$_ACPCONTROL QIO is issued, the declared destination is created in the X25-SERVER database.

When issuing an IO$_ACPCONTROL QIO for a declared destination, for which you have already created an ACL, you should ensure that the destination name you specify in the PSI$C_NTD_NAME item field of the network process declaration block matches the unique name of the destination you created in the Destination Access Control Database.

If you create an ACL for a destination already created by an IO$_ACPCONTROL QIO, you should ensure that the destination name you use when creating the ACL matches the unique destination name you entered in the PSI$C_NTD_NAME item in the network process declaration block.

For more information about the IO$_ACPCONTROL QIO and network process declaration block, see the *VAX P.S.I. X.25 Programmer's Guide.*

4.5.9 Security Database CATCHALL Entries

A CATCHALL entry, if specified in a security database, is selected by PSI Security if a requested agent or object entry cannot be found. You can use a CATCHALL entry in the Remote DTE or Access Node Rights Databases to control the access to and from your system for certain agents. PSI Security selects an agent CATCHALL entry if it cannot find an entry for an agent in the appropriate Agent Rights Database.

NOTE

If an agent has no rights identifiers associated with it, PSI Security will select a CATCHALL entry, if there is one.

If an agent has no rights identifiers associated with it, and there is no CATCHALL entry, PSI Security will CLEAR calls from this agent.

You can use a CATCHALL entry in each of the Object Access Control Databases to control the access to objects on your system. In all cases, PSI Security selects an object CATCHALL entry if it cannot find an ACL for the specified object in an Object Access Control Database.

NOTE

If an object has no ACL associated with it, no rights identifier match is possible and PSI Security will select a CATCHALL entry, if there is one.

If an object has no ACL associated with it, and there is no CATCHALL entry, the object is unprotected. PSI Security will ALLOW calls to this object from known agents.

For more information about CATCHALL entries, see Section 5.3.2.

4.5.10 Protecting Multi-host Nodes From Access Nodes

You can use PSI Security to protect your Multi-host node from outgoing calls from users on Access nodes in your DECnet network. You do this by creating and granting rights identifiers to Access nodes in a similar way to which you create and grant rights identifiers to users on your system. The rights identifiers are contained in the Access Node Rights Database.

When an Access node attempts to make an outgoing call via your Multi-host node, PSI Security checks the Access Node Rights Database for the rights identifiers associated with that Access node. You can protect your Multi-host node from all unauthorized outgoing calls from each Access node in your DECnet network. If you do not protect your Multi-host node, all outgoing calls from users on the Access node, via your Multi-host node, are allowed.

NOTE

For maximum security, you must ensure that PSI Security is set up on each Access node on your DECnet network, to control incoming and outgoing calls to and from that node.

4.6 PSI Security and MAIL

In a DECnet-VAX system, the configuration parameters and access control information for incoming MAIL calls are contained in the object database. The privilege required for users who want to send MAIL is NETMBX. This privilege allows users to assign a channel to the NET device, and is required to create a logical link to a remote node.

In VAX PSI, the PSI Security databases contain rights identifiers associated with users who are allowed to make outgoing X.25 calls.

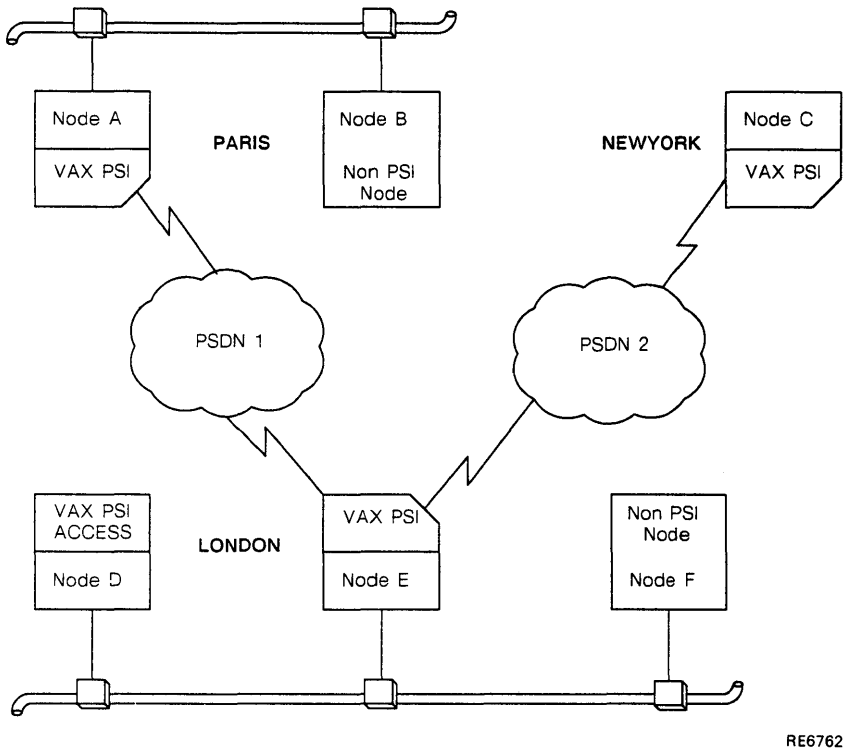
4.6.1 PSI Security and Poor Man's Routing

Using VAX PSI, local and remote users have the potential to use VMS MAIL to send information across a PSDN to a destination by specifying each node in the route from the source node to the destination node. This method of specifying routing information to send mail is known as Poor Man's Routing (PMR).

Indiscriminate use of PMR by local and other users to send and forward MAIL via your local DTEs across a PSDN can lead to an increase in network costs.

PSI Security includes a facility to restrict the use of PSDNs only to selected users, and to ensure that general users cannot use PMR to send mail across PSDNs. The following sections describe how PMR works, and show how you can use PSI Security to restrict the use of PMR to prevent local and other users abusing your local DTEs and access to PSDNs.

Figure 4–4 PSI Security and MAIL



4.6.2 Preventing Poor Man's Routing

4.6.2.1 Preventing Indiscriminate Use of VMS MAIL

In Figure 4–4, users on non-PSI node B and PSI Multi-host node A in Paris can send and receive VMS MAIL to and from other nodes on their local DECnet. Similarly, users on Access node D and Multi-host node E in London can send and receive VMS MAIL to and from other nodes on their local DECnet.

However, a user on non-PSI node B in Paris can also send VMS MAIL across PSDN 1 to Multi-host node E in London (or any other node on that DECnet) using PMR and PSI MAIL. For example, by specifying a command of the form:

```
MAIL> SEND TEXT.TXT
To:     A::PSI%network.remote_dte::E::user-name
```

where *network* is the network name and *remote_dte* is the remote DTE address of node E.

Note that in this case, node A is charged for the X.25 call.

Normally, you grant the rights identifier PSI$X25_USER to the processes and users who are allowed to make outgoing X.25 calls. This rights identifier ensures that general users cannot make outgoing X.25 calls.

However, for users on non-PSI nodes in your network, PMR using your Multi-host node works because outgoing X.25 calls use the DECnet MAIL object's username.

To prevent the indiscriminate use of poor man's routing by users on non-PSI nodes in your network, **do not** grant the DECnet process the PSI$X25_USER rights identifier.

To restrict the use of poor man's routing to known remote DTEs only, use PSI Security to allow MAIL restricted access to the known remote DTEs. You can do this by specifying MAIL as a local user, with rights identifiers that allow restricted outgoing access to known remote DTEs only.

4.6.2.2 Preventing Indiscriminate Use of PSI MAIL

Incoming PSI MAIL to Non-PSI Nodes

In Figure 4–4, users on PSI Multi-host node A in Paris could send PSI MAIL using PMR to remote nodes across PSDN 1. They could send PSI MAIL to a user on non-PSI node F in London by specifying each node in the route from their node to the destination node. For example, they could specify a command of the form:

```
MAIL> SEND TEXT.TXT
TO:     PSI%network.remote_dte::E::F::user-name
```

to send PSI MAIL to a user on node F, where *network* is the network name and *remote_dte* is the remote DTE address of node E. In this case, node A is charged for the X.25 call.

To prevent incoming PSI MAIL to non-PSI nodes on your network

PSI Security performs local DTE and destination checks on your Multi-host node for incoming PSI MAIL calls before MAIL runs. However, your Multi-host node requires a DECnet logical link to forward MAIL to the non-PSI nodes. This DECnet logical link requires NETMBX privilege.

To prevent remote DTEs using PMR to send MAIL to non-PSI nodes in your network:

- Create an account in which to run PSI MAIL. This should be a different account to the DECnet MAIL object account.
- Grant the username associated with the PSI_MAIL object the PSI$X25_USER rights identifier.
- **Do not** grant the username associated with the PSI_MAIL object NETMBX privilege.

Note that this prevents all redirection of incoming PSI MAIL to non-PSI nodes on your network.

Forwarding PSI MAIL Calls via Multi-host Nodes

In Figure 4–4, users on PSI node A in Paris could use PSI node E in London to forward PSI MAIL to PSI node C in New York, using a command of the form:

```
MAIL> SEND TEXT.TXT
To: PSI%network1.remote_dte1::PSI%network2.remote_dte2::C::user-name
```

where *network1* and *network2* are the network names, and *remote_dte1* and *remote_dte2* are the remote DTE addresses of node E. In this case, node A is charged only for the X.25 call to node E. However, node E is charged for the X.25 call to node C.

Indiscriminate use of your node by users on remote nodes to forward PSI MAIL could lead to large network costs for your system. In addition, it may be illegal to inadvertently forward third party traffic via your system.

To prevent remote nodes using your node to forward third party traffic

You can prevent your node from forwarding third party traffic by setting up PSI Security and privileges for incoming and outgoing calls:

- Create an account in which to run PSI MAIL. This should be a different account to the DECnet MAIL object account.
- Set up PSI Security to restrict incoming PSI MAIL from known remote DTEs only. You can do this by creating rights identifiers and ACLs for incoming calls to the PSI_MAIL destination.
- Set up PSI Security to prevent outgoing PSI MAIL that uses the username associated with the PSI_MAIL object
- Grant the username associated with the PSI_MAIL object the PSI$X25_USER rights identifier.
- **Do not** grant the username associated with the PSI_MAIL object NETMBX privilege.

In this way, you can prevent remote nodes from using your node to forward third party traffic.

4.7 PSI Security and Incoming X.29 Calls

The X25-SERVER database defines the destinations for incoming calls to VAX PSI. If you have enabled the X29-SERVER module, VAX PSI sets up the X29-SERVER destination to handle incoming X.29 calls. Associated with this destination is the X29-SERVER database, containing the destinations for incoming X.29 calls.

- An X.29 destination with an associated object has an object record specifying the command procedure that runs when the incoming X.29 call is accepted, and the VMS username under which this happens.
- An X.29 destination without an associated object is passed to the VMS login sequence. In such cases, your system is protected by VMS Security.

You should ensure that X29-SERVER destinations have objects associated with them. In this way, incoming X.29 calls are either passed to the appropriate object, such as an application or process, or to the VMS login sequence. Note that a self declared X.29 network process is routed to an X.29 program.

You should also ensure that X29-SERVER applications and processes have adequate security facilities, such as application usernames and passwords.

4.7.1 Protecting the X29_SERVER Destination

You can set up PSI Security to protect the X29_SERVER destination in the X25-SERVER database on your system by granting incoming X.29 access only to known remote terminals.

Note that for remote terminal users who use a PSDN PAD, you can allow incoming X.29 calls to your system by granting incoming X.29 access to the PSDN PAD. However, all users who have access to that PSDN PAD may have the ability to make incoming X.29 calls to your system.

4.7.1.1 Using VMS Security to Protect Your System Against Incoming X.29 Calls

For all incoming X.29 access to your system, you should ensure that your system is protected adequately by VMS Security. In particular, you should make full use of the VMS Security alarms and auditing facilities to detect dialup interactive logins, login failures breakin attempts and other access to your system. You can enable VMS Security alarms to send messages to the security terminal whenever such events are detected.

The following are examples of VMS Security alarm messages relevant to X.29 access. In these examples, remote terminal user GREEN is attempting to login to node ROBIN. The VMS Security alarms on node ROBIN have been set up to indicate login failures. In these examples, the remote terminal is connected to remote DTE 23428881234 in the PSS network.

```
%%%%%%%%%%  OPCOM  18-APR-1988 14:17:05.90  %%%%%%%%%%
Security alarm on ROBIN / Dialup interactive login failure
 Time:  18-APR-1988 14:17:05.88
 PID:  00008053
 User Name: GREEN
 Dev Name: _VTA5367  (PSS.23428881234)

%%%%%%%%%%  OPCOM  18-APR-1988 14:17:25.21  %%%%%%%%%%
Security alarm on ROBIN / Dialup interactive login failure
 Time:  18-APR-1988 14:17:25.20
 PID:  00008054
 User Name: GREEN
 Status:  %LOGIN-F-INVPWD, invalid password
 Dev Name: _VTA5368  (PSS.23428881234)

%%%%%%%%%%  OPCOM  18-APR-1988 14:17:33.85  %%%%%%%%%%
Security alarm on ROBIN / Dialup interactive login failure
 Time:  18-APR-1988 14:17:33.83
 PID:  00008054
 User Name: <login>
 Status:  %LOGIN-F-NOSUCHUSER, no such user
 Dev Name: _VTA5368  (PSS.23428881234)

%%%%%%%%%%  OPCOM  18-APR-1988 14:18:49.48  %%%%%%%%%%
Security alarm on ROBIN / Dialup interactive breakin detection
 Time:  18-APR-1988 14:18:49.21
 PID:  00008057
 User Name: GREEN
 Password: ABRACADABRA
 Dev Name: _VTA5370  (PSS.23428881234)

%%%%%%%%%%  OPCOM  18-APR-1988 14:18:49.89  %%%%%%%%%%
Security alarm on ROBIN / Dialup interactive login failure
 Time:  18-APR-1988 14:18:49.87
 PID:  00008057
 User Name: GREEN
 Status:  %LOGIN-F-INVPWD, invalid password
 Dev Name: _VTA5370  (PSS.23428881234)
```

Using the DCL `SHOW INTRUSION` command, you can obtain details of the login and breakin attempts. For example:

```
$ SHOW INTRUSION
Intrusion        Type        Count  Expiration   Source
   TERM_USER     SUSPECT         5  14:42:24.94  PSS.23428881234:GREEN
   TERMINAL      SUSPECT         1  14:22:33.82  PSS.23428881234:

$ SHOW INTRUSION
Intrusion        Type        Count  Expiration   Source
   TERM_USER     INTRUDER        6  14:25:25.36  PSS.23428881234:GREEN
   TERMINAL      SUSPECT         1  14:22:33.82  PSS.23428881234:
```

See the *Guide to VMS Security* for more information about the use and features of VMS Security.

4.8 Incoming Calls to Your System

Incoming calls to your system are checked by PSI Security to see if you have allowed incoming access for calls from specified remote DTEs.

The following sections show you the actions that allow specified remote DTEs to make incoming calls to your system. Refer to Chapter 6 for details and examples of the commands to use.

Note that in each of the following sections, two rights identifiers are created:

- A rights identifier to protect the local DTE(s) associated with incoming calls to your system
- A rights identifier to protect the destination(s) on your system

If you create a rights identifier for each local DTE and for each destination, the security databases become complex. However, this method provides a high level of security for incoming calls to your system when set up correctly.

You may decide to create rights identifiers that can be used to protect one or more local DTEs and/or one or more destinations against incoming calls to your system. Alternatively, you may decide to provide incoming protection for local DTEs only. These options reduce the number of rights identifiers required for your system and the complexity of the security databases. However, they do not provide the highest level of security for incoming calls to your system.

4.8.1 Incoming Calls to a Native or Multi-host Node

If your system is a Native or Multi-host node, and you want to allow a destination on your node to receive incoming calls from specific remote DTEs:

1. Create a rights identifier and add it to the System Rights Database, to protect your local DTE(s).

   Create a rights identifier and add it to the System Rights Database, to protect the destination on your node.

2. Create an entry for the local DTE(s) in the Local DTE Access Control Database, with an ACL specifying the local DTE rights identifier and incoming access.

   Note that your node may have one or more local DTEs to provide access to one or more PSDNs. You can create a separate rights identifier to protect each local DTE.

3. Create an entry for the destination in the Destination Access Control Database, with an ACL specifying the destination rights identifier and incoming access.

   Note that you can create a separate rights identifier to protect each destination on your node.

4. Grant the rights identifiers you have created to each remote DTE in the Remote DTE Rights Database that is allowed to make incoming calls to your node.

These actions allow incoming calls from the specified remote DTEs via the specified local DTE(s) to the specified destination on your Native or Multi-host node.

4.8.2 Incoming Calls via a Multi-host Node to an Access Node

If your node is a Multi-host node, and you want to allow an Access node to receive incoming calls from specific remote DTEs via your Multi-host node:

1. Create a rights identifier and add it to the System Rights Database, to protect your local DTE(s).

   Create a rights identifier and add it to the System Rights Database, to protect the Access node defined as a destination on your Multi-host node.

2. Create an entry for each local DTE in the Local DTE Access Control Database, with an ACL specifying the rights identifier and incoming access.

   Note that your Multi-host node can have one or more local DTEs to provide access to one or more PSDNs. You can create a separate rights identifier to protect each local DTE.

3. Create an entry for the Access node in the Destination Access Control Database, with an ACL specifying the destination rights identifier and incoming access.

4. Grant these rights identifiers to each remote DTE in the Remote DTE rights Database that is allowed to make incoming calls to the Access node via your Multi-host node.

These actions allow the specified remote DTEs to make incoming calls to an Access node, via your Multi-host node and the specified local DTE(s).

Note that PSI Security on an Access node can clear an incoming call even though the incoming call has been allowed to proceed by PSI Security on your Multi-host node.

You can create rights identifiers to protect each Access node defined as a destination on your Multi-host node. This security control is optional if you are satisfied that PSI Security is set up correctly on each Access node. If there is little or no security on the Access nodes in your DECnet network, you must set up security on your Multi-host node to protect the Access nodes from unauthorized incoming calls.

Generally, you need only provide the same level of protection for each Access node connected to your Multi-host node. However, you can create different rights identifiers to provide different levels of protection. For example, you may want to allow all incoming calls to some Access nodes, but only allow certain incoming calls to other Access nodes.

4.8.3 Incoming Calls to an Access Node (via a Connector Node)

An Access node receives incoming calls via a Connector node and associated local DTEs. The Connector node can be a VAX PSI Multi-host node or an X25router. If the Connector node is a VAX PSI Multi-host node, you must ensure that PSI Security is set up on the Multi-host node to allow incoming calls to your Access node. Note that the X25router does not support PSI Security. See the X25router documentation for more information.

If your node is an Access node and you want to allow a destination on your Access node to receive incoming calls from specific remote DTEs:

1. Create a rights identifier and add it to the System Rights Database, to protect the local DTE associated with the Connector node that handles the incoming calls.

   Note that local DTE checks for incoming calls to your Access node are optional. If you do not mind which Connector node handles the incoming calls to your Access node, you can omit this part of step 1 and all of step 2.

   Create a rights identifier and add it to the System Rights Database, to protect the destination on your Access node.

2. Create an entry for the local DTE associated with the Connector node in the Local DTE Access Control Database, with an ACL specifying the local DTE rights identifier and incoming access.

   Note that by specifying the local DTE on your Access node, you are specifying which Connector node handles the incoming calls to your Access node.

3. Create an entry for the destination in the Destination Access Control Database, with an ACL specifying the destination rights identifier and incoming access.

   Note that you can create a separate rights identifier to protect each destination on your Access node.

4. Grant the rights identifiers to each remote DTE that is allowed to make incoming calls to your Access node. These actions create an entry with these rights identifier for each of the remote DTEs in the Remote DTE Rights Database.

These actions allow incoming calls to the specified destination on your Access node from the specified remote DTE(s), via the Connector node with access to the specified local DTE.

4.8.4 Incoming Calls to a Combination Node

If your node is a Combination node, it has both VAX PSI and VAX PSI Access installed and operates in both Multi-host and Access modes.

To set up PSI Security for incoming calls to destinations on your Combination node and other nodes, you must set up the security databases for both the Multi-host and Access parts of your node.

For example, you must set up the Multi-host part of your Combination node so that specified remote DTEs can make incoming calls via the local DTEs connected to your Combination node to destinations on your node and to Access nodes in your DECnet network. You must set up the Access part of your Combination node also, so that remote DTEs can make incoming calls to destinations on your node, via the local DTEs connected to other Connector nodes in your DECnet network.

See the above sections to determine the actions required to set up PSI Security for incoming calls to destinations on the Access and Multi-host parts of your Combination node.

4.8.5 Typical Restrictions on Incoming Calls

Incoming Calls from Remote DTEs

Using PSI Security, you can restrict calls from remote DTEs to your system. For example, you could restrict incoming calls to those from certain remote DTEs in the United Kingdom public packet switching system (PSS). You can do this by protecting your local DTE against all incoming calls other than those from within PSS, and by granting incoming access to specified DTEs within PSS only. Any call from an unspecified DTE within PSS, or any DTE outside PSS would be cleared on trying to access your system.

Incoming Calls to Destinations

You could restrict incoming calls from certain remote DTEs to specific processes on your system. For example, you could protect your local DTE against all incoming calls from remote DTEs outside PSS, and grant all remote DTEs within PSS incoming access to a specific user application only. Any call from outside PSS, or to a process other than the specified application, would be cleared on trying to access your system.

Incoming Reverse Charge Calls

You can specify which remote DTEs are allowed to make incoming reverse charge calls. Usually, all calls are charged to the agent making the call. Unless a remote DTE is specifically granted the right to make reverse charge calls, PSI Security clears incoming reverse charge calls. See Section 5.2 for more information about call charging.

4.9 Outgoing Calls from Your System

Outgoing calls from your system are checked by PSI Security to see if you have allowed users outgoing access via the specified local DTE to the specified remote DTE.

The following sections show you the actions that allow users on your system to make outgoing calls to remote DTEs. Refer to Chapter 6 for details and examples of the commands to use.

Note that in each of the following sections, two rights identifiers are created:

- A rights identifier to protect the local DTE(s) associated with outgoing calls from your system
- A rights identifier to allow outgoing access to specific remote DTE(s)

If you create a rights identifier for each local DTE and for each group of remote DTEs, the security databases become very complex. However, this method provides a high level of security for incoming calls to your system.

You may decide to create rights identifiers that can be used to protect one or more local DTEs against outgoing calls to one or more remote DTEs. Alternatively, you may decide to protect local DTEs only against outgoing calls. These options reduce the number of rights identifiers required for your system and the complexity of the security databases. However, they do not provide the highest level of security for outgoing calls from your system.

4.9.1 Outgoing Calls from a Native or Multi-host Node

If your node is a Native or Multi-host node and you want to allow certain users on your node to make outgoing calls to specific remote DTEs:

1. Create a rights identifier and add it to the System Rights Database on your node to protect your local DTE(s).

   Create a rights identifier and add it to the System Rights Database on your node to allow outgoing access to the remote DTE(s).

2. Create an entry for your local DTE(s) in the Local DTE Access Rights Database, with an ACL specifying the local DTE rights identifier and outgoing access.

   Note that your Native or Multi-host node can have one or more local DTEs to provide access to one or more PSDNs. You can provide protection for each local DTE separately by creating a rights identifier for each one.

3. Create an entry for each specific remote DTE in the Remote DTE Access Control the Database, with an ACL specifying the remote DTE rights identifier and outgoing access.

4. Grant the rights identifiers to each user in the User Rights Database who is allowed to make outgoing calls to the specified remote DTE(s).

These actions allow the specified users on your Native or Multi-host node to make outgoing calls to the remote DTEs you have specified, via the local DTE(s) you have specified.

4.9.2 Outgoing Calls via a Multi-host Node from an Access Node

If your node is a Multi-host node, and you want to allow an Access node to make outgoing calls to specific remote DTEs:

1. Create a rights identifier and add it to the System Rights Database on your Multi-host node to protect your local DTE from outgoing calls from users on Access nodes.

   Create a rights identifier and add it to the System Rights Database on your Multi-host node to allow outgoing access to the remote DTE(s).

2. Create an entry for your local DTE in the Local DTE Access Rights Database, with an ACL specifying the local DTE rights identifier and outgoing access.

   Note that your Multi-host node can have one or more local DTEs to provide access to one or more PSDNs. You can provide protection for each local DTE separately by creating an identifier for each one.

3. Create an entry for each specific remote DTE in the Remote DTE Access Control Database, with an ACL specifying the remote DTE rights identifier and outgoing access.

4. Grant the rights identifiers to each Access node in the Access Node Rights Database that is allowed to make outgoing calls.

These actions allow any user on the Access node to make outgoing calls to the remote DTE(s) specified, via your Multi-host node and the specified local DTE(s). Note that security must be set up on the Access node to prevent all users on that node from making outgoing calls at will to the specified remote DTEs via your Multi-host node. On your Multi-host node, you can only specify the Access nodes that can make outgoing calls, and not the users on those nodes.

4.9.3 Outgoing Calls from an Access Node (via a Connector Node)

An Access node makes outgoing calls via a Connector node and associated local DTEs. The Connector node can be a VAX PSI Multi-host node or an X25router. If the Connector node is a VAX PSI Multi-host node, PSI Security must be set up on the Multi-host node to allow outgoing calls from your Access node. Note that the X25router does not support PSI Security. See the X25router documentation for more information.

If your node is an Access node, and you want to allow certain users on your Access node to make outgoing calls to specific remote DTEs:

1. Create a rights identifier and add it to the System Rights Database on your Access node to allow outgoing access to the remote DTE(s).

2. Create an entry for each specific remote DTE in the Remote DTE Access Control Database, with an ACL specifying the remote DTE rights identifier and outgoing access.
3. Grant the rights identifier to each user in the User Rights Database who is allowed to make outgoing calls to the specified remote DTEs.

These actions allow the specified users on your Access node to make outgoing calls to the remote DTE(s) specified. Note that PSI Security on an Access node does not check outgoing access to a local DTE; it only checks outgoing access to a remote DTE. If you create an ACL entry for a local DTE associated with a Connector node for outgoing calls from your Access node, it has no effect.

4.9.4 Outgoing Calls from a Combination Node

If your node is a Combination node, it has both VAX PSI and VAX PSI Access installed.

To set up PSI Security for outgoing calls to remote DTEs from your Combination node and other nodes, you must set up the security databases for both the Multi-host and Access parts of your node.

For example, you must set up the Multi-host part of your Combination node so that users on your node and other Access nodes on your DECnet network can make outgoing calls to specific remote DTEs via the local DTEs connected to your node. You must also set up the Access part of your Combination node, so that users on your node can make outgoing calls to specific remote DTEs, via the local DTEs connected to other Connector nodes in your DECnet network.

See the sections described above to determine the actions required to set up PSI Security for outgoing calls from users, via the Access and Multi-host parts of your Combination node.

4.9.5 Typical Restrictions on Outgoing Calls

Outgoing Calls from Users

Using PSI Security, you can restrict outgoing calls from specific users on your system to stop expensive outgoing calls. You can restrict outgoing calls to specific remote DTEs. You could, for example, restrict outgoing calls from a group of users on your system to remote DTEs within a specified network, for example, the United Kingdom network, PSS. Any outgoing calls from unspecified users, or to remote DTEs outside PSS, would be cleared.

Outgoing Calls from Access Nodes

If your system is a Multi-host node, you can restrict outgoing calls to remote DTEs from users on specific Access nodes. In a large system, this is a useful facility that allows you to reduce network operation costs. For example, you could restrict outgoing calls from all Access nodes, and only allow incoming calls. Any outgoing calls from Access nodes would be cleared by PSI Security on your Multi-host node.

Outgoing Reverse Charge Calls

Note that unless you specify the users who are allowed to make outgoing reverse charge calls to specific remote DTEs, you are charged for all outgoing calls originating from your system.

5
How PSI Security Works

This chapter describes in more detail the concepts of PSI Security introduced in Chapter 4. In particular, it describes the security databases and the security mechanisms that control the use of PSI Security. It also includes examples to show how PSI Security works.

The chapter also explains the overall checking procedure for dealing with incoming and outgoing calls. Following this are some descriptions of how you can use the security system. (For more detailed examples, including the commands to set up the databases, see Chapter 6.)

5.1 Identifiers

PSI Security uses the following types of identifier:

- Rights identifiers that you define for your system, to specify the rights associated with groups or individual agents. Rights identifiers are general identifiers and are used with access actions by PSI Security to control incoming and outgoing calls.
- PSI Security specific identifiers. These rights identifiers are used internally by PSI Security to define which users and processes can access VAX PSI by issuing IO$_ ACCESS and IO$_ACPCONTROL QIO calls.

See the *Guide to VMS System Security* for more information about identifiers, in particular system defined identifiers and the System Rights Database.

5.1.1 Rights Identifiers

Rights identifiers are general identifiers consisting of alphanumeric strings of between 1 and 31 characters.

You create rights identifiers to specify the rights of a group of agents or an individual agent. Note that a "right", defined as a rights identifier, is a label of convenience you give to an agent so that you can specify the access to objects you wish to associate with that agent.

As an example, you could create the rights identifier PAYROLL, to specify the payroll department in your company, or PSI_MAIL to specify the PSI MAIL utility. You could create the rights identifier PSI_OUTGOING to specify that an agent in possession of that right can attempt to make outgoing calls. You could also, for example, create the rights identifier OUTGOING_NO_CHARGE to specify that an agent in possession of that right can attempt to make an outgoing reverse charge call.

You can use the UIC identifiers defined in the System Rights Database for the users and processes on your system. UIC identifiers are presented in numeric or alphanumeric format. For example, [219,387], or [PAYROLL,JONES]. Note that for a process, VAX PSI uses the UIC identifier for the user who initiated the process. See the *Guide to VMS System Security* for more information about UIC identifiers.

NOTE

1. If no rights identifiers have been created for an agent, either as a member of a group or as an individual, PSI Security will clear incoming or outgoing calls from that agent.
2. A call can only proceed if you have specified the correct access actions in the ACL for the object. Take care not to confuse the name of the rights identifiers that you define with the access actions defined in PSI Security.

5.1.2 PSI Security Specific Identifiers

PSI Security uses two optional rights identifiers to provide basic security controls for systems on which the PSI Security databases have not been set up, and which therefore do not have a full implementation of security. These identifiers are:

PSI$X25_USER — the rights identifier that you must grant to any user or process that is allowed to access VAX PSI by issuing an IO$_ACCESS QIO.

PSI$DECLNAME — the rights identifier that you grant to any process that is allowed to declare itself a network process by issuing an IO$_ACPCONTROL QIO.

These rights identifiers must be defined in the System Rights Database using VMS AUTHORIZE before the VAX PSI software is loaded. If they are not defined before the software is loaded and are defined subsequently, they have no effect and users require NETMBX privilege only to send calls via VAX PSI.

For a user issuing an IO$_ACCESS QIO:

1. VAX PSI checks that the PSI$X25_USER identifier is defined. If the identifier is defined, and the user owns it, the user can make an IO$_ACCESS QIO to VAX PSI.
2. If PSI$X25_USER is defined but is not owned by the user, VAX PSI checks for BYPASS privilege. With BYPASS, the user can make an IO$_ACCESS QIO to VAX PSI. Without PSI$X25_USER or BYPASS, the user is nonprivileged and cannot make an IO$_ACCESS QIO.
3. If PSI$X25_USER is not defined, VAX PSI checks for NETMBX privilege. With NETMBX privilege, the user can make an IO$_ACCESS QIO; otherwise the user is nonprivileged.

Note that there is no check for BYPASS privilege if PSI$X25_USER is not defined.

For an IO$_ACPCONTROL QIO, the checks are similar:

1. VAX PSI checks that the PSI$DECLNAME identifier is defined. If the identifier is defined, and the process owns it, the process can make an IO$_ACPCONTROL QIO to VAX PSI to set up a network process.
2. If PSI$DECLNAME is defined but is not owned by the process, VAX PSI checks for BYPASS privilege. With BYPASS, the process can make an IO$_ACPCONTROL QIO. Without PSI$DECLNAME or BYPASS, the process is nonprivileged and cannot make an IO$_ACPCONTROL QIO.
3. If PSI$DECLNAME is not defined, VAX PSI checks for NETMBX privilege. With NETMBX, the process can make an IO$_ACPCONTROL QIO; otherwise the process is nonprivileged.

Note that there is no check for BYPASS privilege if PSI$DECLNAME is not defined.

5.2 Access Actions

Access actions are defined by VAX PSI, and specify the access to particular objects for individual agents or groups of agents. You grant access to an object by creating an ACL in which you specify the rights identifiers and access actions you want to associate with the object.

The following access actions can be specified in an ACL entry:

NONE	No access to VAX PSI.
INCOMING	Incoming non-reverse charge calls to VAX PSI.
INCOMING+REVERSE_CHARGE	All incoming calls to VAX PSI, including reverse charge calls.
OUTGOING	Outgoing reverse charge calls from VAX PSI.
OUTGOING+CHARGE	All outgoing calls from VAX PSI, including reverse charge calls.

The access actions INCOMING and OUTGOING imply that your system is NOT charged for any calls to or from objects with ACLs specifying those actions. Note that access actions are linked by the '+' character. Also, you can abbreviate access actions provided the abbreviation is not ambiguous.

For example, you could create an ACL for a remote DTE in which you include the rights identifier PAYROLL and specify the access action OUTGOING. An agent in possession of the rights identifier PAYROLL will be allowed to make an outgoing reverse charge call to the remote DTE because the ACL for the remote DTE includes the rights identifier PAYROLL and the access action OUTGOING.

5.3 Access Control Lists

This section describes how Access Control Lists (ACLs) work in PSI Security. See the *Guide to VMS System Security* for a full description of Access Control Lists.

5.3.1 ACL Structure

An ACL consists of ACL entries (ACEs) that grant or deny access to a particular system object. The following is an example of an ACL for a particular local DTE in the Local DTE Access Control Database:

```
(IDENTIFIER=THEY_PAY,ACCESS=INCOMING+OUTGOING)
(IDENTIFIER=WE_PAY,ACCESS=INCOMING+REVERSE_CHARGE+OUTGOING+CHARGE)
```

This ACL consists of two entries; each entry consists of a rights identifier followed by a set of access actions.

For the rights identifier THEY_PAY, the access actions are INCOMING and OUTGOING; that is, incoming non-reverse charge calls and outgoing reverse charge calls are allowed.

For the rights identifiers WE_PAY, the access actions are INCOMING+REVERSE_ CHARGE and OUTGOING+CHARGE; that is, any incoming or outgoing call, normal or reverse charge, is allowed.

Note that when you create an ACL for an object, you can only specify rights identifiers that you have created in the System Rights Database. An agent can only gain access to the object when you have granted it the appropriate rights identifiers.

5.3.2 CATCHALL Object Database Entries

Each Object Access Control Database can have a special entry known as a CATCHALL entry. If PSI Security fails to find an ACL for an object, it will select the CATCHALL entry, if one has been specified. The CATCHALL entry has an ACL associated with it.

You can use a CATCHALL entry to prevent unauthorized calls to or from your system. For example, you can create a CATCHALL entry in the Local DTE Access Control Database to deny or restrict access to or from your local DTEs collectively. The CATCHALL entry can contain a rights identifier or wildcard, and access actions specifying the type of access required. In this way, you can deny or restrict access to your local DTEs, depending on the access actions specified in the CATCHALL ACL.

Note that CATCHALLs can also be used in the Remote DTE Rights Database and the Access Node Rights Database. For example, you can grant a rights identifier to a CATCHALL entry in the Remote DTE Rights Database and specify this rights identifier and ACCESS=NONE in a CATCHALL entry in the Local DTE Access Control Database. Thus, unknown remote DTEs trying to make incoming calls to your system will be denied access.

5.3.3 Match-all ACL Entries

Note that match-all ACL entries of the form:

`IDENTIFIER= *, ACCESS= access-action`

are allowed, where * is a wildcard character.

Such an entry is usually the last ACL entry, in which the access specified is NONE. In this case, if PSI Security fails to match rights identifiers in entries at the top of an object ACL, it will select the match-all entry because a match will occur between the wild card and the first rights identifier in the agent's list.

Note that you can use a match-all for an object which requires only limited security. For example, you could create a match-all for the MAIL utility, allowing access to those agents to whom you have granted rights identifiers. In this case, if an agent has no rights identifiers at all, it cannot have access to the MAIL utility because it has no rights identifiers to match with the match-all wildcard.

5.3.4 The ACL Matching Algorithm

PSI Security only checks objects which have ACLs.

Incoming Calls

For incoming calls, PSI Security checks the following objects:

Native, Multi-host and Access nodes:

1. A **local DTE**
2. A **destination**

The first check must be successful for the call to proceed. Note that the second check is optional. If PSI Security has not been set up for a destination, an incoming call will proceed to that destination if the local DTE security check is successful.

Outgoing Calls

For outgoing calls, PSI Security checks the following objects:

Native and Multi-host nodes:

1. A **local DTE**
2. A **remote DTE**

Both checks must be successful for the outgoing call to proceed.

Access nodes:

1. A **remote DTE** check only.

   This check must be successful for the outgoing call to proceed. Note that PSI Security does not perform a local DTE check for outgoing calls from an Access node.

Object ACL Checks

When an agent attempts to make an incoming or outgoing call, PSI Security obtains the rights identifiers for the agent from the appropriate Agent Rights Database.

When PSI Security has obtained the rights identifiers for the agent, it applies the first object check. If this check is successful, it applies the second object check (if appropriate).

The object checks follow the same algorithm. When applying an object check, PSI Security:

1. Selects the ACL for the object from the appropriate Object Access Control Database.
2. Attempts to match one of the agent's rights identifiers with a rights identifier from an ACL entry for the object.

The algorithm selects the first match it finds. If there is no match, the call is cleared.

If an agent has rights identifiers but there is no ACL for the object, the algorithm selects a CATCHALL entry, if there is one. If there is no rights identifier match with the CATCHALL entry, the call is cleared.

If an agent has no rights identifiers but there is an ACL for the object or a CATCHALL entry, the call is cleared.

NOTE

If an agent has rights identifiers but there is no ACL for the object and no CATCHALL entry, the call is **ALLOWED**.

If an agent has no rights identifiers, the object has no ACL and there is no CATCHALL entry, the call is **ALLOWED**.

If the first object check is successful, PSI Security allows the call to proceed to the second object check (where appropriate).

If the second object check is successful, PSI Security allows the call to proceed to or from your system, according to the access actions specified in the selected ACL entry for the second object.

Note that for incoming calls, provided that the local DTE check is successful, the destination check is optional. If no security is set up for a destination, the incoming call is allowed to proceed according to the access actions specified in the selected ACL entry for the local DTE.

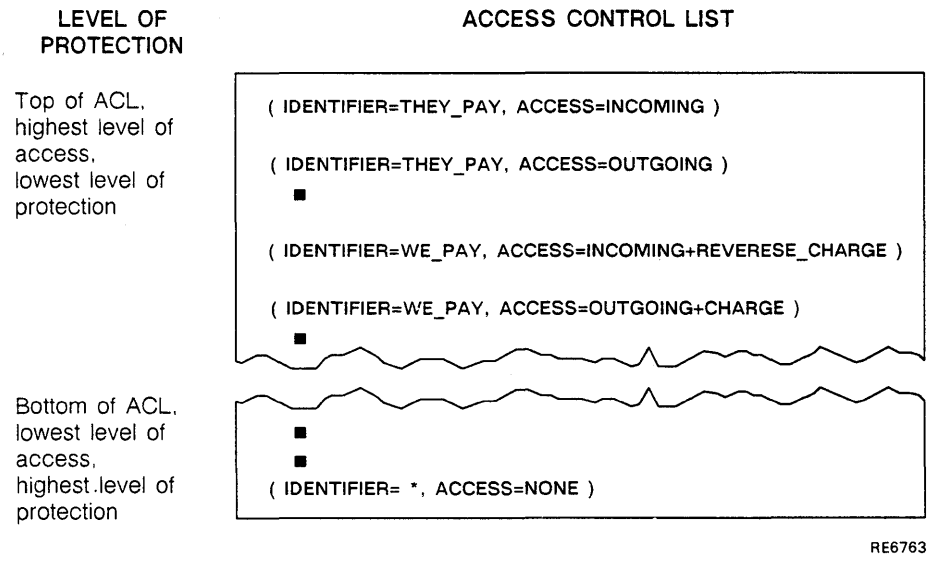
5.3.5 The Order of ACL Entries

The order of entries in an ACL is important. The PSI Security ACL matching algorithm searches each entry in turn, from the first entry at the top of the ACL, for the first rights identifier match it can find. When it finds a match, it stops searching. If a match occurs in an entry further down the ACL, it has no effect. The highest placed entry is always selected in preference to those below it when a match occurs.

ACL entries for objects that require several levels of protection must be entered carefully. You must take care to position the entries, so that greater protection occurs nearer the end of the ACL for the desired match. You can use the /AFTER=*acl-entry* qualifier to ensure that you enter ACL entries in the desired order. Note that the ACL entry specified in the /AFTER qualifier must already exist.

For example, for incoming calls from remote DTEs to a local DTE connected to your system, you may decide to apply greater protection against incoming reverse charge calls than against incoming non-reverse charge calls. In this case, you would create ACL entries in the Local DTE Access Control Database in the order shown in Figure 5–1.

Figure 5–1 The Ordering of ACL Entries

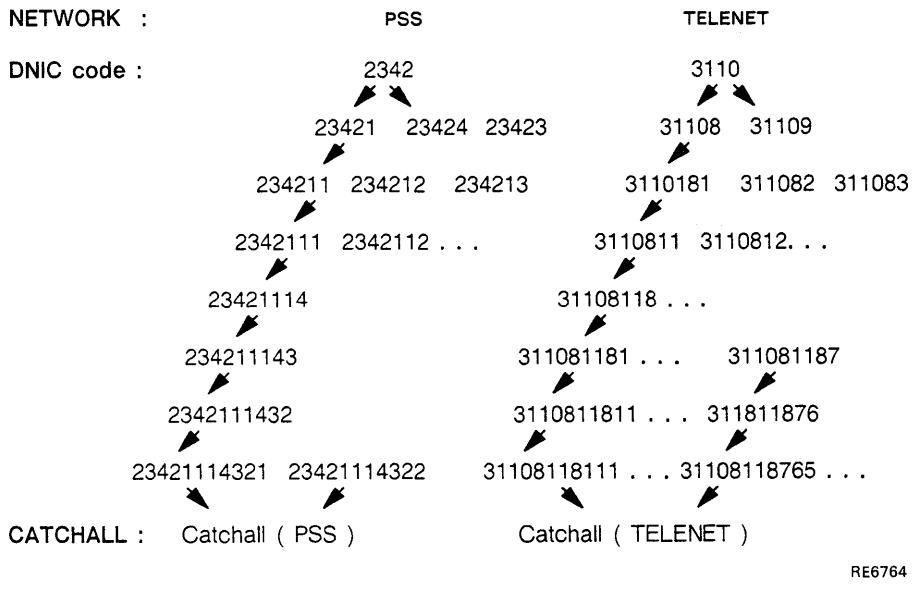


Remote DTEs with the rights identifier THEY_PAY can make incoming non-reverse charge calls only, while those with the rights identifier WE_PAY can make both non-reverse charge and reverse charge calls. Remote DTEs with rights identifiers other than THEY_PAY or WE_PAY cannot gain access to the local DTE because the last ACL entry is a match-all entry which matches all rights identifiers and specifies no access. See Section 5.3.3 for more information about this type of ACL entry.

5.4 Remote DTE Selection and the Remote DTE Tree

The Remote DTE Access Control Database groups remote DTEs according to the networks with which they are associated. The remote DTE entries for a network are organized as a tree structure, with entries that branch out from the top level entry that identifies the network. This top level entry is called the Data Network Identifier Code (DNIC). PSI Security searches the database for the remote DTE entry with an address that is the closest match to that specified. The organization of remote DTEs in the Remote DTE Access Control Database is shown in Figure 5–2.

Figure 5–2 The Remote DTE Tree



Each entry in the DTE tree is referred to as a node. Each node in the tree is either a part of a remote DTE address or a whole remote DTE address, and each node has an ACL.

When trying to locate a remote DTE address in the Remote DTE Access Control Database for an outgoing call, PSI Security uses a remote DTE matching algorithm to select an entry as follows:

1. The system selects the entry that matches the remote DTE address most closely. For example, if there are entries for 2342, 23421, 23422, 23423, 234211, 234212, 2342213, 2342111 and so on (as in Figure 5–2), the system will select the entry 234223 as the closest match to 23422364321.
2. If there is no match for the remote DTE, even for the first four digits, the system will select a CATCHALL entry (if there is one).
3. If the previous two procedures fail to select an entry, the outgoing call is cleared.

CATCHALL entries must be used with a network qualifier in the Remote DTE Access Control Database, to indicate the network to which the CATCHALL applies. If used without a network qualifier, the CATCHALL will apply to all remote DTEs associated with all networks connected to your system.

NOTE

If a remote DTE entry does not have an ACL associated with it, and there is no CATCHALL entry for that network, the outgoing call is **ALLOWED**.

See Section 5.3.2 for more information about CATCHALLs.

5.5 The Agent Rights Databases

This section describes the Agent Rights Databases and the contents of each database. Examples are included to show the use of rights identifiers in each database. See Section 5.7 for a description of how the security mechanism uses these databases.

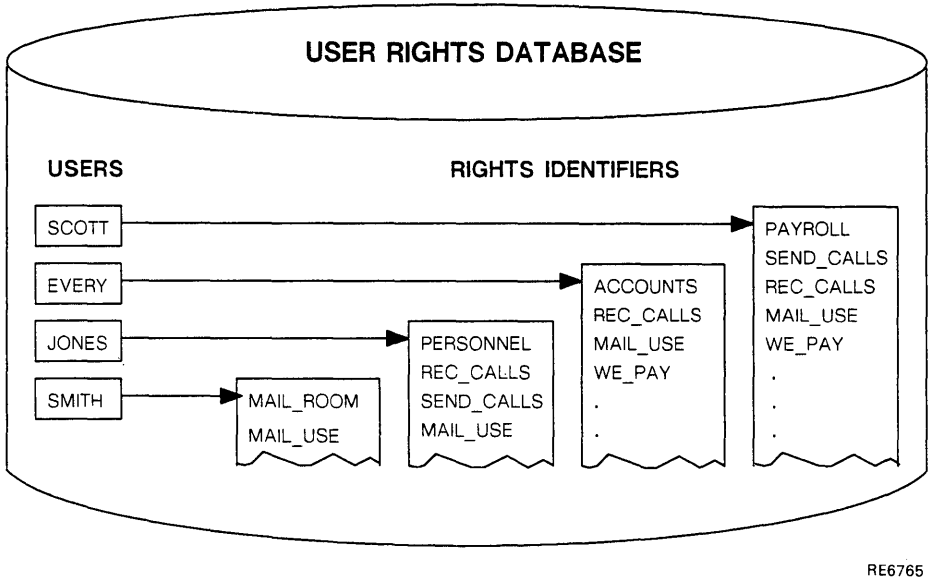
5.5.1 The User Rights Database

The User Rights Database contains the rights identifiers associated with the users on your system who want to make outgoing calls to remote DTEs. The User Rights Database consists of:

1. A list of users on your system. Users are specified by UIC identifiers defined in the System Rights Database, or by alphanumeric usernames that you have defined.
2. A list of the rights identifiers for each user who is allowed to make outgoing calls from your system.

Refer to Figure 5–3 for a diagram of the User Rights Database.

Figure 5–3 User Rights Database



In Figure 5–3, the users are SCOTT, EVERY, JONES and so on. Each user has a list of rights identifiers such as PERSONNEL, SEND_CALLS, MAIL_USE and so on. The user SCOTT has rights defined by the identifiers PAYROLL, SEND_CALLS, REC_CALLS, MAIL_USE and WE_PAY. The access actions associated with these rights identifiers are specified in the object ACLs contained in the Object Access Control Databases.

Note that you can use UIC identifiers that are defined in the System Rights Database, as well the rights identifiers you have defined for VAX PSI; for example, UIC identifiers such as [20,115] and rights identifiers [MAIL_ROOM,SMITH].

For information about the commands used to set up and manage the User Rights Database, see Section 6.4.1.

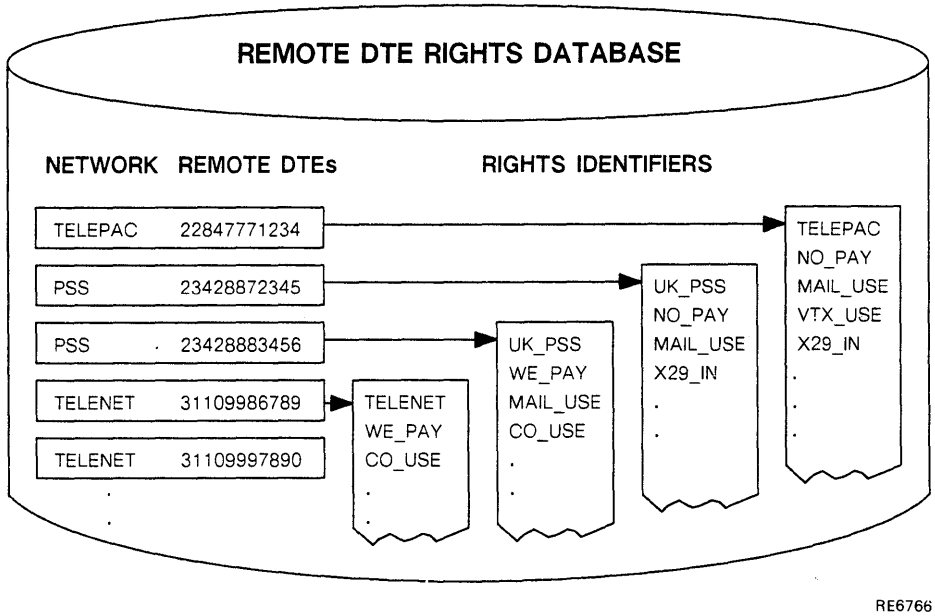
5.5.2 The Remote DTE Rights Database

The Remote DTE Rights Database contains the rights identifiers associated with remote DTEs that want to make incoming calls to your system. The Remote DTE Rights Database consists of:

1. A list of the remote DTEs that are allowed to make incoming calls to your system. These remote DTEs are specified by their addresses, and can include a character string specifying the PSDN, or other name.
2. A list of the rights identifiers for each remote DTE that is allowed to make incoming calls to your system.

Refer to Figure 5–4 for a diagram of the Remote DTE Rights Database.

Figure 5–4 Remote DTE Rights Database



In Figure 5–4, the remote DTEs are listed according to the PSDN with which they are associated: TELEPAC, the Swiss packet switching system; PSS, the United Kingdom packet switching system; TELENET, a US packet switching system; and so on.

Each remote DTE has a list of rights identifiers, TELEPAC, UK_PSS, MAIL_USE, WE_PAY and so on. Thus, remote DTE 31109986789 has access rights defined by the rights identifiers TELENET, WE_PAY and CO_USE. The access actions associated with these rights identifiers are specified in the ACLs contained in the Object Access Control Databases.

For information about the commands used to set up and manage the Remote DTE Rights Database, see Section 6.4.1.

5.5.3 The Access Node Rights Database

The Access Node Rights Database contains the rights identifiers associated with Access nodes that are allowed to make outgoing calls to remote DTEs via your Multi-host node. The Access Node Rights Database consists of:

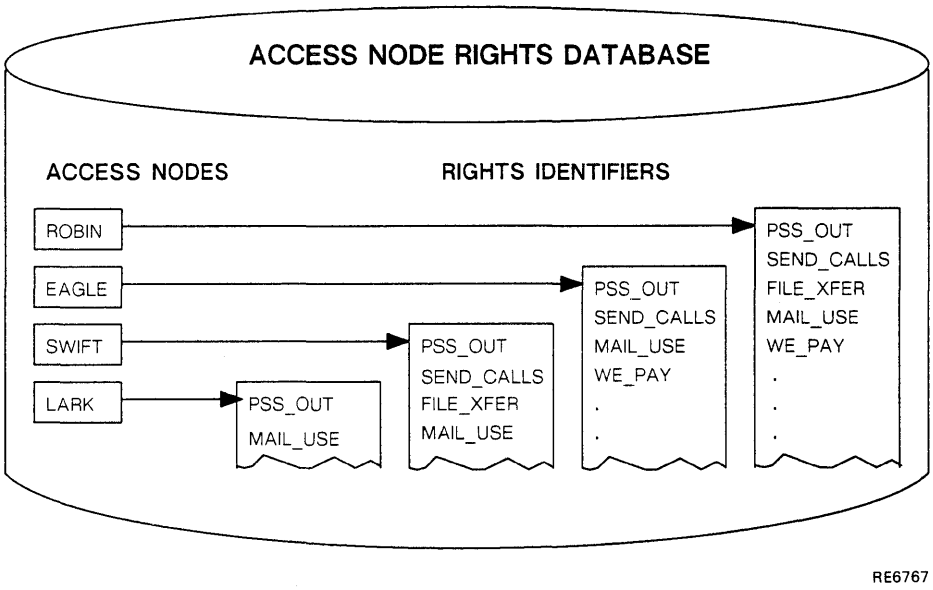
1. A list of the Access nodes that are allowed to make outgoing calls via your Multi-host node.

   The Access nodes are specified by their DECnet node names.

2. A list of rights identifiers for each Access node that is allowed to make outgoing calls via your Multi-host node.

Refer to Figure 5–5 for a diagram of the Access Node Rights Database.

Figure 5–5 Access Node Rights Database



In Figure 5–5, the Access nodes are ROBIN, EAGLE, LARK and so on. Each Access node has a list of rights identifiers such as PSS_OUT, SEND_CALLS, FILE_XFER and so on. The access actions associated with these rights identifiers are specified in the ACLs contained in the Object Access Control Databases.

For information about the commands used to set up and manage the Access Node Rights Database, see Section 6.4.1.

5.6 The Object Access Control Databases

This section describes the Object Access Control Databases and the contents of each database. Examples are included to show the use of rights identifiers and access actions in ACLs for object entries. See Section 5.7 for a description of how the security mechanism uses these databases.

5.6.1 The Local DTE Access Control Database

The Local DTE Access Control Database contains the ACLs that specify the rights identifiers and access actions associated with local DTEs for:

- Incoming calls from remote DTEs to Native, Multi-host or Access nodes
- Outgoing calls to remote DTEs from local users on a Native or Multi-host node

  Note that there is no local DTE check for outgoing calls to remote DTEs from users on an Access node.
- Outgoing calls to remote DTEs from Access nodes on a Multi-host node

For Native and Multi-host nodes, the Local DTE Access Control Database contains:

1. A list of the local DTEs connected to your system.

   Your system may be able to access one or more PSDNs, via one or more local DTEs. It is possible for your system to access a single PSDN via more than one local DTE.
2. An ACL for each local DTE entry, containing the rights identifiers and access actions associated with the DTE.

For Access nodes, the Local DTE Access Control Database contains:

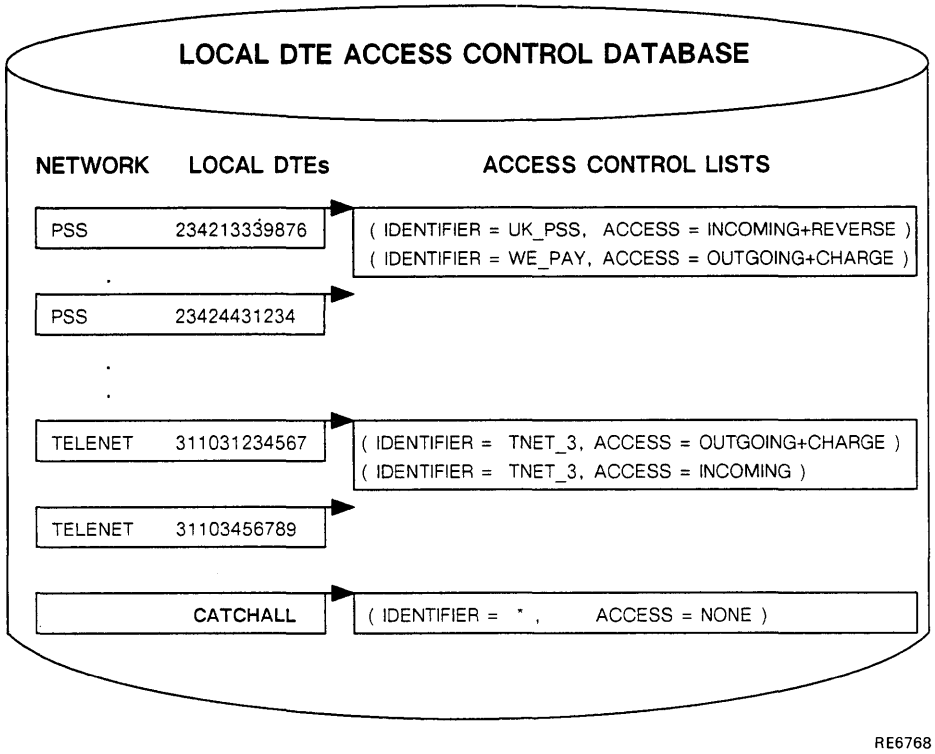
1. A list of the local DTEs which handle incoming calls to your node.

   Note that an Access node receives incoming calls via a Connector node. The Connector node may have access to one or more PSDNs via one or more local DTEs, and can be a VAX PSI Multi-host node or an X25router.
2. An ACL for each local DTE entry, containing the rights identifiers and access actions associated with the local DTE for incoming calls only.

   If you create an ACL to protect a local DTE for outgoing calls from your Access node, it has no effect. PSI Security on your Access node does not perform a local DTE check for outgoing calls.

Refer to Figure 5–6 for a diagram of the Local DTE Access Control Database on a Native or Multi-host node. Note that Access nodes have a similar database with ACLs specifying incoming access only.

Figure 5–6 Local DTE Access Control Database



In Figure 5–6, local DTEs with addresses starting with 2342 are connected to the United Kingdom PSS network. Local DTE 23423339876 allows incoming reverse charge calls from remote DTEs with the rights identifier UK_PSS.

Local DTE 311031234567 is connected to the US TELENET network. Users with the rights identifier TNET_3 can make outgoing calls, including reverse charge calls, using this local DTE because the ACL contains an entry that specifies the rights identifier TNET_3 with access actions OUTGOING+CHARGE. Also, remote DTEs with the rights identifier TNET_3 can make incoming non-reverse charge calls via this local DTE.

When specifying local DTEs, you must use a network qualifier to specify the network associated with each local DTE, such as PSS or TELENET. Although each local DTE connected to a PSDN has a unique address, you can the define different routes and access points as separate networks in the X25-PROTOCOL and X25-ACCESS databases. Where several local DTEs are connected to one or more PSDNs, the addition of a network qualifier allows you to set up PSI Security for each route and access point uniquely.

You can protect local DTEs individually by creating an ACL for each one, or you can protect them collectively by creating an ACL that applies to all of them. See Sections 6.4.2.1 and 6.4.2.4 for information about the commands to use.

For information about the commands used to set up and manage the Local DTE Access Control Database, see Section 6.4.2.1.

5.6.2 The Remote DTE Access Control Database

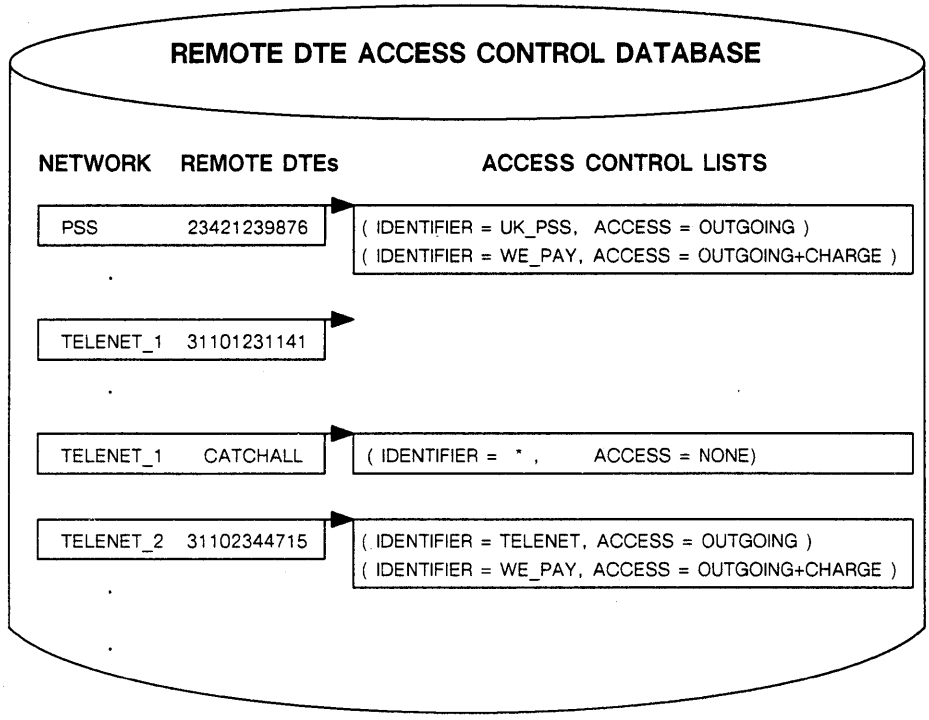
The Remote DTE Access Control Database is used in all configurations of VAX PSI. It contains the ACLs that specify the rights identifiers and access actions associated with remote DTEs for outgoing calls from your system. The Remote DTE Access Control Database consists of:

1. A list of remote DTEs that users on your system are allowed to access.
2. An ACL for each remote DTE entry, specifying the rights identifiers and outgoing access actions associated with the remote DTE.

When specifying remote DTEs, you must use a network qualifier to specify the network associated with the local DTE that handles the outgoing calls to the remote DTE. Although each local DTE connected to a PSDN has a unique address, you can the define different routes and access points as separate networks in the X25-PROTOCOL and X25-ACCESS databases. Where several local DTEs are connected to one or more PSDNs, the addition of a network qualifier allows you to set up PSI Security for each route and access point uniquely.

Refer to Figure 5–7 for a diagram of the Remote DTE Access Control Database.

Figure 5–7 Remote DTE Access Control Database



In Figure 5–7, the ACL for the remote DTE entry 23421239876 contains the rights identifier WE_PAY and the access action OUTGOING+CHARGE. A user on your node can make an outgoing call to this remote DTE if you have granted the rights identifier WE_PAY to the user in the User Rights Database.

Also, users to whom you have granted the rights identifier TELENET can make outgoing reverse charge calls to remote DTE 31102344715. Note that the outgoing call to this remote DTE is handled by the local DTE associated with the network specified by TELENET_2.

For information about the commands used to set up and manage the Remote DTE Access Control Database, see Section 6.4.2.2.

5.6.3 The Destination Access Control Database

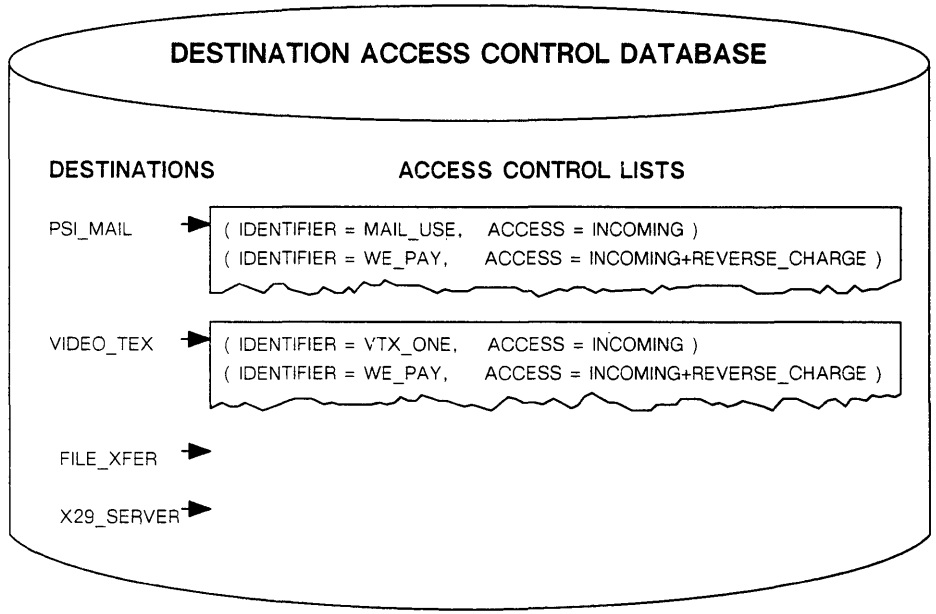
The Destination Access Control Database is used in all configurations of VAX PSI. It contains the ACLs that specify the rights identifiers and access actions associated with X.25 destinations for incoming calls from remote DTEs.

The Destination Access Control Database consists of:

1. A list of VAX PSI destinations on the host (such as electronic mail).
2. An ACL for each destination entry containing the rights identifiers and access actions associated with the destination.

Refer to Figure 5–8 for a diagram of the Destination Access Control Database.

Figure 5–8 Destination Access Control Database



In Figure 5–8, the destinations are PSI_MAIL (electronic mail), VIDEO_TEX and FILE_XFER (user written applications),and X29_SERVER (using the host from a remote terminal).

The ACL entry (IDENTIFIER=MAIL_USE, ACCESS=INCOMING) associated with the destination PSI_MAIL specifies the right to send incoming mail to your node. If you grant the rights identifier MAIL_USE to a remote DTE, it can send mail to your node.

The entry (IDENTIFIER=VTX_ONE, ACCESS=INCOMING +REVERSE_CHARGE) associated with the destination VIDEO_TEX specifies the right to the make incoming calls to use the videotext utility. If a remote DTE is granted the rights identifier VTX_ ONE, it can use the videotext utility.

Security checks on a destination are optional. In such cases, if incoming access is permitted via the local DTE (specified in the Local DTE Access Control Database), PSI Security allows an incoming call to proceed.

Destination names in the Destination Access Control Database must match the destinations defined in the X25-SERVER database. To place security restrictions on a destination, you associate an ACL with it.

NOTE

Only X.25 destinations are included in the X25-SERVER database. Although the X29-SERVER destination is included, individual X.29 destinations are not included.

You can set up PSI Security for declared destinations that are created when a process declares itself a network process via an IO$_ACPCONTROL QIO.

You can specify a declared destination and protect it with ACLs in the Destination Access Control Database before the destination is created by the declared network process. Alternatively, if a declared destination is defined already in the X25-SERVER database and has no security, you can create an ACL in the Destination Access Control Database to protect it. For more information about declared destinations, see Section 4.5.8.

On Multi-host nodes, Access nodes are defined as destinations in the X25-SERVER database. Therefore, you can set up Access nodes as destinations in the Destination Access Control Database on your Multi-host node, to control incoming calls to Access nodes.

5.7 The Security Checking Procedure

This section describes the security mechanism used to check that agents have the correct rights identifiers and access to objects when making incoming and outgoing calls.

Figures 5–9 and 5–10 show how PSI Security checks incoming and outgoing calls for the different configurations of VAX PSI.

Note that if the security databases are not set up when VAX PSI is first started, no security checks are made by the security software. See Chapters 6 and 7 for information about the commands used by PSI Security to set up the security databases.

5.7.1 Checking Incoming Calls

For incoming calls to your system, the agent is a remote DTE and the objects are a local DTE and, optionally, an X.25 destination. To check incoming calls to your system (refer to Figure 5–9), PSI Security:

1. Obtains the rights identifiers associated with the remote DTE from the Remote DTE Rights Database, using the remote DTE address supplied in the call packet.
2. Checks the protection you have placed on the local DTE associated with the incoming call.

   PSI Security:

   a. Searches for the local DTE entry in the Local DTE Access Control Database and obtains the associated ACL.

      Note that for incoming calls to Access nodes, this check is optional. If you have not set up protection for local DTEs for incoming calls to your Access node, PSI Security proceeds to step 3.

   b. Attempts to match the rights identifiers from the Remote DTE Rights Database with the rights identifiers in the ACL.

   c. Checks the access actions contained in the ACL entry when a match is found. If the access is INCOMING or INCOMING+REVERSE_CHARGE, the call is continued; otherwise the call is cleared.

      If no match is found between the rights identifiers from the Remote DTE Rights Database and the ACL from the Local DTE Access Control Database, the call is cleared.

3. Checks the protection you have placed on the destination, if a destination is associated with the incoming call. Note that the destination check is optional. If you have not set up security for the destination, the checking procedure ends at step 2. The call is allowed, provided that the local DTE check is verified.

   PSI Security:

   a. Searches for the destination entry in the Destination Access Control Database and obtains the associated ACL.

   b. Attempts to match the rights identifiers from the Remote DTE Rights Database with the rights identifiers in the ACL. If there is no match, the call is cleared.

   c. Checks the access actions contained in the ACL when a match is found. If the access is INCOMING or INCOMING+REVERSE_CHARGE, the call is continued; otherwise the call is cleared.

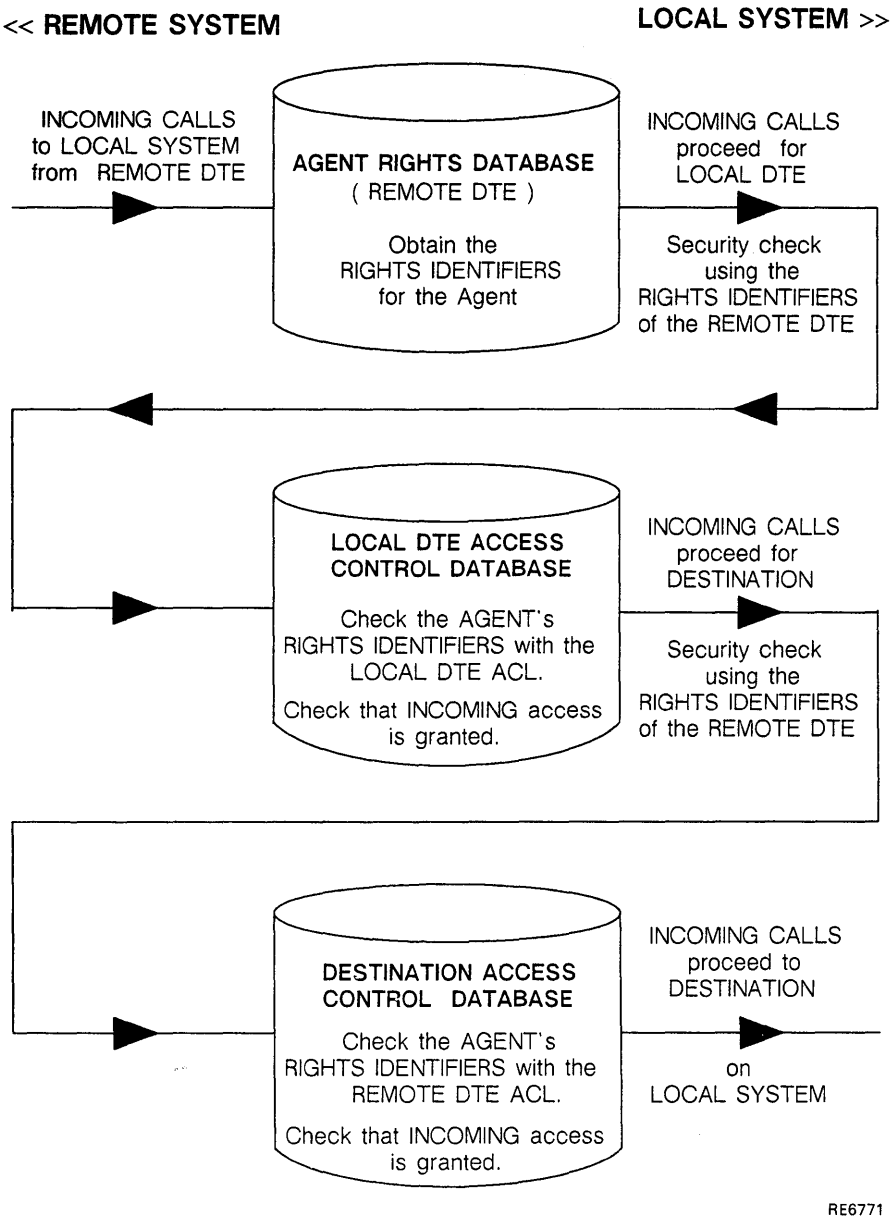
      If the call is to a destination that points to an Access node, the call is routed to the Access node.

If there is no ACL associated with an object, a CATCHALL entry is selected, if there is one. In this case, the incoming call is either allowed or cleared, depending on a rights identifier match and the access actions specified in the CATCHALL ACL.

NOTE

If there is no ACL associated with an object and no CATCHALL ACL, the incoming call is **ALLOWED**.

Figure 5–9 PSI Security Checks for Incoming Calls



5.7.2 Checking Outgoing Calls

For outgoing calls from VAX PSI, the agent is a user, process or Access node and the object is a remote DTE. To check outgoing calls from your system (refer to Figure 5–10), PSI Security:

1. Obtains the rights identifiers associated with:
   - The user or process for an Access or Native node.
   - The user, process or Access node for a Multi-host node.

   The rights identifiers for a user are obtained from the User Rights Database, while those for an Access node are obtained from the Access Node Rights Database.
2. Checks the protection you have placed on the local DTE associated with the outgoing call.

   For outgoing calls, PSI Security checks the local DTE protection on Native and Multi-host nodes only. PSI Security does not check the local DTE protection on an Access node for outgoing calls.

   For Native and Multi-host nodes, PSI Security:

   a. Searches for the local DTE entry in the the Local DTE Access Control Database, and obtains the associated ACL.

   b. Attempts to match the rights identifiers from the User or Access Node Rights Database with the rights identifiers specified in the ACL.

   c. When a match is found, PSI Security checks the access specified in the ACL. If the access is OUTGOING or OUTGOING+CHARGE, the call is continued; otherwise the call is cleared.
3. Checks the protection you have placed on the Remote DTE. PSI Security:

   a. Searches for the remote DTE entry in the Remote DTE Access Control Database, and obtains the associated ACL.

   b. Attempts to match the rights identifiers from the User or Access Node Rights Database with the rights identifiers specified in the ACL.

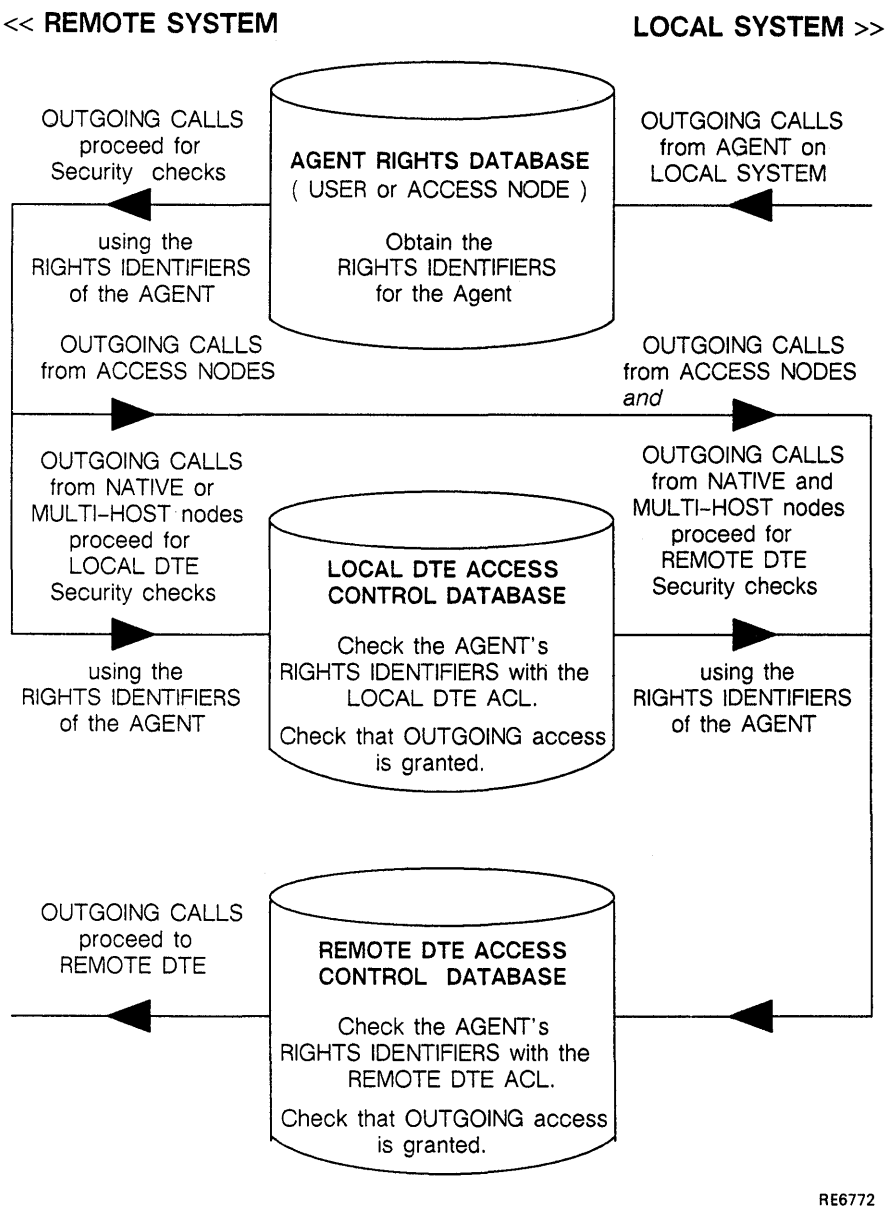
   c. When a match is found, PSI Security checks the access specified in the ACL. If the access is OUTGOING or OUTGOING+CHARGE, the call is continued; otherwise the call is cleared.

If there is no ACL associated with an object, a CATCHALL entry is selected, if there is one. In this case, the outgoing call is either allowed or cleared, depending on a rights identifier match and the access actions specified in the CATCHALL ACL.

NOTE

If there is no ACL associated with an object and no CATCHALL ACL, the outgoing call is **ALLOWED**.

Figure 5–10 PSI Security Checks for Outgoing Calls



6

How to Use PSI Security

This chapter describes how to set up PSI Security using the PSIAUTHORIZE Utility and provides a summary of the PSIAUTHORIZE commands. Examples that show how to use the these commands for different configurations of VAX PSI are included towards the end of the chapter.

For more information about a PSIAUTHORIZE command, see the appropriate section in Chapter 7, which provides a full description of each command.

6.1 The PSIAUTHORIZE Utility

The PSIAUTHORIZE utility is a system management tool used to set up PSI Security. You use PSIAUTHORIZE to control access to VAX PSI and to allocate resources to users.

NOTE

1. You must create entries in the PSI Security databases before you can use the wildcard character in PSIAUTHORIZE commands.
2. When entering PSIAUTHORIZE commands, do not mix wildcard and other characters in the same parameter. Use either a wildcard (*) or ASCII characters, but not both. Note that the wildcard character cannot be used to specify users.
3. When specifying local and remote DTEs, you must use the /NETWORK qualifier.

6.2 Starting PSI Security

To invoke PSI Security, enter this command at the DCL prompt:

```
$ RUN SYS$SYSTEM:PSIAUTHORIZE
```

The system responds with:

```
PSI-authorize>
```

You can then enter one of the commands described in the following sections. A full description of each command is given in Chapter 7.

To exit PSIAUTHORIZE, issue the `EXIT` command at the `PSI-authorize>` prompt, or press <CTRL/Z>.

6.2.1 Privileges/Restrictions

The use of PSIAUTHORIZE requires TMPMBX, NETMBX, OPER and SECURITY privilege.

The PSIAUTHORIZE utility requires write access to RIGHTSLIST.DAT in the directory SYS$SYSTEM. Write access to this file is normally restricted to users with the system UIC or SYSPRV privilege. PSIAUTHORIZE is installed with SYSLCK privilege.

Users having BYPASS privilege will bypass the PSI Security checks.

If PSI Security is not enabled on your system, users require NETMBX privilege only to use VAX PSI.

6.3 PSIAUTHORIZE Commands

6.3.1 PSIAUTHORIZE Command Format

The format of PSIAUTHORIZE commands is as follows:

```
PSI-authorize> COMMAND command-parameters command-qualifiers
```

6.3.2 PSIAUTHORIZE Command Summary

The summary of PSIAUTHORIZE commands is as follows:

```
ADD/IDENTIFIER identifier

DEFINE/KEY key-name equivalence-sting
           /[NO]ECHO
           /[NO]IF STATE={state-name,..}
           /[NO]LOCK_STATE
           /[NO]SET_STATE=state-name
           /[NO]TERMINATE

EXIT

GRANT/IDENTIFIER identifier agent
                            /NODE node-name
                            /REMOTE_DTE dte-address
                                       /NETWORK=network-name
                            /USER user-name

HELP command

REMOVE/IDENTIFIER identifier

REVOKE/IDENTIFIER identifier agent
                             /NODE node-name
                             /REMOTE_DTE dte-address
                                        /NETWORK=network-name
                             /USER user-name

SET DESTINATION destination-name
                /CATCHALL
                                  /ACL=acl-list
                                  /AFTER=acl-entry
                                  /DELETE
                                  /LIKE=destination-name
                                  /NEW

SET LOCAL_DTE dte-address/NETWORK=network-name
              CATCHALL
                          /ACL=acl-list
                          /AFTER=acl-entry
                          /DELETE
                          /LIKE=dte-address
                          /NEW

SET LOCAL_NODE
                /ACL=acl-list
                /AFTER=acl-entry
                /DELETE
                /NEW
```

```
SET REMOTE_DTE dte-address/NETWORK=network-name
               CATCHALL
                         /ACL=acl-list
                         /AFTER=acl-entry
                         /DELETE
                         /LIKE=dte-address
                         /NEW

SHOW DESTINATION destination-name
                 /CATCHALL

SHOW LOCAL_DTE dte-address/NETWORK=network-name
               CATCHALL

SHOW LOCAL_NODE

SHOW REMOTE_DTE dte-address/NETWORK=network-name
                CATCHALL

SHOW/IDENTIFIER identifier
                         /BRIEF
                         /FULL

SHOW/RIGHTS agent
           /NODE node-name
           /REMOTE_DTE dte-address/NETWORK=network-name
           /USER user-name
```

6.3.3 PSIAUTHORIZE Command Descriptions

Using PSIAUTHORIZE you control access to VAX PSI and its resources by creating and modifying the security databases.

The System Rights Database is a permanent database that contains the rights identifiers defined for agents associated with your system.

The commands that control data in the System Rights Database are:

ADD/IDENTIFIER	Creates a new rights identifier and adds it to the System Rights Database
REMOVE/IDENTIFIER	Removes an existing rights identifier from the System Rights Database

The Agent Rights Databases are volatile databases that contain a list of agents and their rights identifiers. There are three Agent Rights Databases:

- The User Rights Database
- The Access Node Rights Database
- The Remote DTE Rights Database

The commands that control and display data in the Agent Rights Databases are:

GRANT/IDENTIFIER	Grants a rights identifier to a specified agent
REVOKE/IDENTIFIER	Revokes a rights identifier from a specified agent
SHOW/IDENTIFIER	Displays the names and values of a specified rights identifier on the current output device
SHOW/RIGHTS	Displays the names of all rights identifiers held by a specified agent

The Object Access Control Databases are volatile databases that contain a list of objects and the ACLs that specify the rights identifiers and access actions associated with those objects. There are three Object Access Control Databases:

- The Local DTE Access Control Database
- The Remote DTE Access Control Database
- The Destination Access Control database

The commands that control and display data in the Object Access Control Databases are:

SET DESTINATION	Creates, modifies or deletes an ACL entry for a destination, according to the qualifiers specified
SET LOCAL_NODE	Creates, modifies or deletes an ACL entry for a local node on which the command is entered, according to the qualifiers specified
SET LOCAL_DTE	Creates, modifies or deletes an ACL entry for a local DTE, according to the qualifiers specified
SET REMOTE_DTE	Creates, modifies or deletes an ACL entry for a remote DTE, according to the qualifiers specified
SHOW DESTINATION	Displays the ACL, if there is one, for a destination, according to the qualifiers specified
SHOW LOCAL_NODE	Displays the ACL, if there is one, for the local node on which the command is entered, according to the qualifiers specified
SHOW LOCAL_DTE	Displays the ACL, if there is one, for a local DTE, according to the qualifiers specified
SHOW REMOTE_DTE	Displays the ACL, if there is one, for a remote DTE, according to the qualifiers specified

6.3.4 PSIAUTHORIZE Command Parameter Descriptions

acl-list

The format of PSIAUTHORIZE ACL lists follows the standard VMS format for access control lists:

```
ACL=((IDENTIFIER=identifier,ACCESS=access-action),...)
```

access-action

The type of access action associated with an object. The following types of access action can be specified:

NONE	No access at all
INCOMING	Incoming non-reverse charge calls only
INCOMING+REVERSE_CHARGE	All incoming calls
OUTGOING	Outgoing reverse charge calls only
OUTGOING+CHARGE	All outgoing calls

You can abbreviate access actions provided the abbreviation is not ambiguous. Note that:

REVERSE	Implies INCOMING+REVERSE_CHARGE
CHARGE	Implies OUTGOING+CHARGE

Wildcards are not permitted.

agent

The type of agent.

For incoming calls, the agent is a remote DTE, so the command parameter *agent* takes the form of a dte-address. See command parameter *dte-address* for more details.

For outgoing calls, the agent is either a local user, process or, for Multi-host nodes only, an Access node.

Local users are identified by their UICs or system rights identifiers, defined by the VMS Authorize utility. See command parameter *user-name* for more details.

Processes are defined by the UICs or rights identifiers of the user who initiated the process. See command parameter *user-name* for more details.

Access nodes are defined by their node-names, as defined in the Access Node Rights Database. See command parameter *node-name* for more details.

The following wildcard can be used with an Access node or remote DTE *agent* command parameter:

*	Specifies all agents of that type with rights identifiers defined by the VMS Authorize utility

Note that the wildcard character * cannot be used when specifying users. See command parameter *user-name* for more details.

destination-name

The destination name. There are two types of destination used by PSI Security:

1. Destinations that are created in the Destination Access Control Database from those defined in the X25-SERVER database using NCP commands.

   These destinations must have been defined in the X25-SERVER database before they can be protected by ACLs in the Destination Access Control Database. For information about defining destinations in the X25-SERVER database using NCP commands, see Section 2.3.4.

2. Declared destinations that are created in the Destination Access Control Database by declared network processes.

   A declared destinations is defined in the X25-SERVER database when a process declares itself to be a network processes by issuing an IO$_ACPCONTROL QIO.

   You can specify a declared destination and protected it with ACLs in the Destination Access Control Database before the destination is created in the X25-SERVER database by the declared network process. When the IO$_ACPCONTROL QIO is issued, you must specify the declared destination name in the PSI$C_NTD_NAME item field of the network process declaration block.

   If a declared destination is defined already in the X25-SERVER database and has no security, you can create an ACL in the Destination Access Control Database to protect it. You must specify the declared destination name as created from the PSI$C_NTD_NAME item field of the network process declaration block in the X25-SERVER database.

   For more information, see the *VAX P.S.I. X.25 Programmer's Guide.*

The following wildcard is allowed:

*	Implies all X25-SERVER destinations that have been defined in the X25-SERVER database using NCP commands

dte-address

The local or remote DTE address. The special meaning of incomplete remote DTE numbers is explained in Section 5.4. The command parameter *dte-address* is used with both agent and object commands.

With agent commands, it is used to specify remote DTEs only with the /REMOTE_DTE command qualifier. These commands grant, revoke and show rights identifiers for remote DTEs in the Remote DTE Rights Database.

With object commands, *dte-address* is used to specify local and remote DTEs, and is unqualified. Object commands create, modify, delete and show ACL entries for local and remote DTEs, and express explicitly whether the command applies to a local or remote DTE.

The following wildcards apply to both local and remote DTEs:

*
: For local DTEs, implies all local DTEs for all networks. If used with the /NETWORK qualifier, implies only the local DTEs associated with that network.

  For remote DTEs, implies all remote DTEs that are currently defined for the network specified by the /NETWORK qualifier. If used with /NETWORK=*, this implies all remote DTEs that are currently defined for all networks.

CATCHALL
: For local DTEs, this is an entry that applies if an ACL for a requested local DTE cannot be found.

  If your system has more than one local DTE associated with more than one network, you can define a local DTE CATCHALL for each network by including the /NETWORK qualifier. If you omit the /NETWORK qualifier, the local DTE CATCHALL applies to all local DTEs associated with all networks on your system. Note that if you specify both forms of local DTE CATCHALL, PSI Security will select a network specific local DTE CATCHALL entry in preference to the global local DTE CATCHALL entry.

  For remote DTEs, this is an entry that applies if an ACL for a requested remote DTE associated with a specified network cannot be found. Note that you can define a remote DTE CATCHALL entry for each network.

identifier

Rights identifier name. An arbitrary name used as a label to provide a matching mechanism for access rights.

Only user defined rights identifiers can be granted/revoked.

The wildcard character * is allowed with the SHOW/IDENTIFIER and SHOW/RIGHTS commands.

*
: When used with SHOW/IDENTIFIER command, implies all rights identifiers defined in the System Rights Database.

  When used with the SHOW/RIGHTS command, implies all rights identifiers held by the specified agent, and defined in the System Rights Database.

network-name

Unique network name. Each PSDN connected to a Native or Multi-host VAX PSI node, or to a Connector node, is identified by a unique *network-name* using the /NETWORK qualifier. This unique network identifier is used for two reasons:

1. Your Native or Multi-host node may be connected to more than one PSDN.
2. Your Access node needs to know which PSDN is connected to which Connector node.

Although network names are used also by Access nodes, they must be unique across ALL your VAX PSI systems. For more information about network names, see the *VAX P.S.I. Installation Procedures.*

The network associated with a local DTE is specified in the X25-PROTOCOL database on the Multi-host or Native node to which the local DTE is connected. This network is also specified in the X25-ACCESS database on any Access nodes in the system. Usually, the *network-name* used to specify the network is the profile name of the PSDN to which the local DTE is connected.

Note that the network associated with a remote DTE must be the same as the network associated with the local DTE which handles incoming or outgoing calls to the remote DTE. Therefore, the *network-name* associated with a remote DTE is the same as the *network-name* associated with the local DTE that handles incoming and outgoing calls to the remote DTE. Usually, this *network-name* is the profile name of the PSDN to which the local DTE is connected.

However, if your Multi-host or Native node has access via several local DTEs to the same PSDN, you can distinguish each route and access point to the PSDN uniquely. For example, you can distinguish between two local DTEs connected to PSS, by creating two *network-name* parameters, such as PSS_1 and PSS_2, in the X25-PROTOCOL database on your Multi-host or Native node. You can then set up PSI Security for each route and access point to PSS uniquely by using the /NETWORK qualifier.

The following wildcard is allowed for remote DTEs only:

*	Implies all networks for the remote DTEs that are already defined in the Remote DTE Access Control Database

node-name

The name of an Access node. Access nodes can be considered as agents for outgoing calls via a Connector node. The Connector node may have access to one or more PSDNs via one or more local DTEs, and can be a VAX PSI Multi-host node or an X25router. Access node names must be the same as those defined in the DECnet-VAX Configuration Database using NCP.

The command parameter *node-name* is used with the /NODE command qualifier.

The following wildcard is allowed:

*	Implies all Access nodes that are currently defined in the Access Node Rights Database

object

The type of object for incoming and outgoing calls.

For incoming calls, the objects are a local DTE and a destination.

For outgoing calls, the objects are a local DTE and a remote DTE.

Therefore, the command parameter *object* takes the form of a local DTE address, remote DTE address, or a destination name.

For more details, see the appropriate command parameter.

user-name

The name of a local user. A local user can act as an agent for outgoing calls. Local users are identified by their UICs or system rights identifiers, defined by the VMS Authorize utility.

The command parameter *user-name* is used with the command qualifier /USER.

Several forms of *user-name* are allowed:

[*abc,xyz*]	specifies a UIC for a user
[*MAIL_ROOM,SMITH*]	specifies a user by system rights identifier, defined by the VMS Authorize utility

6.4 Commands to Set Up and Modify the PSI Security Databases

This section describes the commands used to set up and modify the VAX PSI Security Databases. The main qualifiers used with the commands are also described. For full details of each command, see Chapter 7.

6.4.1 Commands to Set Up and Modify the Agent Rights Databases

You must have read and write access to SYS$SYSTEM:RIGHTSLIST.DAT to set up or modify an Agent Rights Database.

6.4.1.1 Commands to ADD, REMOVE and SHOW Rights Identifiers in the System Rights Database

To add, remove and show rights identifiers from the System Rights Database on your node, you use the following commands:

`ADD/IDENTIFIER` *identifier*

creates a new rights identifier and adds it to the System Rights Database.

`REMOVE/IDENTIFIER` *identifier*

removes a rights identifier from the System Rights Database.

`SHOW/IDENTIFIER` *identifier*

displays information from the System Rights Database about the specified rights identifier on the current output device.

Note that the System Rights Database is permanent. You can use the VMS AUTHORIZE utility to create and add new rights identifiers for PSI Security. However, you must use the PSIAUTHORIZE utility to invoke PSI Security and to grant and revoke rights identifiers to agents in the PSI Security Databases.

6.4.1.2 Commands to GRANT, REVOKE AND SHOW Rights Identifiers in the Agent Rights Databases

Note that while the System Rights Database is permanent, the Agents Rights Databases are volatile.

Information concerning users and their rights identifiers can be added to the System Rights Database with the VMS AUTHORIZE utility. However, to invoke PSI Security, you must use PSIAUTHORIZE to grant and revoke rights identifiers to users defined in the User Rights Database.

You use the following commands to grant, revoke and show the rights identifiers owned by the agents associated with your system:

`GRANT/IDENTIFIER` *identifier agent* `command-qualifiers`

grants a rights identifier to an agent defined in an Agent Rights Database.

The identifiers that have been granted to an agent constitute that agent's rights.

`REVOKE/IDENTIFIER` *identifier agent* `command-qualifiers`

revokes an identifier from an agent defined in an Agent Rights Database.

`SHOW/RIGHTS` *agent* `command-qualifier`

displays the rights identifiers of an agent defined in an Agent Rights Database.

The agent is either a user or Access node for outgoing calls, or a remote DTE for incoming calls.

Command Qualifiers

The main qualifiers that can be used with these commands are:

- /USER - specifies that the rights identifier is associated with a user in the User Rights Database. /USER is the default.
- /REMOTE_DTE - specifies that the rights identifier is associated with a remote DTE in the Remote DTE Rights Database.
- /NODE - specifies that the rights identifier is associated with an access node in the Access Node Rights Database.
- /NETWORK - used with the /REMOTE_DTE qualifier to specify the network name associated with the remote DTE.

Other qualifiers that can be used with these commands are described in Chapter 7.

6.4.2 Commands to Set Up and Modify the Object Access Control Databases

Note that you must have defined rights identifiers for the agents associated with your system before you create ACLs that include these rights identifiers to protect the objects associated with your system.

When specifying local and remote DTEs, you must use the /NETWORK command qualifier to specify the network with which the DTE is associated.

6.4.2.1 Commands to SET and SHOW ACL Entries in the Local DTE Access Control Database

The Local DTE Access Control Database is used to protect local DTEs from incoming and outgoing calls.

The following commands are used to SET and SHOW ACL entries for local DTEs in the Local DTE Access Control Database:

```
SET LOCAL_DTE dte-address command-qualifiers
```

creates, modifies or deletes an ACL entry for the specified local DTE, according to the command qualifiers specified.

If your system is an Access node, you should create a local DTE ACL for incoming calls only. Local DTE ACLs specifying outgoing access have no effect as PSI Security on your Access node does not check local DTEs for outgoing calls. Use the /NETWORK qualifier to specify the local DTE on the Multi-host node via which your Access node receives calls.

If your system is a Native or Multi-host system, use the /NETWORK qualifier to specify the local DTE via which your node sends and receives calls. If your system has more than one local DTE connected to more than one network, you must use the /NETWORK qualifier.

```
SET LOCAL_DTE CATCHALL command-qualifier
```

creates, modifies or deletes an ACL entry for the local DTE CATCHALL entry, according to the /ACL command qualifier specified. You can use the /NETWORK command qualifier to specify the network with which the local DTE CATCHALL is associated.

```
SHOW LOCAL_DTE dte-address command-qualifier
```

displays the ACL entries for the specified local DTE.

If your system is an Access node, use the /NETWORK qualifier to display the ACL entries for the local DTE on the Multi-host node via which your Access node receives calls. PSI Security on your Access node does not check local DTEs for outgoing calls.

If your system is a Native or Multi-host system, use the /NETWORK qualifier to display the ACL entries for the local DTE via which your node sends and receives calls. If your system has more than one local DTE connected to more than one network, you must use the /NETWORK qualifier.

```
SHOW LOCAL_DTE CATCHALL command-qualifiers
```

displays the ACL entries for the local DTE CATCHALL entry, according to the command qualifiers specified. If you have defined local DTE CATCHALL entries for each network associated with your system, you can use the /NETWORK command qualifier to display the local DTE CATCHALL associated with that network.

Command Qualifiers

The main qualifiers that are used with these commands are:

- /ACL=*acl-entry* - used with the SET command to specify the ACL entry for the local DTE
- /NETWORK=*network-name* - to specify the name of the network associated with the local DTE.

  This qualifier must be used if your system has more than one local DTE connected to a single network, or if your system has several local DTEs connected to more than one network.

Other qualifiers that can be used with these commands are described fully in Chapter 7.

6.4.2.2 Commands to SET and SHOW ACL Entries in the Remote DTE Access Control Database

The Remote DTE Access Control Database is used to protect remote DTEs from outgoing calls.

You can SET and SHOW ACL entries for remote DTEs in the Remote DTE Access Control Database by using the following commands:

```
SET REMOTE_DTE dte-address/NETWORK=network-name command-qualifiers
```

modifies or deletes an ACL entry for the specified remote DTE, according to the command qualifiers specified. You must use the /NETWORK command qualifier to specify the network with which the remote DTE is associated.

```
SET REMOTE_DTE CATCHALL/NETWORK=network-name command-qualifiers
```

creates, modifies or deletes an ACL entry for a remote DTE CATCHALL entry. You must use the /NETWORK command qualifier to specify the network with which the remote DTE is associated.

```
SHOW REMOTE_DTE dte-address/NETWORK=network-name
```

displays the ACL entries for the specified remote DTE. You must use the /NETWORK command qualifier to specify the network with which the remote DTE is associated.

```
SHOW REMOTE_DTE CATCHALL/NETWORK=network-name
```

displays the ACL entries for the remote DTE CATCHALL entry. You must use the /NETWORK command qualifier to specify the network with which the remote DTE CATCHALL entry is associated.

Command Qualifiers

The main qualifiers that are used with these commands are:

- /ACL=*acl-entry* - used with the SET command to specify the ACL entry for the remote DTE
- /NETWORK=*network-name* - you use the /NETWORK command qualifier to specify the network with which the remote DTE is associated.

  This qualifier must be used if your system is connected to more than one network.

Other qualifiers that can be used with these commands are described fully in Chapter 7.

6.4.2.3 Commands to SET and SHOW ACL Entries in the Destination Access Control Database

The Destination Access Control Database is used to protect destinations from incoming calls.

Local destinations other than declared destinations must have been defined in the X25-Server Database on each node using NCP. For more information, see Section 2.3.4.1.

Access nodes that are destinations on a Connector node must have been defined as such in the X25-Server Database using NCP. For more information, see Section 2.3.4.2.

You can SET and SHOW ACL entries for destinations in the Network Access Control Database by using the following commands:

`SET DESTINATION` *`destination-name`* `command-qualifiers`

creates, modifies or deletes an ACL entry for the specified destination entry, according to the command qualifiers specified.

`SET DESTINATION/CATCHALL command-qualifiers`

creates, modifies or deletes an ACL entry for the destination CATCHALL entry, according to the command qualifiers specified.

`SHOW DESTINATION` *`destination-name`* `command-qualifier`

displays the ACL entries for the specified destination.

`SHOW DESTINATION/CATCHALL`

displays the ACL entries for the destination CATCHALL entry.

Command Qualifiers

The main qualifiers that are used with these commands are:

- /ACL=*acl-entry* - used with the SET command to specify the ACL entry for the destination.
- /CATCHALL - to specify the destination CATCHALL entry.

Other qualifiers that can be used with these commands are described fully in Chapter 7.

6.4.2.4 Commands to SET and SHOW Local Node ACL Entries

The commands to SET and SHOW local node ACL entries entered on a local node apply to the node itself. Therefore, these commands do not require any command parameters.

The command SET LOCAL_NODE is equivalent to the SET LOCAL_DTE CATCHALL command, while the command SHOW LOCAL_NODE is equivalent to the SHOW LOCAL_DTE CATCHALL command.

The SET LOCAL_NODE command can be used when there is only one local DTE in a configuration, or if there are no special requirements for each local DTE/node in a system.

To SET and SHOW ACL entries for your local node, you can use the following commands:

```
SET LOCAL_NODE command-qualifiers
```

creates, modifies or deletes an ACL entry for the node itself, according to the command qualifiers specified.

```
SHOW LOCAL_NODE
```

displays the ACL entries for the node itself. This command does not take any command qualifiers.

Command Qualifiers

The main qualifier that is used with the SET command is:

- /ACL=*acl-entry* - used with the SET command to specify the ACL entry for the local node. Note that for Access nodes, you can only create ACLs specifying incoming access; PSI Security on your Access node does not check local DTEs for outgoing calls.

Other qualifiers that can be used with the SET command are described fully in Chapter 7.

6.4.3 The DEFINE/KEY Command

```
DEFINE/KEY
```

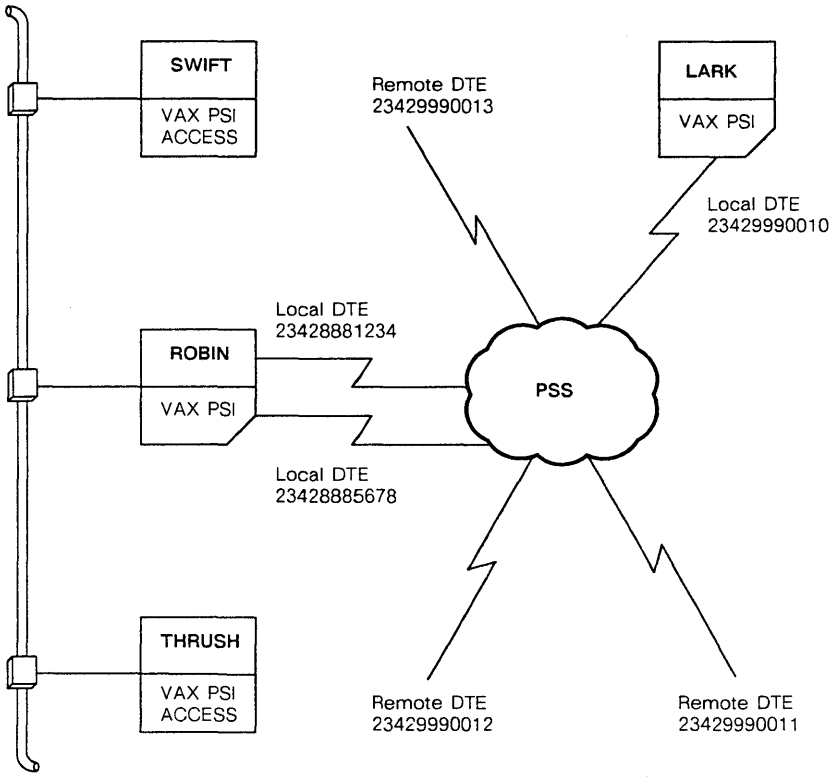
defines text or commands produced when a certain key on a terminal keyboard is typed. DEFINE/KEY is similar to the VMS DCL DEFINE/KEY command.

The default key definitions are loaded from the file that has the logical name PSIAUTHORIZE$INIT. The default key definitions can be seen by typing or printing this file.

6.5 Examples of How to Set Up PSI Security

The examples in this section show how to set up PSI Security for different configurations of VAX PSI. Read through the examples for details of the commands applicable to your system. Figure 6–1 shows the VAX PSI installation used in the examples.

Figure 6–1 An Example of a Typical PSI Installation



In Figure 6–1, node ROBIN is a Multi-host node and node LARK is a Native node. Both have VAX PSI installed. Nodes THRUSH and SWIFT are Access nodes with VAX PSI Access installed. ROBIN is connected to the PSS network via two local DTEs, 23428881234 and 23428885678. LARK is connected to the PSS network via local DTE 23429990010.

The examples show you how to set up PSI Security on Multi-host node ROBIN to:

- Allow incoming calls from known remote DTEs (see Section 6.5.1).
- Protect destinations (see Section 6.5.2).
- Allow users to make outgoing calls (see Section 6.5.3).

- Protect against unauthorized incoming and outgoing calls using CATCHALL commands (see Section 6.5.4).
- Set up PSI Security for incoming and outgoing calls to and from Access nodes THRUSH and SWIFT (see Section 6.5.5).
- Set up Access node THRUSH for incoming and outgoing calls via Multi-host node ROBIN (see Section 6.5.6).

In addition, Section 6.5.7 describes how you can set up PSI Security for a Combination node.

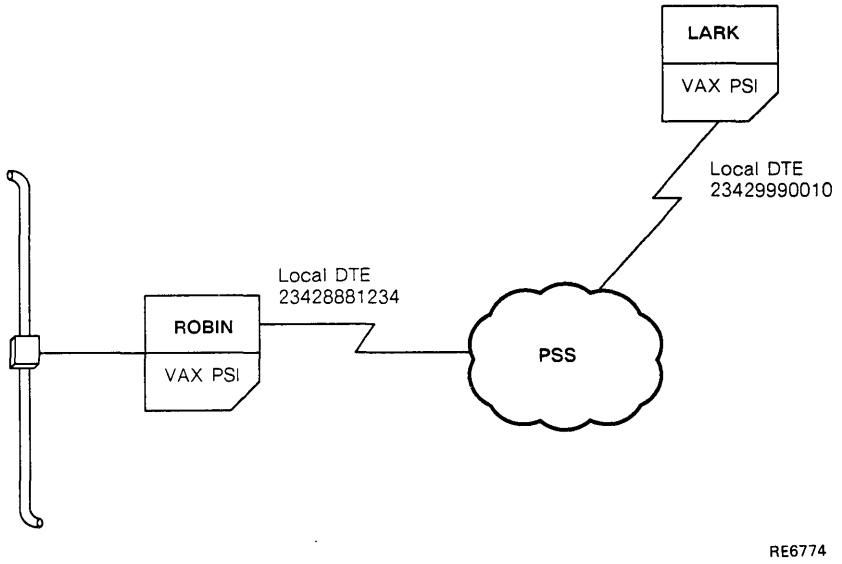
Note that the examples assume that the security systems on each node have not been used before; that is, the security databases are empty. In addition, valid abbreviated forms of the PSIAUTHORIZE commands are shown.

6.5.1 Example 1: Allowing Incoming Calls from Known Remote DTEs

This example shows you how to set up PSI Security to allow incoming calls to your node from known remote DTEs.

Multi-host node ROBIN is used in this example (see Figure 6–2).

Figure 6–2 Example 1: Allowing Incoming Calls



The following commands are entered on node ROBIN:

To protect a local DTE against incoming calls

```
PSI-authorize> ADD/ID PSI_INCOMING
PSI-authorize> SET LOCAL_DTE 23428881234 -
               _ /NET=PSS/ACL=(ID=PSI_INCOMING,ACC=INC)
```

These commands create the new rights identifier PSI_INCOMING and define the protection against incoming calls to node ROBIN via local DTE 23428881234. Only remote DTEs with the rights identifier PSI_INCOMING can make incoming calls via this local DTE. The access specified allows incoming non-reverse charge calls only.

To allow incoming calls from node LARK

```
PSI-authorize> GRANT/ID PSI_INCOMING /REMOTE_DTE/NET=PSS 23429990010
```

This command grants the rights identifier PSI_INCOMING to remote DTE 23429990010; that is, the remote DTE connected to node LARK.

You can allow other known remote DTEs to make incoming calls to node ROBIN by granting the rights identifier PSI_INCOMING selectively. For example,

```
PSI-authorize> GRANT/ID PSI_INCOMING /REMOTE_DTE/NET=PSS 23429990011
PSI-authorize> GRANT/ID PSI_INCOMING /REMOTE_DTE/NET=PSS 23429990012
PSI-authorize> GRANT/ID PSI_INCOMING . . .
PSI-authorize> . . .
```

In this way, you can grant incoming access to known remote DTEs only.

To allow incoming reverse charge calls from node LARK

You can set up PSI Security to allow incoming reverse charge calls to your node. You can allow such calls by creating and granting rights identifiers to the appropriate agents, and by specifying the appropriate access actions in object ACLs. For example:

```
PSI-authorize> ADD/ID PSI_WEPAY
PSI-authorize> SET LOCAL_DTE 23428881234/NET=PSS -
               _ /ACL=(ID=PSI_WEPAY,ACC=INC+REVERSE)
PSI-authorize> GRANT/ID PSI_WEPAY/REMOTE_DTE 23429990010
```

These commands create the new rights identifier PSI_WEPAY to allow incoming reverse charge calls to node ROBIN via local DTE 23428881234 for remote DTEs with that rights identifier. In this case, remote DTE 23429991234 (that is, node LARK) is allowed to make incoming reverse charge calls.

Note that if your system has several local DTEs, you can set up your local DTEs to accept incoming and incoming reverse charge calls as you wish. For example, you may choose to restrict incoming reverse charge calls to one local DTE.

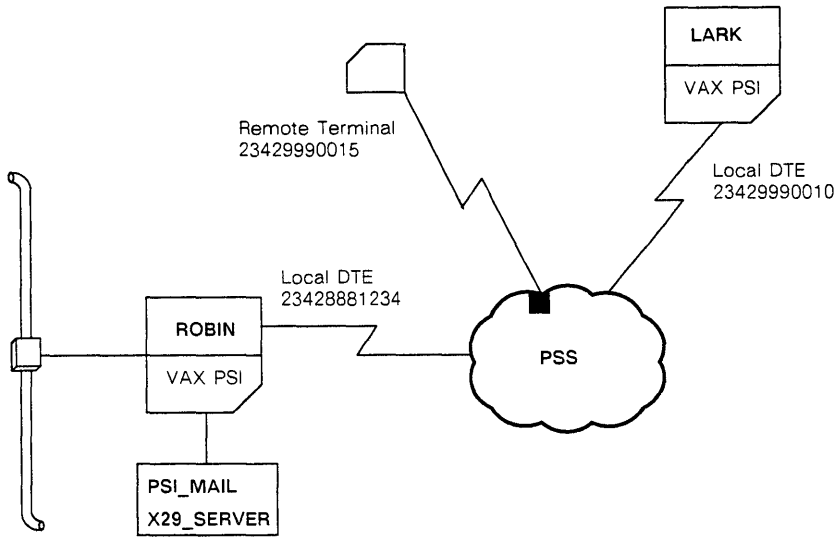
6.5.2 Example 2: Allowing Incoming Calls to a Particular Destination

This example shows you how to set up PSI Security to protect destinations on your node.

Protecting destinations on your node is an optional PSI Security measure. However, you should set up PSI Security for destinations if you want to provide a higher level of protection for your system.

Multi-host node ROBIN is used in this example (see Figure 6–3).

Figure 6–3 Example 2: Allowing Incoming Calls to a Particular Destination



The following commands are entered on node LARK:

To Protect the PSI-MAIL Destination on node ROBIN

```
PSI-authorize> ADD/ID PSI_MAIL
PSI-authorize> SET DEST PSI-MAIL /ACL=(ID=PSI_MAIL,ACC=INC)
PSI-authorize> GRANT/ID PSI_MAIL /REMOTE_DTE/NET=PSS 23429990010
```

These additional commands protect the PSI-MAIL destination on node ROBIN against incoming calls from unknown remote DTEs. In Example 1, node LARK (connected to remote DTE 23429990010) is granted incoming access to node ROBIN via local DTE 23428881234. In this example, node LARK is granted incoming access specifically to the PSI_MAIL destination.

Other known remote DTEs can be allowed to make incoming PSI-MAIL calls to node ROBIN also. For example,

```
PSI-authorize> GRANT/ID PSI_MAIL /REMOTE_DTE/NET=PSS 23429990011
PSI-authorize> GRANT/ID PSI_MAIL /REMOTE_DTE/NET=PSS 23429990012
PSI-authorize> GRANT/ID PSI_MAIL . . .
PSI-authorize> . . .
```

Note that you can create separate rights identifiers and ACLs to protect different destinations. Using this method, you can grant several levels of incoming access from known remote DTEs to different destinations on your system.

To Protect the X29-SERVER Destination on node ROBIN

Note that you should only grant incoming X.29 access to known remote terminals and PADs. If you allow incoming access to a public PAD, all users who have access to that public PAD may have the ability to make incoming X.29 calls to destinations on your system (see Section 4.7 for more information about X.29 security).

For example, to allow a remote terminal, connected to known remote DTE 23429990015, to make incoming X.29 calls, you can enter the following commands on node ROBIN:

```
PSI-authorize> ADD/ID PSI_X29
PSI-authorize> SET DEST X29-SERVER -
                _ /ACL=(ID=PSI_X29,ACC=INC)
PSI-authorize> GRANT/ID PSI_X29 /REMOTE_DTE/NET=PSS ,23429990015
```

These commands allow the remote terminal, connected to remote DTE 23429990015, to make incoming calls to the X29-SERVER destination on node ROBIN. Note that in Example 1, remote DTE 23429990015 is allowed incoming access to node ROBIN via local DTE 23428881234.

By granting the rights identifier PSI_X29 selectively, you can allow other known remote terminals or PADs to make incoming X.29 calls to node ROBIN. In this way, you can grant incoming X.29 access to known remote DTEs only.

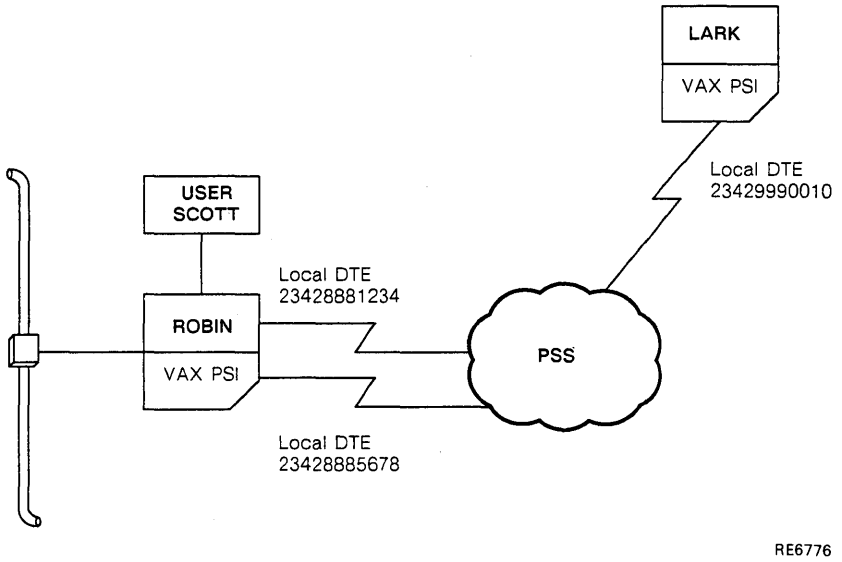
6.5.3 Example 3: Allowing Outgoing Calls

This example shows you how to allow users on your node to make outgoing calls.

Note that for Access nodes, you cannot set up local DTE protection for outgoing calls. PSI Security on your Access node does not support outgoing protection for local DTEs on the Connector or Multi-host node via which you have access to a PSDN. You can only protect access to remote DTEs for users on your Access node (see Section 6.5.7).

Multi-host node ROBIN is used in this example (see Figure 6–4).

Figure 6–4 Example 3: Allowing Outgoing Calls



The following commands are entered on node ROBIN:

To protect the local DTEs

```
PSI-authorize> ADD/ID PSI_OUTGOING
PSI-authorize> SET LOCAL_DTE 23428881234 -
_ /NET=PSS/ACL=(ID=PSI_OUTGOING,ACC=OUT+CHARGE)
PSI-authorize> SET LOCAL_DTE 23428885678 -
_ /NET=PSS/ACL=(ID=PSI_OUTGOING,ACC=OUT)
```

These commands create the rights identifier PSI_OUTGOING and define the protection for local DTEs 23428881234 and 23428885678 with the specified ACLs.

To allow user SCOTT outgoing access

```
PSI-authorize> GRANT/ID PSI_OUTGOING /USER SCOTT
```

This command grants the rights identifier PSI_OUTGOING to user SCOTT. Note that user SCOTT can now make outgoing calls and outgoing reverse charged calls via local DTEs 23428881234 and 23428885678 to any remote DTE.

Access to remote DTEs can be restricted by setting up ACLs to allow access to known remote DTEs only.

To allow user SCOTT to making outgoing calls to node LARK (known remote DTE 23429990010)

```
PSI-authorize> SET REMOTE_DTE 23429990010 -
               _ /NET=PSS/ACL=(ID=PSI_OUTGOING,ACC=OUT+CHARGE)
```

This additional command allows user SCOTT to making outgoing calls to node LARK; that is, to remote DTE 23429990010. Similar commands can be entered to allow user SCOTT, and other users, to make outgoing calls to other known remote DTEs. For example:

```
PSI-authorize> GRANT/ID PSI_OUTGOING /USER EVERY
PSI-authorize> GRANT/ID . . .
  .
  .
  .
PSI-authorize> SET REMOTE_DTE 23429990011 -
               _ /NET=PSS/ACL=(ID=PSI_OUTGOING,ACC=OUT+CHARGE)
PSI-authorize> SET REMOTE_DTE 23429990012 -
               _ /NET=PSS/ACL=(ID=PSI_OUTGOING,ACC=OUT+CHARGE)
PSI-authorize> SET REMOTE_DTE . . .
PSI-authorize> . . .
```

6.5.4 Example 4: Using Wildcard, Match-all and CATCHALL Commands

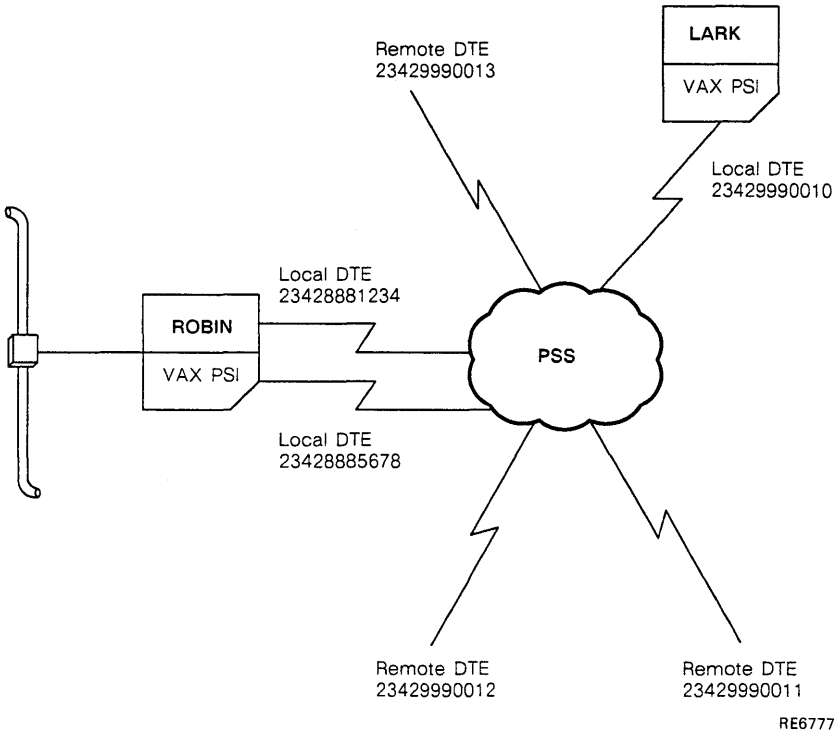
This example shows you how to use wildcard, match-all and CATCHALL commands for incoming and outgoing calls to and from your system:

- You can use an object wildcard to create an ACL entry for each object entry in a database, including the CATCHALL entry if there is one. Using a wildcard, you can allow unrestricted access to objects for known and trusted agents. For example, you may wish to allow trusted users to make unrestricted outgoing calls to any remote DTE within a PSDN. See Sections 6.1 and 6.3.4 for more information about wildcards.
- You can use database entries of the form (IDENTIFIER=*, ACCESS= . . .) to create match-all ACL entries to protect objects on your system against known agents. For example, you can ensure that a specific local DTE is not used for incoming calls, and is used only for outgoing calls. You can ensure that agents are granted the appropriate access by granting rights identifiers selectively. See Section 5.3.3 for more information about match-alls.
- You can use CATCHALL entries to create match-all ACL entries of the form (IDENTIFIER=*, ACCESS=NONE) to protect objects on your system against unknown agents. For example, you can protect your local DTEs from incoming calls from unknown remote DTEs. See Section 5.3.2 for more information about CATCHALL entries.

Note that for Access nodes, you cannot set up CATCHALL local DTE protection for outgoing calls. PSI Security on your Access node does not support outgoing protection for local DTEs on the Connector or Multi-host node via which you have access to a PSDN. You can only protect access to remote DTEs for users on your Access node.

Multi-host node ROBIN is used in this example (see Figure 6–5).

Figure 6–5 Example 4: Using Wildcard, Match-all and CATCHALL Commands



For example, the following commands are entered on node ROBIN:

To allow unrestricted outgoing access via local DTEs collectively

```
PSI-authorize> ADD/ID PSI_ALL
PSI-authorize> SET LOCAL_DTE * /ACL=(ID=PSI_ALL, -
               _ ACC=OUT+CHARGE)
```

This command allows any user with the rights identifier PSI_ALL to make outgoing calls via any of the local DTEs defined in the Local DTE Access Control Database on node ROBIN. Note that the command applies to all local DTE database entries, including any CATCHALL entry.

The following commands allow users SCOTT and EVERY on node ROBIN to make unrestricted outgoing calls to any remote DTE in the PSS network (that is, to all remote DTEs with addresses starting with the DNIC for PSS):

```
PSI-authorize> GRANT/ID PSI_ALL /USER SCOTT
PSI-authorize> GRANT/ID PSI_ALL /USER EVERY
PSI-authorize> SET REMOTE_DTE 2342 /NEW -
               _ /NET=PSS/ACL=(ID=PSI_ALL,ACC=OUT+CHARGE)
```

Note the use of the /NEW qualifier to delete any existing ACL for remote DTE entry 2342 and replace it with the new ACL.

Similarly, the following commands allow known and trusted remote DTEs to make incoming calls via local DTE 23428881234 to all destinations defined in the Destination Access Control Database on node ROBIN:

```
PSI-authorize> GRANT/ID PSI_ALL /REMOTE_DTE 23429990010
PSI-authorize> GRANT/ID PSI_ALL /REMOTE_DTE 23429990011
PSI-authorize> GRANT/ID PSI_ALL /REMOTE_DTE 23429990012
PSI-authorize> SET LOCAL_DTE 23428881234 /NET=PSS -
                 _ /ACL=(ID=PSI_ALL,ACC=INC)
PSI-authorize> SET DEST * /ACL=(ID=PSI_ALL,ACC=INC)
```

Note that these commands also allow the specified remote DTEs to make incoming calls to any Access nodes already defined in the Destination Access Control Database on node ROBIN. PSI Security on the Access nodes can be set up to restrict such incoming calls.

To create a local DTE match-all entry

```
PSI-authorize> SET LOCAL_DTE 23428885678 /NET=PSS -
               _ /ACL=(ID=*,ACC=NONE)
```

This command, when placed at the end of the ACL for local DTE 23428885678, denies access for agents whose rights identifiers do not match with those in entries higher up the ACL.

In these examples, incoming calls to local DTE 23429995678 are cleared. Outgoing calls via local DTE 23429995678 from agents with either PSI_OUTGOING or PSI_ALL are allowed.

To create a local DTE CATCHALL entry

```
PSI-authorize> SET LOCAL_DTE CATCHALL /ACL=(ID=*,ACC=NONE)
```

This command ensures that you have set up PSI Security correctly so that known but restricted agents do not gain access to objects on your system inadvertently. In the example above, the CATCHALL ACL specifies ACCESS=NONE.

The protection works because an object CATCHALL ACL entry is always selected if there is no ACL associated with an object, or if a rights identifier match for a known but restricted agent does not occur for entries higher up the ACL.

Note that unknown agents, with no rights identifiers, are automatically denied access to your system by PSI Security.

6.5.5 Example 5: Setting Up PSI Security On a Multi-host Node

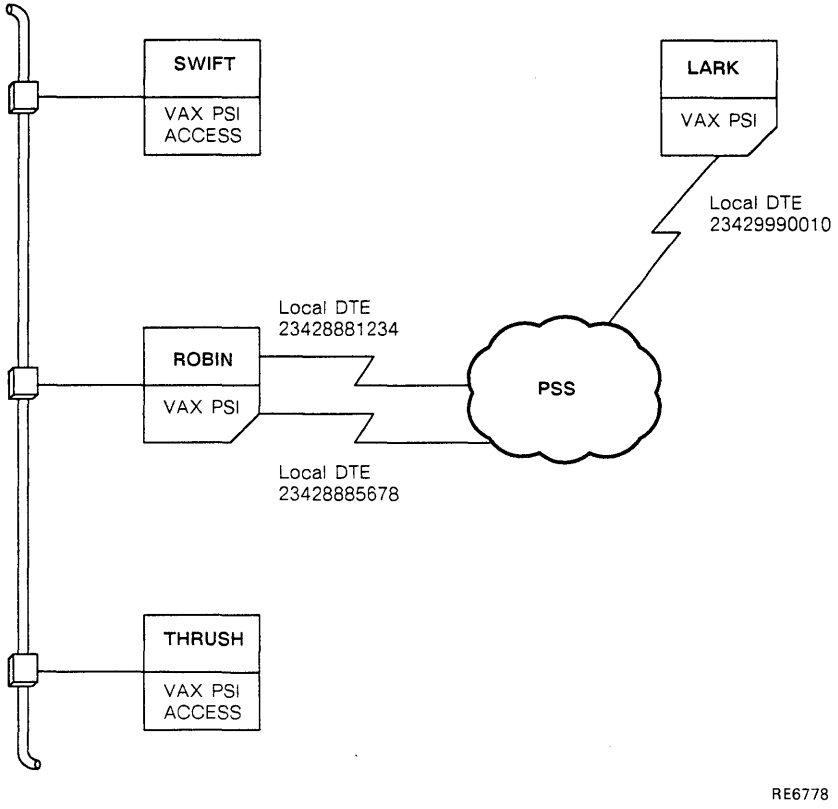
This example shows how to set up PSI Security on a Multi-host node for calls to and from Access nodes.

PSI Security destination checks are optional for incoming calls. Incoming calls to Access nodes that pass the local DTE check on your Multi-host node are allowed to proceed unless you set up PSI Security to restrict such calls.

Also, outgoing calls from users on an Access node that pass the local and remote DTE checks on your Multi-host node are allowed to proceed unless you set up PSI Security to prevent such calls. If you do not set up PSI Security for Access nodes on your Multi-host, you could incur large operating costs if you allow Access node users to make outgoing calls at will.

Multi-host node ROBIN is used in this example (see Figure 6–6).

Figure 6–6 Example 5: Setting Up PSI Security On a Multi-host Node



For example, the following commands are entered on node ROBIN:

To allow incoming calls to nodes THRUSH and SWIFT from node LARK only

```
PSI-authorize> ADD/ID PSI_ACCESS
PSI-authorize> SET LOCAL_DTE 23428881234 /NET=PSS -
               _ /ACL=(ID=PSI_ACCESS,ACC=INC)
PSI-authorize> SET DEST THRUSH /ACL=(ID=PSI_ACCESS,ACC=INC)
PSI-authorize> SET DEST SWIFT  /ACL=(ID=PSI_ACCESS,ACC=INC)
PSI-authorize> GRANT/ID PSI_ACCESS /REMOTE_DTE 23429990010
```

These commands create a new rights identifier on node ROBIN, to allow remote DTE 23429990010 (that is, node LARK) to make incoming calls to Access nodes THRUSH and SWIFT (specified as destinations on Multi-host node ROBIN). The commands specify that local DTE 23428881234 (connected to node ROBIN) can handle incoming calls from node LARK to nodes THRUSH and SWIFT.

You can grant the rights identifier PSI_ACCESS selectively to allow other known remote DTEs to make incoming calls via Multi-host node ROBIN to Access nodes THRUSH and SWIFT.

To allow outgoing calls from users on nodes THRUSH and SWIFT to node LARK

```
PSI-authorize> GRANT/ID PSI_ACCESS /NODE THRUSH
PSI-authorize> GRANT/ID PSI_ACCESS /NODE SWIFT
PSI-authorize> SET LOCAL_DTE 23428881234 /NET=PSS -
               _ /ACL=(ID=PSI_ACCESS,ACC=OUT+CHARGE)
PSI-authorize> SET REMOTE_DTE 23429990010 /NET=PSS -
               _ /ACL=(ID=PSI_ACCESS,ACC=OUT+CHARGE)
```

These additional commands allow all users on nodes THRUSH and SWIFT to make outgoing calls to node LARK (that is, to remote DTE 23429990010). The commands specify that local DTE 23428881234 connected to node ROBIN can handle outgoing calls to node LARK from nodes THRUSH and SWIFT.

Note that PSI Security must be set up on nodes THRUSH and SWIFT to control which users can make outgoing calls.

The two local DTE commands used in these examples specify the same rights identifier, PSI_ACCESS. These two commands can be combined to form a single ACL entry for the local DTE as follows:

```
PSI-authorize> SET LOCAL_DTE 23428881234/NET=PSS -
               _ /ACL=(ID=PSI_ACCESS,ACC=INC+OUT+CHARGE)
```

This single ACL entry controls both incoming and outgoing calls to Access node THRUSH using the single rights identifier PSI_ACCESS.

6.5.6 Example 6: Setting Up PSI Security On an Access Node

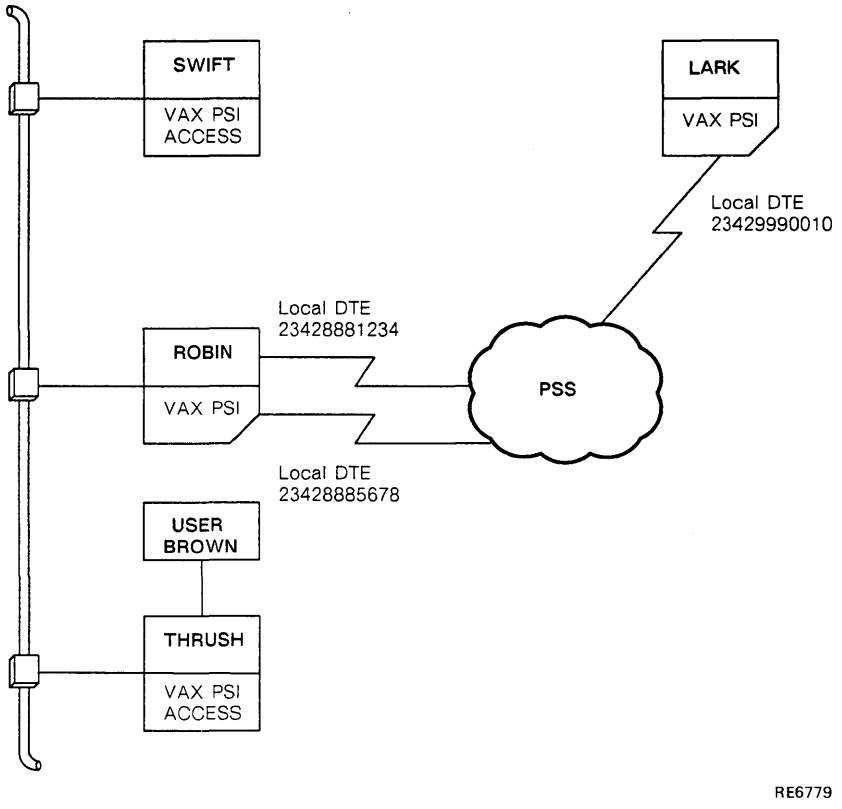
This example shows how to set up PSI Security on an Access node for incoming and outgoing calls via a Connector node.

Note that your Access node may have access to a PSDN via an X25router. This communications product does not support PSI Security. In this case, you must set up PSI Security on your Access node to protect your system from incoming calls from unknown remote DTEs.

Note that if the Connector node via which you have access to a PSDN is a VAX PSI Multi-host node, you should still set up PSI Security on your Access node. You should not rely on PSI Security on the Multi-host node to protect your Access node.

Access node THRUSH is used in this example (see Figure 6–7). Note that the commands are similar to those used in other configurations of VAX PSI.

Figure 6–7 Example 6: Setting Up PSI Security On an Access Node



For example, the following commands are entered on node THRUSH:

To allow incoming calls from node LARK

```
PSI-authorize> ADD/ID VALID_DTE
PSI-authorize> SET LOCAL_DTE 23428881234/NET=PSS -
               _ /ACL=(ID=VALID_DTE,ACCESS=INCOMING)
PSI-authorize> GRANT/ID VALID_DTE/REMOTE_DTE 23429990010
```

These commands create the new rights identifier VALID_DTE on Access node THRUSH, to allow node LARK (that is, remote DTE 23429991234) to make incoming calls.

You can create separate rights identifiers and ACLs to protect destinations on an Access node. The commands are similar to those used on a Multi-host node (see Example 6.5.2). Using this method, you can grant several levels of incoming access from known remote DTEs to different destinations on your system.

To allow outgoing calls from user BROWN to node LARK

```
PSI-authorize> SET REMOTE_DTE 23429990010/NET=PSS -
               _ /ACL=(ID=VALID_USER,ACC=OUT+CHARGE)
PSI-authorize> GRANT/ID VALID_USER /USER BROWN
```

Note that PSI Security on an Access node does not perform a local DTE check for outgoing calls.

PSI Security must be set up on Multi-host node ROBIN to allow outgoing calls from node THRUSH. Example 6 specifies the commands to allow node THRUSH to make outgoing calls via local DTE 23428881234 connected to node ROBIN. Together with these additional commands, user BROWN on node THRUSH can make outgoing calls via node ROBIN to node LARK (that is, to remote DTE 23429991234).

6.5.7 Setting Up PSI Security on a Combination Node

This section explains how to set up security for incoming and outgoing calls to and from Combination nodes. The commands used to set up security on a Combination node are similar to those that apply to Connector and Access nodes.

Combination nodes combine the capabilities of both Access and Connector nodes. They can operate in Access and Multi-host modes and have both VAX PSI and VAX PSI Access installed:

- Using VAX PSI in Multi-host mode, users on a Combination node can access directly the networks to which the combination node is connected.
- Using VAX PSI Access in Access mode, users on a Combination node can access indirectly one or more networks via the Connector nodes to which these networks are connected.

When setting up security on a Combination node, you should distinguish between the agents and objects for each mode of operation. You must grant rights identifiers to the appropriate agents and create ACLs to protect the appropriate objects. However, certain agents and objects may be common to both the Access and Multi-host parts of the Combination node, so that PSI Security can use the same rights identifiers and ACLs for both modes of operation.

To set up security on a Combination node, you must first set up security for the Multi-host part and then for the Access part. The following sections provide guidelines to show you what to do.

Setting Up Security for a Combination Node Operating in Multi-host Mode

In Multi-host mode, your Combination node acts as a Connector node and may have access to one or more networks via one or more local DTEs. In this mode of operation:

- For incoming calls, the agent is a remote DTE and the objects are a local DTE and, optionally, a destination.

  The destination can be either an application on your node, or an application on an Access node connected to your node. In this case, you may want to define the Access node as a destination on your node.
- For outgoing calls, the agent is either a user on your node or an Access node connected to your node, and the objects are a local DTE and a remote DTE.

The commands to set up PSI Security on a Combination node operating in Multi-host mode are similar to the commands to set up PSI Security on a Multi-host node. See Sections 6.5.1 to 6.5.6 for examples of these commands.

Setting Up PSI Security for a Combination Node Operating in Access Mode

In Access mode, your Combination node acts as an Access node and may have access to one or more networks via one or more Connector nodes. In this mode of operation:

- For incoming calls, the agent is a remote DTE and the objects are a local DTE and, optionally, a destination on your node.

  The local DTE is associated with the Connector node that handles the incoming call.

- For outgoing calls, the agent is a user on your node and the object is a remote DTE.

The commands to set up PSI Security on a Combination node operating in Access mode are similar to the commands to set up PSI Security on an Access node. See Sections 6.5.1 to 6.5.7 for examples of these commands.

6.5.8 An Example PSI Security Command File

The PSI Security databases you set up using PSIAUTHORIZE are volatile. You must edit the PSI_SECURITY.COM file in SYS$MANAGER to include the PSI Security commands you have entered using the PSIAUTHORIZE utility. Then, when you reboot your system, the command procedure STARTPSI.COM activates PSI Security by running PSI_SECURITY.COM to reset the databases.

This section contains an example PSI_SECURITY.COM file for node ROBIN, the Multihost node used in the examples described in Section 6.5 and illustrated in Figure 6–1. The command file contains the PSI Security commands applicable to node ROBIN and described in the examples.

This example PSI_SECURITY.COM file restricts incoming and outgoing calls to node ROBIN, so that:

- Incoming calls from some known remote DTEs can have unrestricted incoming access
- Incoming calls from other known DTEs can have restricted access, such as access to mail and X.29 only.
- All calls from unknown remote DTEs (not specified in the command file) will be cleared.
- Outgoing calls from some users can have unrestricted access to known and unknown remote DTEs
- Outgoing calls from other users can have restricted access to known remote DTEs only
- Outgoing calls from Access nodes THRUSH and SWIFT can have restricted access to known remote DTEs only

```
!
! Start of example PSI_SECURITY.COM
!
! This is an example command procedure to show how the PSI Security
! databases can be set up when PSI is loaded
!
! You can enter additional PSIAUTHORIZE  commands after the command
! procedure has run by entering:
!
! $ run sys$system:psiauthorize
!
! Remember to edit the additional commands into PSI_SECURITY.COM
! before you reload PSI
!
! Create the optional rights identifiers used to provide basic security
!
add/id   psi$x25_user
add/id   psi$declname
!
! Create the rights identifiers you wish to use to implement PSI Security
!
add/id   psi_incoming
add/id   psi_wepay
add/id   psi_mail
add/id   psi_x29
add/id   psi_outgoing
add/id   psi_all
add/id   psi_access
!
! Set up a list and grant rights identifiers to known remote DTEs
! that are allowed incoming access
!
! Known Remote DTE 23429990010
!  Allowed unrestricted incoming access
!
grant/id psi_incoming    /remote_dte/net=pss   23429990010
grant/id psi_wepay       /remote_dte/net=pss   23429990010
grant/id psi_mail        /remote_dte/net=pss   23429990010
grant/id psi_all         /remote_dte/net=pss   23429990010
grant/id psi_access      /remote_dte/net=pss   23429990010
!
! Known Remote DTE 23429990011
!  Allowed unrestricted incoming access
!
grant/id psi_incoming    /remote_dte/net=pss   23429990011
grant/id psi_mail        /remote_dte/net=pss   23429990011
grant/id psi_all         /remote_dte/net=pss   23429990011
grant/id psi_access      /remote_dte/net=pss   23429990011
!
! Known Remote DTE 23429990012
!  Allowed unrestricted incoming access
!
grant/id psi_incoming    /remote_dte/net=pss   23429990012
grant/id psi_mail        /remote_dte/net=pss   23429990012
```

```
grant/id psi_all          /remote_dte/net=pss   23429990012
!
! Known Remote DTE 23429990013
!  Allowed restricted incoming access
!
grant/id psi_incoming     /remote_dte/net=pss   23429990013
grant/id psi_mail         /remote_dte/net=pss   23429990013
!
! Known Remote DTE 23429990014
!  Allowed restricted incoming access
!
grant/id psi_incoming     /remote_dte/net=pss   23429990014
grant/id psi_mail         /remote_dte/net=pss   23429990014
!
! Known Remote DTE 23429990015
!  Allowed restricted incoming access
!
grant/id psi_incoming     /remote_dte/net=pss   23429990015
grant/id psi_mail         /remote_dte/net=pss   23429990015
grant/id psi_x29          /remote_dte/net=pss   23429990015
!
! Set up a list and grant rights identifiers to users who are
! allowed:
!
! Restricted outgoing access
!
grant/id psi_outgoing /user scott
grant/id psi_outgoing /user every
grant/id psi_outgoing /user smith
grant/id psi_outgoing /user jones
!
! Unrestricted outgoing access
!
grant/id psi_all /user scott
grant/id psi_all /user every
!
! Set up a list and grant rights identifiers to Access nodes that
! are allowed restricted outgoing access
!
grant/id psi_access /node thrush  ! Access node THRUSH
grant/id psi_access /node swift   ! Access node SWIFT
!
! Protect the LOCAL DTEs for incoming and outgoing calls
!
set local_dte 23428881234 /net=pss /acl=(id=psi_incoming,acc=inc)
set local_dte 23428881234 /net=pss /acl=(id=psi_wepay,acc=inc+reverse)
set local_dte 23428881234 /net=pss /acl=(id=psi_outgoing,acc=out+charge
set local_dte 23428881234 /net=pss /acl=(id=psi_all,acc=inc)
set local_dte 23428881234 /net=pss /acl=(id=psi_access,acc=inc+out+char
!
set local_dte 23428885678 /net=pss /acl=(id=psi_outgoing,acc=out)
!
! Set up LOCAL DTE wildcards
!
```

```
set local_dte *                    /acl=(id=psi_all,acc=out+charge)
!
! Set up LOCAL DTE match-alls
!
set local_dte 23428885678 /net=pss /acl=(id=*,acc=none)
!
! Set up LOCAL DTE CATCHALL
!
set local_dte catchall             /acl=(id=*,acc=none)
!
! Set up DESTINATION security for incoming calls
!
set dest psi_mail    /acl=(id=psi_mail,acc=inc)
set dest x29-server  /acl=(id=psi_x29,acc=inc)
!
! Set up DESTINATION wildcard - ACL is only applied to destinations
! already set up; that is PSI_MAIL and X29-SERVER
!
set dest *           /acl=(id=psi_all,acc=inc+reverse)
!
! Set up ACCESS NODES as DESTINATIONS
!
set dest thrush      /acl=(id=psi_access,acc=inc)     ! Access node THRUSH
set dest swift       /acl=(id=psi_access,acc=inc)     ! Access node SWIFT
!
! Set up DESTINATION /CATCHALL
!
set destination/catchall            /acl=(id=*,acc=none)
!
! Set up the access to known REMOTE DTEs for restricted outgoing calls
!
set remote_dte 23429990010 /net=pss/acl=(id=psi_outgoing,acc=out+charge)
set remote_dte 23429990011 /net=pss/acl=(id=psi_outgoing,acc=out+charge)
set remote_dte 23429990012 /net=pss/acl=(id=psi_outgoing,acc=out+charge)
set remote_dte 23429990013 /net=pss/acl=(id=psi_outgoing,acc=out+charge)
set remote_dte 23429990014 /net=pss/acl=(id=psi_outgoing,acc=out+charge)
set remote_dte 23429990015 /net=pss/acl=(id=psi_outgoing,acc=out+charge)
!
! Set up access to remote DTEs within PSS for outgoing calls from
! trusted users only
!
set remote_dte 2342 /new   /net=pss/acl=(id=psi_all,acc=out+charge)
!
! Set up the access to known remote DTEs for outgoing calls from
! Access nodes
!
set remote_dte 23429990010 /net=pss/acl=(id=psi_access,acc=out+charge)
!
! Set up the REMOTE DTE CATCHALL
!
set remote_dte catchall    /net=pss/acl=(id=*,acc=none)
!
! End of example PSI_SECURITY.COM
!
```

7

PSI Security Commands

This chapter describes, in alphabetical order, the PSI Security commands and any associated qualifiers. Examples of how the commands and qualifiers are used are also provided.

The DCL rules for entering commands, described in the *VMS DCL Dictionary*, apply to the PSI Security commands. Thus, you can abbreviate any command, keyword or qualifier provided the abbreviation is not ambiguous. The percent sign (%) is not an allowed wildcard.

NOTE

When you are using the security commands, do not mix wildcards and other characters in the same parameter. Use either a wildcard (*) or ASCII characters, but not both. Note that the wildcard character cannot be used with the /USER qualifier.

ADD/IDENTIFIER

This command creates a new rights identifier and adds it to the System Rights Database.

Format

ADD/IDENTIFIER *identifier*

Command Parameters

identifier

Specifies the rights identifier to be added to the System Rights Database. The identifier is a string of 1 through 32 alphanumeric characters. It must contain at least one non-numeric character and may contain underscores and dollar signs.

Command Qualifiers

None

Example

```
PSI-authorize> ADD/ID UK_PSS
```

This example creates the new rights identifier UK_PSS and adds it to the System Rights Database.

```
PSI-authorize> ADD/ID PAYROLL
```

In this example, the new rights identifier PAYROLL is created and added to the System Rights Database.

Notes

If you try and create an identifier that already exists, you will receive an error message.

See the *Guide to VMS Security* for more information about the System Rights Database.

DEFINE/KEY

This command associates an equivalence string and a set of attributes with a key on the terminal keyboard.

Format

DEFINE/KEY *key-name* *equivalence-string*

Command Parameters

key-name

Specifies the name of the key you are defining.

equivalence string

Specifies the string which is to be processed when you press the key. If the string contains any spaces, enclose the equivalence string in quotation marks.

Description

This command enables you to assign definitions to the keys on certain terminals. The terminals include VT52, VT100 series and terminals with LK201 keyboards.

To define keys on the numeric pads of these terminals you must first issue the DCL commands SET TERMINAL/APPLICATION or SET TERMINAL/NONUMERIC.

Note that default key definitions are loaded from PSIAUTHORIZE$INIT.

Command Qualifiers

/ECHO
/NOECHO

Determines whether or not the equivalence string is displayed on your screen after the key has been pressed. The default is /ECHO. You cannot use /NOECHO with the /NOTERMINATE qualifier.

/IF_STATE= { *state-name* / *state-name,...* }
/NOIF_STATE

Specifies a list of one or more states, one of which must be in effect for the key definition to be valid. If you omit the /IF_STATE qualifier or use the /NOIF_STATE one the current state is used. The state name is an alphanumeric string. States are established with the /SET_STATE qualifier. If you specify only one state name you can omit the parentheses.

/LOCK_STATE
/NOLOCK_STATE

Specifies that the state set by the /SET_STATE qualifier remains in effect until explicitly changed. If you use the /NOLOCK_STATE qualifier the state set by /SET_STATE is in effect only for the next definable key that you press, or for the next read terminating character you type.

The default is /NOLOCK_STATE. The /LOCK_STATE qualifier can only be specified with the /SET_STATE qualifier.

/SET_STATE={*state-name*}
/NOSET_STATE

Causes the specified state-name to be set when the key is pressed. The state name can be any alphanumeric string.

If you omit this qualifier or use /NOSET_STATE the current state that was locked remains in effect.

/TERMINATE
/NOTERMINATE

Specifies whether or not the current equivalence string is to be terminated (that is, processed) when the key is pressed. The default is /NOTERMINATE, which allows you to press other keys before the equivalence string is processed. Pressing RETURN has the same as effect as using /TERMINATE.

Example

```
PSI-authorize> DEFINE/KEY PF3 "SHOW DEST *" /TERMINATE
```

This example defines the key PF3 so that it performs the `SHOW DEST *` command.

```
PSI-authorize> DEFINE/KEY/IF_STATE=ONE PF1 "ONE"
```

This command defines the PF1 key to be "ONE" for state ONE. The state is specified in the same command as the key definition. This is the preferred method for defining keys.

GRANT/IDENTIFIER

This command is used to grant an identifier to a specified agent. The agent is either a user (/USER), remote DTE (/REMOTE_DTE), or an Access node (/NODE). The default is /USER.

Format

GRANT/IDENTIFIER *identifier* { *agent* } { *command qualifiers* }

Command Parameter

identifier

Specifies the identifier to be granted.

{*agent*}

Specifies the agent to whom you are granting the rights identifier.

Valid *agent* options are:

{*user-name*}	to specify a user
{*dte-address*} { * }	to specify remote DTEs
{*CATCHALL*}	to specify the remote DTE CATCHALL entry
{*node-name*} { * }	to specify Access nodes

If you use the wildcard option, the specified rights identifier is granted to all agents that are defined currently in the appropriate Agent Rights Database. The agent type is determined by the agent qualifier. Note that you cannot specify the wildcard option with the /USER qualifier.

Command Qualifiers

/NETWORK= { *network name* / * }

Must be used with the /REMOTE_DTE qualifier to specify the network with which the remote DTE is associated. However, /NETWORK may be omitted if there is only one network associated with your system. The wildcard option (*) specifies all networks that are defined currently.

/CATCHALL

Can be used with the /NODE qualifiers to specify an entry to be used if the requested entry is not found.

/NODE

Specifies that the agent is an Access node.

/REMOTE_DTE

Specifies that the agent is a remote DTE.

/USER

Specifies that the agent is a user. /USER is the default.

Example

```
PSI-authorize> GRANT/ID SEND_CALLS/USER SCOTT
```

This example grants the identifier SEND_CALLS to user SCOTT in the User Rights Database.

```
PSI-authorize> GRANT/ID MAIL_USE */REMOTE_DTE/NET=PSS
```

This example shows how to grant the identifier MAIL_USE to all remote DTEs that belong to the PSS network in the Remote DTE Rights Database.

Notes

If the user you specify is not present in the User Rights Database, you will receive an error message. Note that you cannot specify the wildcard character (*) with the /USER qualifier.

When specifying a remote DTE CATCHALL entry, you must specify the /NETWORK qualifier if more than one network is associated with your system.

REMOVE/IDENTIFIER

This command deletes an existing rights identifier from the System Rights Database.

Format

REMOVE/IDENTIFIER *identifier*

Command Parameters

identifier

Specifies the identifier to be deleted from the System Rights Database.

Command Qualifiers

None

Example

```
PSI-authorize> REMOVE/ID MAIL_USE
```

This example deletes the rights identifier MAIL_USE from the System Rights Database.

```
PSI-authorize> REMOVE/ID PAYROLL
```

In this example, the rights identifier PAYROLL is deleted from the System Rights Database.

Notes

It is advisable to remove ACL entries containing the rights identifier from the Object Access Control Databases before you delete the rights identifier itself.

If the rights identifier is not in the System Rights Database, you will receive an error message.

REVOKE/IDENTIFIER

This command revokes a rights identifier from a specified agent in the appropriate Agent Rights Database, according to the agent command qualifier. The agent can be a user (/USER), a remote DTE (/REMOTE_DTE) or an access node (/NODE). The default is /USER.

Format

REVOKE/IDENTIFIER *identifier* { *agent* } { *command-qualifiers* }

Command Parameters

identifier

Specifies the rights identifier to be revoked.

{*agent*}

Specifies the agent from whom you are revoking the rights identifier. Valid *agent* options are:

{*user-name*}	to specify a user
{*dte-address*} { * }	to specify remote DTEs
{*CATCHALL*}	to specify the remote DTE CATCHALL entry
{*node-name*} { * }	to specify Access nodes

If you use the wildcard option, the specified rights identifier is revoked from all the agents that are defined currently in the appropriate Agent Rights Database. The agent type is determined by the agent qualifier. Note that you cannot specify the wildcard option with the /USER qualifier.

Command Qualifiers

/CATCHALL Can be used with the /NODE qualifiers to specify an entry to be used if the requested entry is not found.

/NETWORK= { *network name* / * }

Must be used with the /REMOTE_DTE qualifier to specify the network with which the remote DTE is associated. However, /NETWORK may be omitted if there is only one network associated with the system. The wildcard option (*) specifies all networks that are defined currently.

/NODE

Specifies that the agent is an Access node.

/REMOTE_DTE

Specifies that the agent is a remote DTE.

/USER

Specifies that the agent is a user. /USER is the default.

Example

```
PSI-authorize> REVOKE/ID WE_PAY/USER SCOTT
```

In this example the identifier WE_PAY is revoked from user Scott.

```
PSI-authorize> REVOKE/ID SEND_CALLS/REMOTE_DTE 2342913132
```

This example revokes the identifier SEND_CALLS from remote DTE 2342913132.

Note

If the agent you specify is not present in the appropriate Agent Rights Database, or the agent does not posses the identifier you specified, you will receive an error message.

SET DESTINATION

This command creates, modifies or deletes an ACL in the Destination Access Control Database.

Format

SET DESTINATION { *destination-name* | /CATCHALL | * } { *command qualifiers* }

Command Parameters

destination-name

Specifies the destination or list of destinations in the Destination Access Control Database.

The wildcard option (*) specifies all destinations that are defined currently in the database, including the destination CATCHALL entry if there is one.

Command Qualifiers

/ACL=*acl-list*

Creates an ACL. If there is already an ACL on the destination, the new ACL is added to the beginning of the existing ACL.

/AFTER=*acl-entry*

Allows the position of the new ACL to be situated after the existing ACL. This qualifier requires that /ACL has been previously specified.

/CATCHALL

Specifies the destination CATCHALL entry in the database.

/DELETE

Deletes the ACL from the specified destination.

/DECLARED_NAME

Specifies that *destination-name* is the name of a destination record created by running a declared network process that handles incoming calls. The destination record contains information about the local DTE subaddress, the remote DTE address, the user data field and the user group identification.

This qualifier cannot be used with the /CATCHALL qualifier.

/LIKE={*destination-name*}

Allows an existing ACL from another specified destination to replace the ACL of this destination.

/NEW

Deletes an existing ACL and replaces it with the new ACL specified by the /ACL qualifier.

Example

```
PSI-authorize> SET DESTINATION MAIL/ACL=(IDENTIFIER= -
               _ MAIL_USE,ACCESS=INCOMING)
```

This example shows how to set-up an ACL for the destination MAIL.

```
PSI-authorize> SET DESTINATION MAIL/DELETE
```

This example deletes the ACL for the MAIL destination.

Note

If the destination has not been defined (by NCP), you will receive an error message. Refer to Chapters 4 and 5 for details on the Destination Access Control Database.

SET LOCAL_DTE

This command creates, modifies or deletes an ACL in the Local DTE Access Control Database, according to the qualifiers specified.

Note that you should **not** use the SET LOCAL_DTE command to create an ACL for outgoing calls from an Access node. If you create such an ACL, it has no effect. PSI Security on an Access node does not perform a local DTE check for outgoing calls.

If your system has more than one local DTE associated with more than one network, you must use the /NETWORK qualifier.

If you create an ACL using the CATCHALL option, the access action specified in the ACL is applied if a requested entry cannot be found in the database. You can use the /NETWORK qualifier to create a local DTE CATCHALL option for each network associated with your system.

If an entry for the specified local DTE does not exist, the command creates one.

Format

SET LOCAL_DTE { *dte-address* | CATCHALL | * } { *command-qualifiers* }

Command Parameters

dte-address

Specifies the local DTE address, or list of local DTE addresses, to be set in the Local DTE Access Control Database.

CATCHALL specifies a local DTE CATCHALL entry in the database.

If the wildcard option (*) is used, the action is applied to all local DTEs that are defined currently in the Local DTE Access Control Database, including CATCHALL entries, if there are any.

Command Qualifiers

/ACL=*acl-list*

Creates the ACL if there is no ACL for the local DTE already. If there is, the new ACL is added to the end of the existing ACL list.

/AFTER=*acl-entry*

Allows the position of the new ACL to be situated after the existing ACL. This qualifier requires that /ACL has been previously specified.

/DELETE

Deletes the ACL from the specified local DTE.

/LIKE=*dte-address*

Allows an existing ACL from another specified local DTE to replace the ACL of this local DTE

/NEW

Deletes an existing ACL and replaces it with the new ACL specified by the /ACL qualifier.

/NETWORK=*network-name*

Must be used to specify the network associated with a local DTE if your system has more than one local DTE associated with more than one network.

For Native or Multi-host nodes, use the /NETWORK qualifier to specify the network associated with the local DTE(s) via which your node sends and receives calls. The *network-name* must be defined in the X25-PROTOCOL database on your node.

For Access nodes, use the /NETWORK qualifier to specify the network associated with the local DTE(s) on the Connector node via which your Access node receives calls. The *network-name* must be defined in the X25-ACCESS database on your node.

You can use the /NETWORK qualifier to create a local DTE CATCHALL option for each network associated with your system. If you omit the /NETWORK qualifier, the CATCHALL applies to all local DTEs associated with all networks on your system. Note that if you define both forms of local DTE CATCHALL, PSI Security will select a network specific local DTE CATCHALL in preference to the global local DTE CATCHALL.

Example

```
PSI-authorize> SET LOCAL_DTE */ACL=(IDENTIFIER=WE_PAY, -
               _ ACCESS=INCOMING+REVERSE_CHARGE)
```

This example places the ACL entry specified, at the head of each ACL for all local DTEs in the Local DTE Access Control Database.

```
PSI-authorize> SET LOCAL_DTE 234273417171 -
               _ /LIKE=255367783421/NET=PSS
```

This command sets the ACL of the first local DTE to that of the second local DTE.

Note

Refer to Chapters 4 and 5 for details about the Local DTE Access Control Database.

SET LOCAL_NODE

This command creates, modifies or deletes an ACL for a local node. This command can be used for systems which have only one local DTE, or for systems in which there are no special security requirements for each local DTE or node.

The SET LOCAL_NODE command is equivalent to the SET LOCAL_DTE CATCHALL command.

Format

SET LOCAL_NODE { *command-qualifiers* }

This command applies to the local node and does not require any command parameters.

Command Qualifiers

/ACL=*acl-list*

Creates an ACL if there is no ACL for the local node already. If there is, the new ACL is added to the end of the existing ACL list.

/AFTER=*acl-entry*

Allows the position of the new ACL to be situated after the existing ACL. This qualifier requires that /ACL has been previously specified.

/DELETE

Deletes the ACL for the local node.

/NEW

Deletes an existing ACL and replaces it with the new ACL specified by the /ACL qualifier.

Example

```
PSI-authorize> SET LOCAL_NODE /ACL=(IDENTIFIER=MAIL_USE, -
```

This example places the ACL entry specified at the top of the ACL for the local node.

Notes

For all nodes, the SET LOCAL_NODE command creates a CATCHALL entry in the Local DTE Access Control Database.

Refer to Chapters 4 and 5 for details about the Local DTE Access Control Database.

SET REMOTE_DTE

This command creates, modifies or deletes an ACL in the Remote DTE Access Control Database, according to the qualifiers specified.

You must use the /NETWORK qualifier to specify the network associated with the remote DTE. Note that the network associated with a remote DTE must be that associated with the local DTE which handles the incoming and outgoing calls to the remote DTE.

For Multi-host and Native nodes, the *network-name* must be defined in the X25-PROTOCOL database.

For Access nodes, the *network-name* must be defined in the X25-ACCESS database.

If you create an ACL using a CATCHALL option, the access action specified in the ACL is applied if a requested entry cannot be found in the database. The CATCHALL option must be used with the /NETWORK qualifier for remote DTEs.

If an entry for the specified remote DTE does not exist, the command creates one.

Format

SET REMOTE_DTE { *dte-address* | CATCHALL | * } { *command-qualifiers* }

Command Parameters

dte-address

Specifies the remote DTE address, or list of addresses, to be set in the Remote DTE Access Control Database.

CATCHALL specifies a remote DTE CATCHALL entry in the database.

The wildcard option (*) specifies all remote DTEs as well as all CATCHALL entries that are defined currently in the database.

These command parameters must be qualified by /NETWORK if more than one network is associated with your system.

Command Qualifiers

/ACL=*acl-list*

Creates the ACL if there is no ACL for the remote DTE already. If there is, the new ACL is added to the end of the existing ACL list.

/AFTER=*acl-entry*

Allows the position of the new ACL to be situated after the existing ACL. This qualifier requires that /ACL has been previously specified.

/DELETE

Deletes the ACL from the specified remote DTE.

/LIKE=*dte-address*

Allows an existing ACL from another specified remote DTE to replace the ACL of this remote DTE.

/NEW

Deletes an existing ACL and replaces it with the new ACL specified by the /ACL qualifier.

/NETWORK= { *network name* | * }

Must be used to specify which network is to be associated with the remote DTE or remote DTE CATCHALL entry. Note that the network associated with a remote DTE must be that associated with the local DTE which handles the incoming and outgoing calls to the remote DTE.

For Multi-host and Native nodes, the *network-name* must be defined in the X25-PROTOCOL database.

For Access nodes, the *network-name* must be defined in the X25-ACCESS database.

The wildcard option (*) specifies all networks that are defined currently.

Examples

```
PSI-authorize> SET REMOTE_DTE */NET=PSS/ACL=(IDENTIFIER= -
               _ MAIL_USE,ACCESS=OUTGOING+CHARGE)
```

This example places the ACL entry specified, at the head of each ACL for remote DTEs in PSS in the Remote DTE Access Control Database.

```
PSI-authorize> SET REMOTE_DTE 2342913147 -
               _ /LIKE=2553673421/NET=PSS
```

This command replaces the ACL of the first remote DTE with that of the second remote DTE.

Note

Refer to Chapters 4 and 5 for details about the Remote DTE Access Control Database.

SHOW DESTINATION

This command displays the ACLs, if any, on a destination or a list of destinations in the Destination Access Control Database.

Format

SHOW DESTINATION { *destination-name* | * | /CATCHALL } { *command-qualifiers* }

Command Parameter

destination-name

Specifies the destination to be shown in the Destination Access Control Database.

The wildcard option (*) specifies all destinations that are defined currently in the database, including the CATCHALL destination if there is one.

Command Qualifiers

/CATCHALL

Used to specify that the destination CATCHALL entry in the database is to be displayed, if there is one. This use of this qualifier in the SHOW DESTINATION command requires that a SET DESTINATION /CATCHALL entry has been defined previously.

/DECLARED_NAME

Specifies that *destination-name* is the name of a destination record created by running a declared network process that handles incoming calls. The destination record contains information about the local DTE subaddress, the remote DTE address, the user data field and the user group identification.

This qualifier cannot be used with the /CATCHALL qualifier.

Example

```
PSI-authorize> SHOW DESTINATION PSI-MAIL
```

This example displays the ACL on the destination PSI-MAIL, as follows:

```
  Destination:  PSI-MAIL
        (IDENTIFIER=MAIL_USE,ACCESS=INCOMING)
        (IDENTIFIER=MAIL_WEPAY,ACCESS=INCOMING+REVERSE_CHARGE)
```

SHOW/IDENTIFIER

This command displays information about the specified rights identifier in the System Rights Database.

Format

SHOW/IDENTIFIER { *identifier* | * }

Command Parameter

identifier

Specifies the rights identifier in the System Rights Database.

The wildcard option (*) specifies all identifiers that are defined currently in the database.

Command Qualifiers

/BRIEF

Gives brief information about rights identifiers.

/FULL

Gives full details about rights identifiers.

Example

```
PSI-authorize> SHOW/IDENTIFIER PSI$X25_USER
```

This example shows information about the identifier PSI$X25_USER, as follows:

```
  Name                    Value                   Attributes
  PSI$X25_USER            %X8001000D              NORESOURCE
```

SHOW LOCAL_DTE

This command displays the ACLs, if any, for the specified local DTE in the Local DTE Access Control Database, according to the command qualifiers specified.

If your system has more than one local DTE associated with more than one network, you must use the /NETWORK qualifier to display the ACL for a local DTE associated with that network.

You can use the CATCHALL option to display the local DTE CATCHALL entry. If you have defined CATCHALL entries for more than one network, use the /NETWORK qualifier to display the relevant local DTE CATCHALL entry.

Format

SHOW LOCAL_DTE { *dte-address* | CATCHALL | * } { *command-qualifier* }

Command Parameter

dte-address

Specifies the local DTE address or list of local DTE addresses to be shown.

CATCHALL specifies a local DTE CATCHALL entry, if one has been defined already.

The wildcard option (*) specifies all local DTEs that are defined currently in database, including CATCHALL entries, if there are any.

Command Qualifier

/NETWORK=*network-name*

Must be used to specify which network is associated with the local DTE if your system has more than one local DTE associated with more than one network.

You can use the /NETWORK qualifier to display the local DTE CATCHALL entry for the network if you have defined such entries for more than one network.

Example

```
PSI-authorize> SHOW LOCAL_DTE 23429225421/NETWORK=PSS
```

This example shows any ACLs for local DTE 23429225421 on the PSS network, as follows:

```
  Network:  PSS  DTE:  23429225421
        (IDENTIFIER=PSS_INCOMING,ACCESS=INCOMING)
        (IDENTIFIER=PSS_OUTGOING,ACCESS=OUTGOING)
```

SHOW LOCAL_NODE

This command displays the ACL, if there is one, for a local node. It does not require any command parameters or command qualifiers.

SHOW LOCAL_NODE is equivalent to SHOW LOCAL_DTE/CATCHALL.

Format

SHOW LOCAL_NODE

Command Parameters

None.

Command Qualifiers

None.

Example

```
PSI-authorize> SHOW LOCAL_NODE
```

This example shows the ACL for the local node, as follows:

```
  Local Node: node-name

        (IDENTIFIER=PSS_INCOMING,ACCESS=INCOMING)
        (IDENTIFIER=PSS_OUTGOING,ACCESS=OUTGOING)
```

SHOW REMOTE_DTE

This command displays the ACLs, if any, for the specified remote DTE or list of remote DTEs in the Remote DTE Access Control Database.

The CATCHALL option can be used with the /NETWORK qualifier to specify the remote DTE CATCHALL entry associated with that network. (Remote DTEs are grouped in the Remote DTE Access Control Database according to the network with which they are associated. Each grouping can have a CATCHALL entry.)

Format

SHOW REMOTE_DTE { *dte-address* / CATCHALL / * } { *command-qualifier* }

Command Parameter

dte-address

Specifies the remote DTE address or list of remote DTE addresses to be displayed, qualified by /NETWORK.

CATCHALL specifies a remote DTE CATCHALL entry in the database, qualified by /NETWORK.

If the wildcard option (*) is used, all remote DTEs that are currently defined in the database, including all CATCHALL entries, are displayed.

Command Qualifier

/NETWORK= { *network name* / * }

Must be used to specify which network is associated with the remote DTE or remote DTE CATCHALL entry. The wildcard option (*) specifies all networks that are defined currently.

Example

```
PSI-authorize> SHOW REMOTE_DTE 23429114321/NETWORK=PSS
```

This example shows any ACLs for remote DTE 23429114321 on the PSS network, as follows:

```
  Network:  PSS  DTE:  23429114321
        (IDENTIFIER=PSS_WEPAY,ACCESS=INCOMING+REVERSE_CHARGE)
        (IDENTIFIER=PSS_NOPAY,ACCESS=INCOMING)
```

SHOW/RIGHTS

This command displays the rights identifiers held by the specified agent in the appropriate Agent Rights Database. The agent is either a user (/USER), a remote DTE (/REMOTE_DTE), or an Access node (/NODE). The default is /USER.

Format

SHOW/RIGHTS *agent* { *command-qualifiers* }

Command Parameter

agent

Specifies the agent whose rights identifiers you wish to display. Valid *agent* options are:

{*user-name*}	to specify a user
{*dte-address*} { * }	to specify remote DTEs
{*CATCHALL*}	to specify the remote DTE CATCHALL entry
{*node-name*} { * }	to specify Access nodes

If you specify the wildcard option, all rights identifiers for all agents that are currently defined in the appropriate Agent Rights Database are displayed. The agent type is determined by the agent qualifier. Note that you cannot specify the wildcard option with the /USER qualifier.

Command Qualifiers

/CATCHALL

Can be used with the /NODE command qualifier to specify the Access Node CATCHALL entry.

/NODE

Specifies that the agent is an Access node.

/REMOTE_DTE

Specifies that the agent is a remote DTE.

/USER

Specifies that the agent is a user.

/NETWORK= { *network name* | * }

Must be used with the /REMOTE_DTE qualifier to specify the network with which the remote DTE is associated. However, /NETWORK may be omitted if there is only one network associated with the system. The wildcard option (*) specifies all networks that are currently defined.

Example

```
PSI-authorize> SHOW/RIGHTS /USER SCOTT
```

This example shows the rights identifiers belonging to the user Scott, as follows:

```
Identifier                      Value               Attributes
  PSI$PSI                       %X80010003          NORESOURCE
  PSI$X25_USER                  %X8001000D          NORESOURCE
```

III

PSI Accounting

This part of the manual consists of four chapters:

- Chapter 8 - Provides an overview of the PSI Accounting facilities and how to use them.
- Chapters 9, 10 and 11 - Provide reference information on using PSI Accounting.

8

Introduction to PSI Accounting

8.1 Overview of PSI Accounting

When VAX PSI is used you can, optionally, record details of the way in which it is used. This may be so that you can charge users for X.25 resources or it may simply be for performance records. The following data can be recorded:

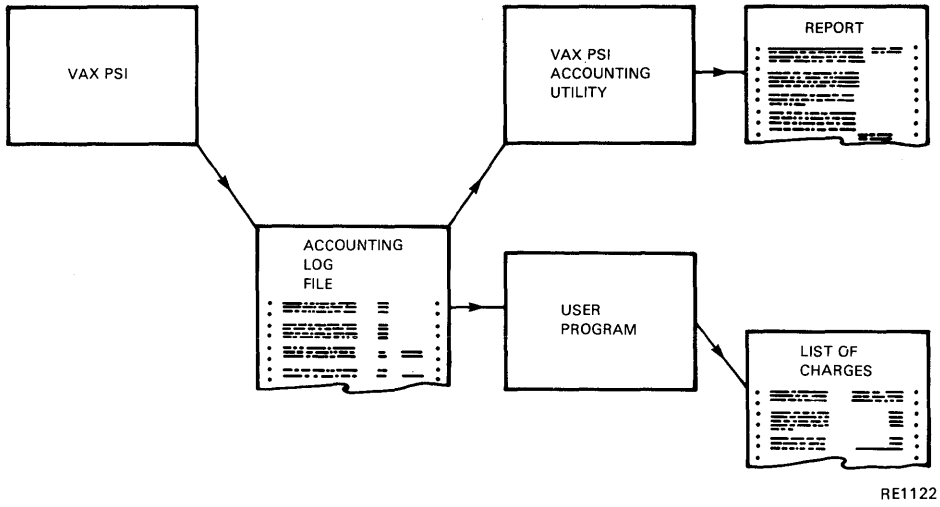
- Data to allow you to calculate the cost of any call.
- Data to allow you to determine who was using the network at any given time (including the remote DTE involved).
- All calls and attempts to call (including failed outgoing access) and all access to Permanent Virtual Circuits (PVCs), on VAX PSI Native, Multi-host and Access systems.

When a set of accounting records have been produced, you can run the PSI Accounting Utility (ACCOUNTING/PSI) to produce a report. Alternatively, you can process the records with your own program.

A sample user accounting program is given in Appendix A. You can use this as provided or adapt the program to suit your own requirements.

Details of generating the accounting records are given in Section 8.3. The accounting record formats are described in Chapter 11.

Figure 8–1 Accounting Options



8.2 The ACCOUNTING/PSI Utility

ACCOUNTING/PSI was derived from the VMS Accounting Utility (ACCOUNTING) to provide a means of processing VAX PSI records and packets. All the VMS command qualifiers are allowed with ACCOUNTING/PSI. However, since PSI does not use all of the fields available with VMS, some of the qualifiers will produce no output (see Chapter 10).

ACCOUNTING/PSI produces four types of output:

- A brief listing of selected records
- A full listing of selected records
- A binary copy of selected and/or rejected records
- A summary report of selected items from selected records

These can be displayed on a terminal or written to a disk or tape file.

ACCOUNTING/PSI processes accounting log data by sharing the currently open log file on a running system or by receiving as input a previously recorded accounting file. All input data must be in binary format.

Use of ACCOUNTING/PSI requires read access to the input accounting file.

8.3 Producing Accounting Records

You specify whether accounting records are produced or not with a command procedure, SYS$MANAGER:PSIACCOUNTING.COM. The command procedure also lets you specify where the records are written to.

You enter

```
@SYS$MANAGER:PSIACCOUNTING
```

followed by one of the commands: ON, OFF, OPEN, CLOSE or SHOW. The commands have the following meaning:

1. ON - Close any current file then start producing accounting records and write them to the file SYS$MANAGER:PSIACCOUNTING.DAT. (This is the default output file.)
2. OFF - Stop accounting record production and close the output file.
3. OPEN *file-name* - Close any current file then start producing accounting records and write them to the file *file-name*.
4. CLOSE - Stop accounting record production and close the output file. This is the same as `OFF`.
5. SHOW - Display an OPCOM message giving the current state of VAX PSI accounting and the current output file name.

ON and OPEN will both close any current output file before opening a new file. No accounting information can be lost between closing the old file and opening the new one.

VAX PSI calls will not be logged if they start with PSIACCOUNTING ON and finish with PSIACCOUNTING OFF.

NOTE

You need SYSPRV privilege to be able to use this command procedure.

You should enable your terminal as follows before using the command procedure:

```
$ REPLY /ENABLE=NETWORK
```

This enables your terminal as a network operator and lets you see OPCOM messages confirming that the requested operations have been successful.

9

Using the ACCOUNTING/PSI Utility

9.1 Starting and Exiting the Utility

The following DCL command starts the utility:

```
$ ACCOUNTING/PSI [file-spec[,...]]
```

Each time you issue the command ACCOUNTING/PSI, it executes a single accounting request. Generally, each request runs until it is complete. However, you can press CTRL/Y and type EXIT to terminate ACCOUNTING/PSI earlier. Pressing CTRL/Y and typing EXIT terminates ACCOUNTING/PSI normally, ensuring that all files are handled properly.

Any VMS user with sufficient VMS privilege to access the accounting file may use ACCOUNTING/PSI.

9.2 Command Qualifiers

The command qualifiers of the ACCOUNTING/PSI command (described in Chapter 10) let you select records from the PSI accounting log file and specify the output format. By default, output is directed to the SYS$OUTPUT device. However, you can specify a separate output file with the /OUTPUT qualifier.

You can further specify whether the output should be in binary or ASCII format. If you specify /BINARY, a binary accounting file is produced that can later be processed with other accounting commands. However, if you specify /FULL or /SUMMARY, or assume the default, an ASCII file is produced.

You select records based on fields in the records and their values. A large number of qualifiers define the field names for selection. Every time you select records, you create a group of one or more unselected records that you can optionally store in a binary output file using the /REJECTED qualifier. You can also sort the records prior to listing them.

9.3 Outputs

The forms of output are listing and binary output.

9.3.1 Listing Output

Listing output consists of screen images. Any terminal supported by VMS with dimensions of at least 80 columns by 24 rows can be used. (You may have to issue the DCL command SET TERMINAL to set the proper dimensions.) Alternatively, listing output can be routed to a file for subsequent printing.

There are three basic screen formats used for displaying data: the brief listing, the full listing, and the summary report. These screen formats are described in the following sections, with a sample of each.

9.3.1.1 Brief Listing Format

The brief listing format provides one line for each record in the accounting file being processed. The output always includes the date and time, the type of record, the subtype, the user name, the ID, the source, and the status.

9.3.1.2 Full Listing Format

The full listing format provides all the data for all selected records in the accounting file being processed. There are small variations in the record formats, based on the presence or absence of data in the record. When the output is directed to a terminal the first screen image of 24 lines of the selected record is displayed and the utility provides a prompt for display of the next screen image. N is the default response for this prompt, in which case the first screen image of the next record is displayed. This does not apply if /OUTPUT=*file* is selected.

9.3.1.3 Summary Listing Format

Summary output is an ASCII file consisting of the specified report-item values grouped by the specified summary items. The summary file reflects the accumulation of these report-items throughout the ACCOUNTING/PSI period requested. The statistics in the summary output are either totals or maximum values of the report-items (See the /REPORT qualifier for details).

The summary output format includes the dates of the accounting period on the first line. The start date appears at the left, optionally followed by the title specified with the /TITLE qualifier. The end date for the report appears at the right margin.

9.3.2 Binary Output Files

A binary output file is itself an accounting log file that is created when an ACCOUNTING/PSI request includes the /BINARY or /REJECTED qualifier. This file contains a subset of the accounting records from the input (source) accounting files. The subset of records included depends on the selection criteria. All resulting files can be used as source data for further reports or summaries, or as input for the ACCOUNTING/PSI Utility with different selection criteria. These files provide you with a set of pre-selected records for use with your own user-written utility.

With the /BINARY qualifier, all selected records are recorded.

The /SORT qualifier gives a sorted output.

With the /REJECTED qualifier, only those records not being selected are directed to this file.

A complete description of the record formats of the log file appear in Chapter 11.

9.4 The Information Provided By ACCOUNTING/PSI

This section gives a brief description of the information provided by ACCOUNTING/PSI. All the fields described below can be displayed using the Full Listing facility. The fields are described in the order in which they are displayed and can be considered as consisting of two groups:

1. Those fields that provide information about the process using VAX PSI.
2. Those fields that provide information about how VAX PSI was used.

9.4.1 Information About the Process Using VAX PSI

For calls handled internally (for example, incoming X.29 calls) the information refers to PSI. See Section 9.4.3 for information about the ACCOUNTING/PSI Utility and incoming X.29 calls.

Username

The name that a user typed to log on to the system, or NETACP for a DLM call.

Account

A field defined by the system manager, to accumulate data on the use of VAX PSI resources.

Process ID

The unique VMS identifier of a process.

Owner ID

The identifier of the process that created the subprocess.

Terminal name

For interactive processes this identifies the terminal using the process. For all other processes this field is blank.

UIC

The User Identification Code of the process.

Priority

The priority of the process (at the time the call was initiated).

Privilege

The privileges of the process.

9.4.2 Information on How VAX PSI is Used

Remote node name

The name of the remote node involved in using VAX PSI Access or VAX PSI Multi-host.

Remote ID

The identifier of the user at the remote node. (At a VAX PSI Access node this will always be PSI.)

Finish time

The time the virtual circuit was terminated.

Start time

The time the virtual circuit was established. This field contains a value only if the connection was successful.

Elapsed time

The difference between the Start time and the Finish time (if the connection was successful).

Bytes sent/received

The counts of the data bytes sent/received. These are used for charging purposes.

Segments sent/received

The counts of the data segments sent/received. A segment is a charging unit which is 64 bytes for most PSDNs. For example, a 64 byte packet would be charged as one segment but a 65 byte packet would be charged as two segments.

See the PSDN literature, to find out the appropriate segment definition.

Packets sent/received

The counts of the data packets sent/received.

Messages sent/received

The counts of the data messages sent/received. A message is a sequence of packets with the more bit set, plus a final packet. Messages are therefore the largest charging unit for data sent or received.

Remote DTE

For outgoing Switched Virtual Circuits (SVCs), this is the DTE address called. For incoming SVCs it is the address of the calling DTE, if that information is provided to PSI by the network. Otherwise, the field is blank.

Local DTE

For outgoing SVCs, this is the address of the calling DTE and for incoming SVCs it is the address of the DTE called. For PVCs, the field is blank.

CUG number

This is the number for a Closed User Group used. This number is assigned by the network when the CUG is subscribed to.

Network

The network identifier. If only one network is available, this field may be blank.

Destination

For an incoming SVC this is the name of the destination receiving the call. If the call has been redirected, it is the name of the last destination to receive the call. For PVCs this field is blank.

Protocol ID

The first four bytes of call data. CCITT recommend these be used to specify the desired protocol (for example, X.29 calls start with 01).

Circuit type

The types of circuit used: the types are as follows;

- Outgoing or Incoming
- Switched Virtual circuit (SVC) or Permanent Virtual Circuit (PVC)
- Unsuccessful
- PSI access or Multi-host
- X.29 (incoming)

Facilities

The facilities requested for the circuit. These are as follows;

- Reverse charging
- Fast select
- CUG
- Bilateral CUG (group of two DTEs)

Clearing reason

An explanatory text about circuit clearance.

Clearing cause

Information from the clear packet sent by the PSDN about the circuit clearance (see your network documentation).

Inc/out throughput class

The incoming and outgoing throughput requested from the PSDN. This may affect the charging rate.

Inc/out packet size

The incoming and outgoing packet size requested from the PSDN. This may affect the charging rate.

Inc/out window size

The incoming and outgoing window size requested from the PSDN. This may affect the charging rate.

Clearing facilities

A copy of the facilities field in the clear packet (refer to your network documentation). This field is displayed in hexadecimal notation. For some networks, it may contain all the charging information you require.

Network device

The NW or NV unit reference.

LCN

The Logical Channel Number associated with the circuit.

Diagnostic

Diagnostic information from the clear packet sent by the PSDN (see your network documentation).

Calling facilities

A copy of the facilities field in the calling packet (refer to your network documentation). The field is displayed in hexadecimal notation.

Accept facilities

A copy of the facilities field in the call accept packet (refer to your network documentation). The field is displayed in hexadecimal notation.

9.4.3 ACCOUNTING/PSI and X.29 Incoming Calls

For X.29 incoming access, the ACCOUNTING/PSI Utility gives SYSTEM as the username because it is the LES$ACP process that accepts the call. This process runs under the account name of the process that started PSI. This is usually SYSTEM.

You can correlate the PSI and VMS accounting records to find out the username of the X.29 incoming call. Note, however, that if virtual terminals have been enabled, it is not possible to correlate the PSI and VMS accounting records.

Enter the following command to display the name of the NV terminal used in the call:

```
$ ACCOUNTING/PSI/FULL/USER=SYSTEM
```

You can then match this information with the entry in VMS accounting. For example, if the above command displayed NVA6: as the terminal used in the call, enter the following command to find the username corresponding to the X.29 access:

```
$ ACCOUNTING/TERM=NVA6:
```

It is possible, however, for the X.29 terminal to connect to a different process after login by using the SET TERMINAL/NOHANGUP facility. This means that a virtual circuit could be associated with more than one username before being cleared.

9.5 File Space

When output from the ACCOUNTING/PSI Utility is directed to a file, it is possible to consume large quantities of disk space in a short period of time. If your disk quota is exceeded during execution of a ACCOUNTING/PSI request, open files are closed and the request is terminated prematurely. To avoid this situation, carefully plan recording requests by estimating the amount of disk space required, according to your usage.

9.5.1 Factors Influencing Processing Time

The size of the file being processed and the type of processing being done (for example, sorting) can require significant processing time, which may be particularly noticeable on heavily-loaded systems. If this becomes a problem, try running accounting jobs in batch mode.

9.6 Error Messages

The *VMS System Messages and Recovery Procedures Reference Volume* lists the messages generated by the VMS Accounting Utility ACCOUNTING and provides explanations and suggested user actions. The error messages associated with a particular qualifier apply when ACCOUNTING/PSI uses the same qualifier.

9.7 Command Summary and Examples

This section contains a summary of the ACCOUNTING/PSI command format and the command qualifiers. A number of examples showing how some of the qualifiers can be used are also included in this section. A complete description of all the command qualifiers can be found in Chapter 10.

9.7.1 Format

$ ACCOUNTING/PSI [*file-spec*[,...]]

Selection Qualifiers:

Qualifiers	Defaults
/[NO]ACCOUNT= { ["-",]*account-name* / ["-",]*account-name,...* }	/NOACCOUNT
/[NO]ADDRESS= { ["-",]*node-address* / ["-",]*node-address,...* }	/NOADDRESS
/BEFORE [=*time*]	See text
/[NO]IDENT= { ["-",]*process-id* / ["-",]*process-id,...* }	/NOIDENT
/[NO]NODE= { ["-",]*node-name* / ["-",]*node-name,...* }	/NONODE
/[NO]OWNER= { ["-",]*owner-process-id* / ["-",]*owner-process-id,...* }	/NOOWNER
/[NO]PRIORITY= { ["-",]*priority* / ["-",]*priority,...* }	/NOPRIORTY
/[NO]REMOTE_ID= { ["-",]*remote-id* / ["-",]*remote-id,...* }	/NOREMOTE_ID
/[NO]REPORT { [=*report-item*] / [=*report-item,...*] }	/NOREPORT
/[NO]SINCE [=*time*]	/NOSINCE
/[NO]TERMINAL= { ["-",]*terminal-name* / ["-",]*terminal-name,...* }	/NOTERMINAL
/[NO]TYPE= { ["-",]*record-type* / ["-",]*record-type,...* }	/NOTYPE
/[NO]UIC= { ["-",]*uic* / ["-",]*uic,...* }	/NOUIC
/[NO]USER= { ["-",]*user-name* / ["-",]*user-name,...* }	/NOUSER

Other qualifiers:

Qualifiers	Defaults
/[NO]BINARY	/NOBINARY
/[NO]BRIEF	/BRIEF
/[NO]FULL	/NOFULL
/[NO]LOG	/NOLOG
/[NO]OUTPUT [=*file-spec*]	/OUTPUT=SYS$OUTPUT
/[NO]REJECTED [=*file-spec*]	/NOREJECTED
/[NO]SORT { [="-",]*sort-item* } { [="-",]*sort-item,...* }	/NOSORT
/[NO]SUMMARY= { *summary-item* } { *summary-item,...* }	/NOSUMMARY
/[NO]TITLE=*title*	/NOTITLE

The following VMS ACCOUNTING command qualifiers are also allowed with the ACCOUNTING/PSI command. However, since they are selection qualifiers and PSI records do not use these fields, there is no effect if they are entered.

Qualifiers	Defaults
/[NO]ENTRY= { ["-",]*queue-entry* } { ["-",]*queue-entry,...* }	/NOENTRY
/[NO]IMAGE= { ["-",]*image-name* } { ["-",]*image-name,...* }	/NOIMAGE
/[NO]PROCESS= { ["-",]*process-type* } { ["-",]*process-type,...* }	/NOPROCESS
/[NO]QUEUE= { ["-",]*queue-name* } { ["-",]*queue-name,...* }	/NOQUEUE
/[NO]STATUS= { ["-",]*exit-status* } { ["-",]*exit-status,...* }	/NOSTATUS

Command Parameter

file-spec[,...]

Specifies one or more accounting files as input to be processed by ACCOUNTING/PSI. If you specify more than one file name, separate them with commas. If you omit the file-spec parameter, data is processed from the default accounting log, SYS$MANAGER :PSIACCOUNTING.DAT.

Wildcard characters are allowed in the file specification.

9.7.2 Restrictions

Some of the above qualifiers are subject to restrictions. These are as follows;

- The /SORT qualifier may only be used for brief or full listings, not for summary reports.
- The /BRIEF, /FULL, /SUMMARY and /BINARY qualifiers are mutually exclusive.
- The /REPORT qualifier requires the /SUMMARY qualifier.
- If a sort item specifies a field that is not present in the record, that record becomes unselected and will be reflected as such in the selected and rejected record counts.

9.7.3 Examples

9.7.3.1 Listing Accounting Files

The accompanying examples illustrate the listing mode of operation. Use this mode when you want to examine the activity of the system, either on a routine basis, or as part of an installation checkout, tuning, or trouble-shooting exercise. No historical record of output is kept.

```
$ ACCOUNTING/PSI
```

This command produces a display of all accounting records in the default accounting file. Since no command qualifiers have been named, ACCOUNTING/PSI applies the following defaults to the command:

- /NOFULL = a brief listing
- *input file* = SYS$MANAGER:PSIACCOUNTING.DAT
- /OUTPUT = current SYS$OUTPUT device

By default, listing begins when the command is issued and ends when you reach the end of the file.

```
$ ACCOUNTING/PSI MYFILE
```

This command also provides a brief listing, but lists from a specified binary input file.

```
$ ACCOUNTING/PSI/FULL
```

The command in this example provides a full listing of all the records in the default accounting file.

9.7.3.2 Selecting Records

You identify groups of accounting records with one or more of the following selection qualifiers:

- /ACCOUNT
- /ADDRESS
- /BEFORE
- /IDENT
- /NODE
- /OWNER
- /PRIORITY
- /REMOTE_ID
- /SINCE
- /TERMINAL
- /TYPE
- /UIC
- /USER

If you omit these qualifiers, the defaults provide for selecting all records. The next example illustrates selection.

```
$ ACCOUNTING/PSI/SINCE=15-APR-1987 -
_ /BEFORE=22-JUN-1987:23:59:99/ACCOUNT=MANUFA/NODE=OSCAR
```

This command selects and lists in brief format only those records for the node OSCAR and its manufacturing account on the specified days.

9.7.3.3 Sorting Records

The ACCOUNTING/PSI Utility includes a sorting facility. You use the /SORT qualifier to specify the fields that you want the sort to occur on and whether the desired sequence is ascending or descending. For example:

```
$ ACCOUNTING/PSI/SORT=(ACCOUNT,USER,-PRIORITY)
```

This command sorts all records in the current accounting file by:

- Account
- User name in ascending order
- Final priority in descending order to provide a brief listing.

You can also sort just those records that you select, by combining one or more selection qualifiers with the /SORT qualifier. However, you only use the /SORT qualifier for brief or full listings, not for summary reports. For example:

```
$ ACCOUNTING/PSI/ACCOUNT=MAIL/SORT=USER
```

This command selects all records for the account MAIL and then sorts them by user name, producing a brief listing.

9.7.3.4 Directing the ACCOUNTING/PSI Output

ACCOUNTING/PSI output can be routed to any supported terminal device or to a disk or tape file. The following command sends its ASCII listing of all records in the accounting file to the file ACCOPY.LIS. This file could then be printed out on a hard-copy device.

```
$ ACCOUNTING/PSI /OUTPUT=ACCOPY
```

You can also direct binary output to a file whenever you need to capture ACCOUNTING/PSI data for future use. For example, you might want to plan for routine performance data gathering for long-term analysis. ACCOUNTING/PSI data can be recorded on a routine basis and summarized to gather data about PSI resource utilization over long periods of time.

9.7.3.5 Using DCL Symbols

It may be convenient to establish DCL symbols for frequently used combinations, as in the accompanying example.

```
$ MY_GROUP == "/USER=(MARY,TOM,DICK,HARRY,BARNEY,ALICE)"
$ ACCOUNTING/PSI 'MY_GROUP'
```

This example shows how you can define the DCL symbol MY_GROUP and use it as a parameter to the ACCOUNTING/PSI command.

10

The ACCOUNTING/PSI Command Qualifiers

This Chapter describes all the ACCOUNTING/PSI command qualifiers and provides examples of their use. The qualifiers are described in alphabetical order and the following conventions are observed:

[ ]	Square brackets are used to enclose optional parameters. All parameters without these brackets are mandatory.
...	Dots mean that more than one value of the associated parameter can be entered.

The DCL Rules for entering commands, described in the *VMS DCL Dictionary*, apply to ACCOUNTING/PSI qualifiers.

/ACCOUNT

Controls whether only those records matching the specified account name(s) are selected. The account name(s) specified must match the account name as specified in the user authorization file.

Format

/ACCOUNT= { (["-",]*account-name*) / (["-",]*account-name*,...) }
/NOACCOUNT

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified account name(s).

account-name
account-name,...

Specifies the account name(s) used to select records.

When you specify the /ACCOUNT qualifier, you must specify at least one account name. If you specify more than one account name, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any account-name in the list.

If you omit the qualifier or specify /NOACCOUNT, the account-name is not used to select records.

Examples

```
$ ACCOUNTING/PSI /ACCOUNT=(ROBERT,RUTH)
```

The command in this example selects records matching the accounts ROBERT and RUTH.

```
$ ACCOUNTING/PSI /ACCOUNT=("-",ROBERT)
```

The command in this example selects records for all accounts except ROBERT.

/BEFORE

Controls whether only those records dated earlier than the specified time are selected. (Where the record date is when the access to VAX PSI was finished.)

Format

/BEFORE [=*time*]

Qualifier Keyword

time

Specifies the time used to select records. Records dated earlier than the specified time are selected. You can specify an absolute time, delta time, or a combination of the two. Observe the syntax rules for date and time described in the *VMS DCL Dictionary*.

Description

If you specify /BEFORE without the time, midnight of the current date is used. If you omit the qualifier, the current date and time is used by default.

Example

```
$ ACCOUNTING/PSI /BEFORE=5-JAN-1988
```

The command in this example selects all records dated earlier than January 5, 1988.

/BINARY

Controls whether output is formatted in either binary or ASCII.

Format

/BINARY
/NOBINARY

Qualifier Keywords

None.

Description

When /BINARY is specified, the output file, specified using the /OUTPUT qualifier, contains identical copies of the input records. If you specify /NOBINARY or omit the qualifier, the output file contains formatted ASCII records.

When used with selection and sorting, the output file contains binary copies of the selected (or rejected) entries only. These are sorted as specified.

Example

```
$ ACCOUNTING/PSI /OUTPUT=MYACC.DAT/BINARY/USER=COBB
```

The command in this example writes accounting data for the user COBB in binary format to the file MYACC.DAT.

/BRIEF

Controls whether a brief format is used in ASCII displays.

Format

/BRIEF

Qualifier Keywords

None.

Description

By default, records are displayed in the brief format. You must specify /FULL to have the full contents of each selected record displayed.

The /BRIEF, /FULL, /SUMMARY, and /BINARY qualifiers are mutually exclusive qualifiers.

Example

```
$ ACCOUNTING/PSI /BRIEF
```

The command in this example displays the brief contents of each selected record.

/FULL

Controls whether a full format is used in ASCII displays.

Format

/FULL
/NOFULL

Qualifier Keywords

None.

Description

By default, records are displayed in the brief format. You must specify /FULL to have the full contents of each selected record displayed.

The /BRIEF, /FULL, /SUMMARY, and /BINARY qualifiers are mutually exclusive qualifiers.

If you specify /NOFULL or omit the qualifier, records are displayed in the brief format.

Example

```
$ ACCOUNTING/PSI /FULL
```

The command in this example displays the full contents of each selected record.

/IDENT

Controls whether only those records matching the specified process identifier number(s) are selected.

Format

/IDENT= { (["-",]*process-id*) / (["-",]*process-id,...*) }
/NOIDENT

Qualifier Keywords

"-"

Specifies that all records are selected except those matching the specified process-id(s).

process-id
process-id,...

Specifies the process-id(s) used to select records. When you specify /IDENT, you must specify at least one process-id. If you specify more than one process-id, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any process-id in the list.

If you specify /NOIDENT or omit the qualifier, the process-id is not used to select records.

Examples

```
$ ACCOUNTING/PSI /IDENT=(25634,045A6B)
```

The command in this example selects records matching the process-id's specified.

```
$ ACCOUNTING/PSI /IDENT=("-",2B758)
```

The command in this example selects all records except those matching the specified process-id.

/LOG

Controls whether informational messages (input file names, selected record counts, rejected record counts) are displayed to SYS$OUTPUT.

Format

/LOG
/NOLOG

Qualifier Keywords

None.

Description

By default, messages are not displayed. If more than one input file is specified in a ACCOUNTING/PSI command with the /LOG qualifier, there is one logging message for each file and a total of selected/rejected record counts is provided.

Example

```
$ ACCOUNTING/PSI /LOG
```

The command in this example displays accounting records along with any informational messages such as selected record counts and rejected record counts.

/NODE

Controls whether only those records matching the specified remote node name(s) are selected. The node name is a unique identifier for DECnet nodes and is only relevant for circuits that use VAX PSI Access.

That is, in a Multi-host node for calls to and from a VAX PSI Access node, and in a VAX PSI Access node, for calls made through the (named) VAX PSI Multi-host node. For other circuits the name field is blank.

Format

/NODE= { ["-",]*node-name* }
{ ["-",]*node-name,...* }
/NONODE

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified node name(s).

node-name
node-name,...

Specifies the node name(s) used to select records. When you specify the /NODE qualifier, you must specify at least one node name. If you specify more than one node name, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any node name in the list.

If you omit the qualifier or specify /NONODE, the node name is not used to select records.

Example

```
$ ACCOUNTING/PSI /NODE=STAR
```

The command in this example selects accounting records for the node STAR.

/OUTPUT

Specifies the output file to which the output is sent.

Format

/OUTPUT [=*file-spec*]
/NOOUTPUT

Qualifier Keyword

file-spec

Specifies the name of the file that is to contain the output.

If you omit the device or directory specification, the current device and default directory are used. If you omit the file name, then the file name of the input file is used. If you omit the file type and the output is ASCII (/NOBINARY or default), the default file type is LIS. If you omit the file type and the output is binary (/BINARY), the default file type is DAT.

Description

If you omit the qualifier, output is directed to the current SYS$OUTPUT device.

Examples

```
$ ACCOUNTING/PSI /OUTPUT=STAT.DAT /BINARY
```

The command in this example copies all accounting records from the currently open accounting file and outputs them to the file STAT.DAT. This allows the accounting file to be copied without closing it.

```
$ ACCOUNTING/PSI /OUTPUT=STAT
```

The command in this example generates a brief ASCII listing of all accounting records on the file STAT.LIS. Notice that the default file type for ASCII output is .LIS.

/OWNER

Controls whether only those records matching the specified owner process identifier number(s) are selected.

Format

/OWNER= { (["-",]*owner-process-id*) / (["-",]*owner-process-id,...*) }
/NOOWNER

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified owner process-id(s).

owner-process-id
owner-process-id,...

Specifies the owner process identification number(s) used to select records. Owner process-ids are only present in subprocesses to specify the process-id(s) of their owner process.

When you specify /OWNER, you must specify at least one owner process-id. If you specify more than one owner process-id, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any owner process-id in the list.

If you specify /NOOWNER or omit the qualifier, the owner process-id is not used to select records.

Example

```
$ ACCOUNTING/PSI /OWNER=0000002C
```

The command in this example selects records that match the specified owner process-id of 0000002C.

/PRIORITY

Controls whether only those records matching the specified base process priority are selected.

Format

/PRIORITY= { (["-",]*priority*) / (["-",]*priority*,...) }
/NOPRIORITY

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified base process priority.

priority
priority,...

Specifies the process base priority used to select records.

When you specify /PRIORITY, you must specify at least one priority. If you specify more than one priority, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any priority in the list.

If you specify /NOPRIORITY or omit the qualifier, the priority is not used to select records.

Examples

```
$ ACCOUNTING/PSI/PRIORITY=3
```

The command in this example selects records that match a base priority of 3.

```
$ ACCOUNTING/PSI/PRIORITY=("-",3)
```

The command in this example selects all records except those that match a base priority of 3.

/REJECTED

Controls whether unselected records are output to a specified file. These records are always in binary format.

Format

/REJECTED [=*file-spec*]
/NOREJECTED

Qualifier Keyword

file-spec

Specifies the name of the file to contain unselected records. If you omit the device or directory specification, the current device and default directory are used. If you omit the file name, then the file name of the input file is used. If you omit the file type, REJ is used.

Description

If you specify /NOREJECTED or omit the qualifier, unselected records are not output.

Example

```
$ ACCOUNTING/PSI/PRIORITY=3/REJECTED=BAD
```

The command in this example outputs all unselected records, that is records that do not have a base priority of 3, to the file BAD.REJ. Notice that the default file type is .REJ.

/REMOTE_ID

Controls whether only those records matching the specified remote-id(s) are selected. The remote-id is a system dependent identifier for DECnet users and is only relevant for circuits that use VAX PSI Access. For other circuits the remote-id field is null.

Format

/REMOTE_ID= { (["-",]*remote-id*) / (["-",]*remote-id,...*) }
/NOREMOTE

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified remote-id(s).

remote-id
remote-id,...

Specifies the remote-id(s) used to select records.

When you specify the /REMOTE_ID qualifier, you must specify at least one remote-id. If you specify more than one, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any remote-id in the list.

If you omit the qualifier or specify /NOREMOTE_ID, the remote-id is not used to select records.

/REPORT

Controls whether a specified item is included in a summary report. One column is generated on the summarization report for each item specified. The /REPORT qualifier requires the /SUMMARY qualifier. See the description of the /SUMMARY qualifier.

Format

/REPORT { ([=*report-item*]) / ([=*report-item*,...]) }
/NOREPORT

Qualifier Keyword

report-item
report-item,...

Specifies the report item(s) used to select records.

You can specify the following items:

Keyword	Meaning
ELAPSED	Total elapsed time
RECORDS	Total records in file (default)

If you specify more than one report item, separate them with commas and enclose the list in paratheses.

The columns on the summarization report appear in the same left-to-right sequence as you give in the list of report items.

Description

If you specify /REPORT without a value (or if you specify /SUMMARY and do not specify /REPORT) then /REPORT=RECORDS is assumed.

Example

```
$ ACCOUNTING/PSI/SUMMARY/REPORT=ELAPSED
```

The command in this example produces a summary report of call elapsed time.

/SINCE

Controls whether only those records dated later than a specified time are selected. (The date on the records is when the access to VAX PSI was finished.)

Format

/SINCE [=*time*]
/NOSINCE

Qualifier Keyword

time

Specifies the time used to select records. Records dated later than the specified time are selected. You can specify an absolute time, delta time, or a combination of the two. Observe the syntax rules for date and time as described in the *VMS DCL Dictionary*.

Description

If you specify /SINCE without time, midnight of the current day is used.

If you specify /NOSINCE or omit the qualifier, no time is used to select records.

Example

```
$ ACCOUNTING/PSI /SINCE=4-DEC-1987
```

The command in this example selects records dated on or later than December 4, 1987.

/SORT

Specifies the sequence of the records in the brief or full listing. THE /SORT qualifier may be used with the /BINARY, /BRIEF, and /FULL qualifiers, but not with /SUMMARY.

Format

/SORT { [="-",]*sort-item* } { [="-",]*sort-item,...* }
/NOSORT

Qualifier Keywords

"-"

Specifies that the sort field is used as a descending key. By default the sequence is ascending.

sort-item
sort-item,...

Specifies the sort item(s) used to select records.

At least one sort item must be specified. If you specify more than one sort item, separate them with commas and enclose the list in parentheses.

You can specify any of the following sort items:

Keyword	Meaning
ACCOUNT	User's account name
ADDRESS	Remote node address
ELAPSED	Elapsed time
FINISHED	Termination time
IDENT	Process identification
NODE	Remote node name
OWNER	Owner process identification
PRIORITY	Process base priority
STARTED	Time call was started
TERMINAL	Terminal name
TYPE	Record type
UIC	User identification code
USER	User's name

Description

If a sort item is preceded by a minus sign (-), then that field is used as a descending key. For example if a sort of user-ids was requested and (-) specified then the highest id number would be first and the lowest at the end of the list. By default keys are assumed to be ascending.

The selected records are sorted according to the sequence specified by the sort items given with the /SORT qualifier prior to writing them to the designated output file. The ordering of sort items in the qualifier value list determines the relative ranking of the keys.

NOTE

If a sort item specifies a field that is not present in a record, that record becomes unselected and will be reflected as such in the counts of selected and rejected records.

Example

```
$ ACCOUNTING/PSI/SORT=(USER,PRIORITY,STARTED)
```

The command in this example sorts the selected records in the sequence specified by the /SORT qualifier, rejecting any records in which either of the three fields are blank.

/SUMMARY

Specifies that a summary of the selected records, grouped by the list of summary keys, be produced. Use the /REPORT qualifier to control what information is summarized. If you omit the /REPORT qualifier, /REPORT=RECORDS is assumed. The /SUMMARY qualifier is required with the /REPORT qualifier.

Format

/SUMMARY { [=*summary-item*] / [=*summary-item*,...] }
/NOSUMMARY

Qualifier Keyword

summary-item
summary-item,...

Specifies the summary item(s) used to select records.

You can specify any of the following summary items:

Keyword	Outputs
ACCOUNT	Account name from the UAF
DATE	YYYY MM DD
DAY	Day of month (1-31)
HOUR	Hour of day (0-23)
MONTH	Month of year (1-12)
NODE	Remote node name
TERMINAL	Terminal name
TYPE	Type of record (logout, batch)
UIC	User identification code
USER	User name from UAF
WEEKDAY	Day of week (0=Sunday, 1=Monday, and so on)
YEAR	Year

If you specify /SUMMARY without a value, then /SUMMARY=USER is assumed.

If you specify more than one summary item, separate them with commas and enclose the list in parentheses.

Description

The summarized items are sorted in ascending order and listed in the same left-to-right sequence that you give in the list of summary items. The output is sent to the SYS$OUTPUT device unless specifically directed elsewhere by the /OUTPUT qualifier.

If you specify /NOSUMMARY or omit the qualifier, no summarization occurs.

Example

```
$ ACCOUNTING/PSI/SUMMARY=ACCOUNT
```

The command in this example generates a summary report by Account name

/TERMINAL

Controls whether only those records matching the specified terminal name(s) are selected. Terminal names are associated with interactive processes.

Format

/TERMINAL= { ["-",]*terminal-name* / ["-",]*terminal-name,...* }
/NOTERMINAL

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified terminal name.

terminal-name
terminal-name,...

Specifies the terminal name(s) used to select records.

When you specify /TERMINAL, you must specify at least one terminal name. Specify terminal names as a standard device names and include the colon (:), for example, TTA6:.

If you specify more than one terminal name, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any terminal-name in the list.

If you specify /NOTERMINAL or omit the qualifier, the terminal name is not used to select records.

Examples

```
$ ACCOUNTING/PSI/TERMINAL=TTB3:
```

The command in this example selects records that match the terminal name TTB3:.

```
$ ACCOUNTING/PSI/TERMINAL=("-",TTB3:)
```

The command in this example selects all records except those that match the terminal name TTB3:.

/TITLE

Specifies the title to be printed in the center of the first line of summary reports. The title line also includes the beginning and ending times for the data summary at the left and right margins, respectively.

Format

/TITLE=*title*
/NOTITLE

Qualifier Keyword

title

Specifies the title to be printed on the summary report. If the title includes spaces or special characters, or if you want to preserve lower case letters, you must enclose it in quotation marks (" ").

Example

```
$ ACCOUNTING/PSI/SUMMARY=ACCOUNT/TITLE="JUNE ACCOUNTING REPORT"
```

The command in this example selects a summary report and writes the title "JUNE ACCOUNTING REPORT" at the top of the report.

/TYPE

Controls whether only those records matching the specified record type are selected.

Format

/TYPE= { ["-",]*record-type* / ["-",]*record-type*,... }
/NOTYPE

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified record type.

record-type
record-type,...

Specifies the record type used to select records. You can specify the following record types:

Keyword	Meaning
FILE	Accounting file forward and backward pointers
PSI	PSI Virtual circuit termination

When you specify /TYPE, you must specify at least one record type. If you specify more than one record type, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any record type in the list.

If you specify /NOTYPE or omit the qualifier, the record type is not used to select records.

Examples

```
$ ACCOUNTING/PSI/TYPE=PSI
```

The command in this example selects records that match the record type PSI.

```
$ ACCOUNTING/PSI/TYPE=("-",PSI)
```

The command in this example selects all records except those that match the record type PSI (in effect the file forward and backward pointers).

/UIC

identification code (UIC) are selected.

Format

/UIC= { ["-",]*uic* / ["-",]*uic*,... }
/NOUIC

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified UIC.

uic
uic,...

Specifies the user identification code (UIC) used to select records.

When you specify /UIC, you must specify at least one UIC. If you specify more than one UIC, separate them with commas and enclose the list in parentheses.

Specify the UIC in the format:

[g,m]

where:

g	is an octal number in the range 0 through 377 representing the group number
m	is an octal number in the range 0 through 377 representing the member number

Square brackets ([ ]) or angle brackets (<>) are required in the UIC specification.

You can specify the asterisk (*) wild card character in either the group or member fields of the UIC specification, or in both.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any UIC in the list.

If you specify /NOUIC or omit the qualifier, the UIC not used to select records.

Example

```
$ ACCOUNTING/PSI/UIC=[360,*]
```

The command in this example selects records that match UICs having a group number of 360.

/USER

Controls whether only those records matching the specified user name are selected. The user name matches the user name in the user authorization file.

Format

/USER= { ["-",]*user-name* / ["-",]*user-name*,... }

Qualifier Keywords

"-"

Specifies that all records are selected except those matching any specified user name.

user-name
user-name,...

Specifies the user name used to select records.

When you specify /USER, you must specify at least one user name. If you specify more than one user name, separate them with commas and enclose the list in parentheses.

Description

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected except those matching any user name in the list.

If you specify /NOUSER or omit the qualifier, the user name is not used to select records.

Examples

```
$ ACCOUNTING/PSI/USER=SASHA
```

The command in this example selects records that match the user name SASHA.

```
$ ACCOUNTING/PSI/USER=("-",SASHA)
```

The command in this example selects all records except those that match the user-name SASHA.

11

PSI Accounting Record Formats

This chapter describes the structure of the data records written to the PSI Accounting log file SYS$MANAGER:PSIACCOUNTING.DAT.

These records are generated by the completion of any VAX PSI virtual circuit access. This includes the following:

- Incoming or outgoing X.25 calls
- Incoming or outgoing X.29 calls
- PVC accesses
- DLM accesses

The accounting record types, the offsets within the accounting records, and the other symbols used in these formats are all defined by the symbolic definition macros $ACRDEF and $PSIDEF.

NOTE

The formats described in this section are valid for VAX PSI V4.2 but are subject to change without further notice at the time of a future VAX PSI release.

11.1 Record Format

A VAX PSI accounting record follows the format of the standard VMS V4.2 accounting records (Refer to the VMS System Management documentation for a description) with the addition of a new record type and a new packet.

An accounting record consists of a record header and a number of information packets. The number of information packets depends on the type of information being recorded.

Figure 11–1 illustrates the general format of the accounting record, and Table 11–1 describes the fields contained in this record. The type field in the accounting record header (described in Table 11–1) is subdivided into five fields. Table 11–2 describes these fields.

Figure 11–1 Accounting Record Format

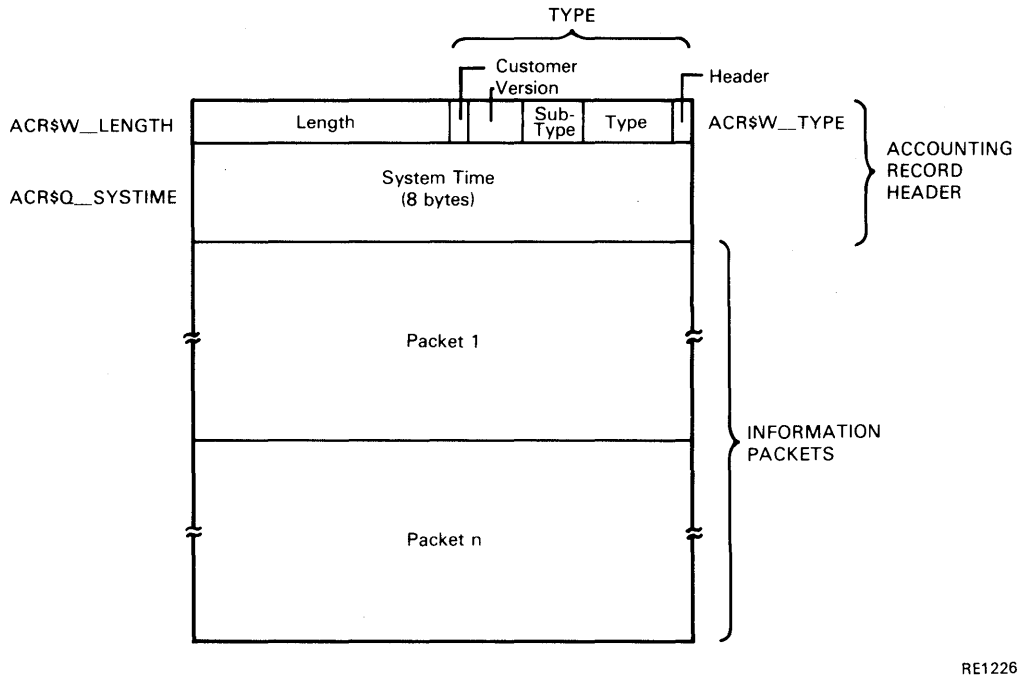


Table 11–1 Descriptions of Accounting Record Fields

Field	Symbolic Offset	Contents
type	ACR$W_TYPE	Information describing the record. This field is subdivided into five fields, as described in Table 11–2 (1 word)
length	ACR$W_LENGTH	Total length of the record. (1 word)
system time	ACR$Q_SYSTIME	Current system time. (1 quad word)
packet 1 - n		Information packets associated with the record. (variable length)

Table 11–2 Descriptions of ACR$W_TYPE FIELDS

Field	Symbolic Offset	Contents
header	ACR$V_PACKET	Identifies this header as a record header. This bit is set to 0. (1 bit)
type	ACR$V_TYPE	Indicates the purpose of the record. Section 11.2 describes the three types of record in the PSI accounting file. (7 bits)
subtype	ACR$V_SUBTYPE	Indicates the type of PSI record. The subtype in this version of PSI is ACR$K_PSI_VCT (Virtual Circuit Termination). (4 bits)
version	ACR$V_VERSION	Indicates the accounting format with which the record is associated. The format for PSI records is ACR$K_VERSION3. (3 bits)
customer	ACR$V_CUSTOMER	Identifies whether the record was written by DIGITAL software or by customer software. If the bit is not set, the record was written by DIGITAL software. If set to 1, the record was written by customer software. (1 bit)

11.2 Accounting Record Types

Accounting record types identify the type of operation that caused the record to be sent. There are three accounting record types which may occur in the VAX PSI accounting file.

Each type of accounting record requires a defined set of packets. Table Table 11–3 describes the accounting record types and lists the packets required by each type.

Table 11–3 Accounting Record Types

Symbol	Meaning	Requires
ACR$K_FILE_FL	Accounting file forward link.	ACR$K_FILENAME
ACR$K_FILE_BL	Accounting file backward link.	ACR$K_FILENAME
ACR$K_PSI	VAX PSI data.	ACR$K_ID

11.3 Accounting Packets

There are three types of accounting packets which may occur in the VAX PSI accounting file:

- Identification Packet
- File Name Packet
- PSI Virtual Circuit Termination Packet

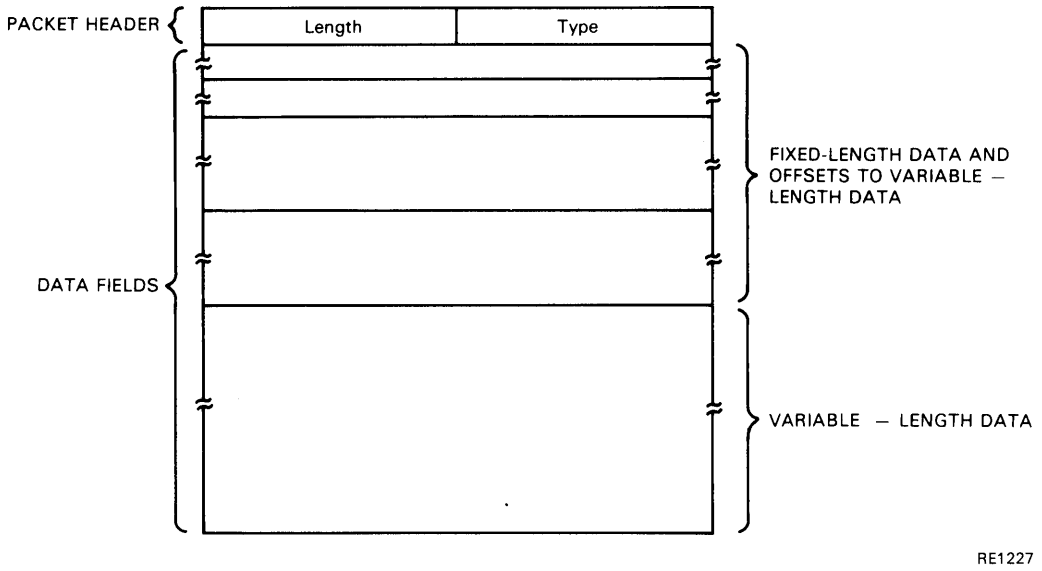
Section 11.3.1 describes the general format of the accounting packets. Sections 11.3.2 through 11.3.4 describe the organization of the different accounting packets.

11.3.1 General Format of Accounting Packets

Each packet type contains a packet header, followed by data fields. The data fields can contain fixed-length data, variable-length data, or offsets to variable-length data. Offsets contain the distance, in bytes, from the beginning of the packet to the variable-length data.

All variable-length data is represented as counted strings. Variable-length data follow the last fixed-length data field in the packet. Figure 11–2 illustrates the general format of the accounting packet.

Figure 11–2 Accounting Packet Format



Accounting packets need not use each type of data field. See Sections 11.3.2 through 11.3.4 for complete descriptions of the fields contained in each accounting packet.

All accounting packets start with a packet header. The packet header uses the same symbolic offsets as the first longword of the record header. Figure 11–3 illustrates the accounting packet header; Table 11–4 describes the fields in this header. The type field in the accounting packet header is subdivided into five fields. Table 11–5 describes these fields.

Figure 11–3 Accounting Packet Header Format

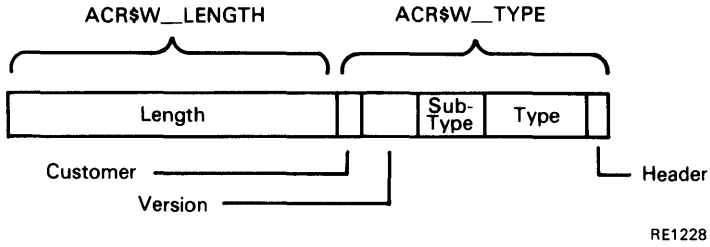


Table 11–4 Descriptions of Accounting Packet Header Fields

Field	Symbolic Offset	Contents
type	ACR$W_TYPE	Information describing the packet (1 word). This field is subdivided into five fields, as described in Table Table 11–5.
length	ACR$W_LENGTH	Total length of the packet. (1 word)

Table 11–5 Descriptions of ACR$W_TYPE Fields

Field	Symbolic Offset	Contents
packet	ACR$V_PACKET	Identifies this header as a packet header. This bit is set to 1. (1 bit)
type	ACR$V_TYPE	Indicates the purpose of the packet. There are currently three packet types. These packet types are described in Sections 11.3.2 through 11.3.4 (7 bits).
subtype	ACR$V_SUBTYPE	Indicates the packet subtype. (4 bits)
version	ACR$V_VERSION	Indicates the accounting format with which the record is associated. The format for PSI records is ACR$K_VERSION3. (3 bits)
customer	ACR$V_CUSTOMER	Identifies whether the record was written by DIGITAL software or by customer software. If the bit is not set, the record was written by DIGITAL software. If set to 1, the record was written by customer software. (1 bit)

11.3.2 Packet Type ACR$K_ID (Identification Packet)

The identification packet identifies the process that caused information to be sent to the accounting manager.

Figure 11–4 depicts the organization of the identification packet; Table 11–6 describes the fields contained in the packet. See Section 11.1 for more information on the packet header.

Figure 11–4 Block Diagram for ACR$KID

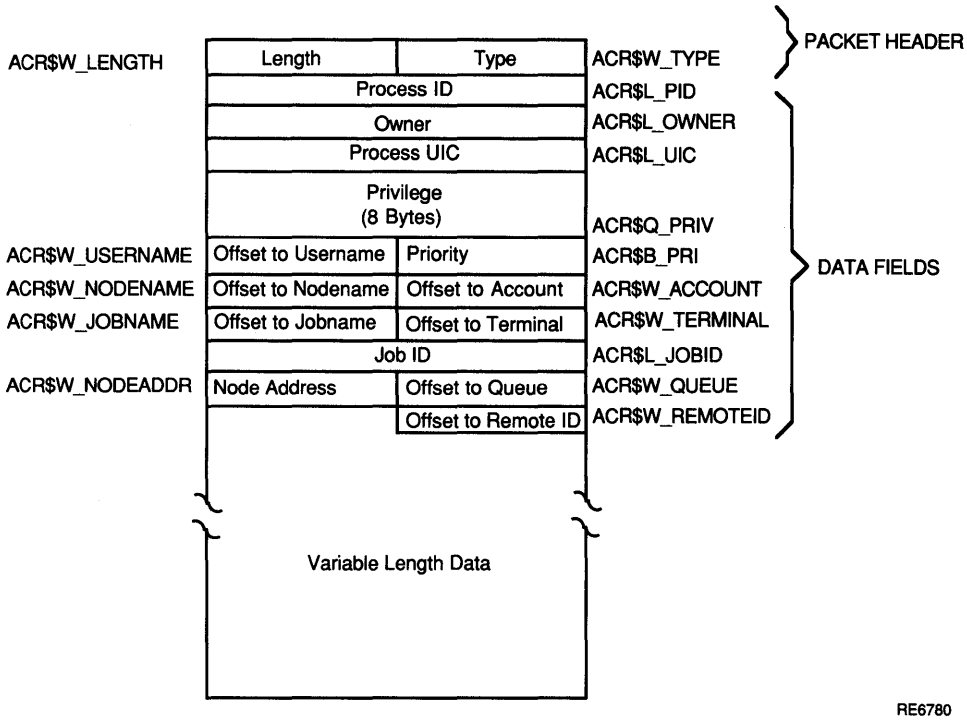


Table 11–6 Field Descriptions for ACR$K_ID

Field	Symbolic Offset	Contents
pid	ACR$L_PID	Process identification of the process. (longword)
owner	ACR$L_OWNER	Process identification of owner process, if the process is a subprocess. If the process is not a subprocess, the value is 0. (longword)
uic	ACR$L_UIC	Process UIC of the process. The UIC can be addressed as two separate words: ACR$W_MEM for the member number, and ACR$W_GRP for the group number. (longword)
privilege	ACR$Q_PRIV	Privileges held by the process at the time of call termination. (quadword)
username	ACR$W_USERNAME	Offset to counted ASCII string containing the user name of the process. (word)
prio	ACR$B_PRI	Priority of the process. (byte)
node name	ACR$W_NODENAME	Offset to counted ASCII string containing the node name of the remote process (word). This is only relevant if the circuit used PSI Access or Multi-host.
account	ACR$W_ACCOUNT	Offset to counted ASCII string containing the account name of the process. (word)
jobname	ACR$W_JOBNAME	Offset to counted ASCII string containing the job name of the process (word). The string will be zero bytes long for PSI records.
terminal	ACR$W_TERMINAL	Offset to counted ASCII string containing the terminal name of the process. (word)
jobid	ACR$L_JOBID	Identification of the job (longword). Zero for PSI records.
node address	ACR$W_NODEADDR	Contains the remote node address (word). This is only relevant if the circuit used PSI Access or Multi-host.
queue	ACR$W_QUEUE	Offset to counted ASCII string containing the name of the queue with which a batch or print job is associated (word). The string will be zero bytes long for PSI records.
remote id	ACR$W_REMOTEID	Offset to counted ASCII

11.3.3 Packet Type ACR$K_FILENAME (File Name Packet)

The file name packet contains the name of the next or previous accounting file to be opened. Figure 11–5 depicts the organization of the file name packet; Table 11–7 describes the fields contained in the packet. See Section 11.1 for more information on the packet header.

Figure 11–5 Block Diagram for ACR$KFILENAME

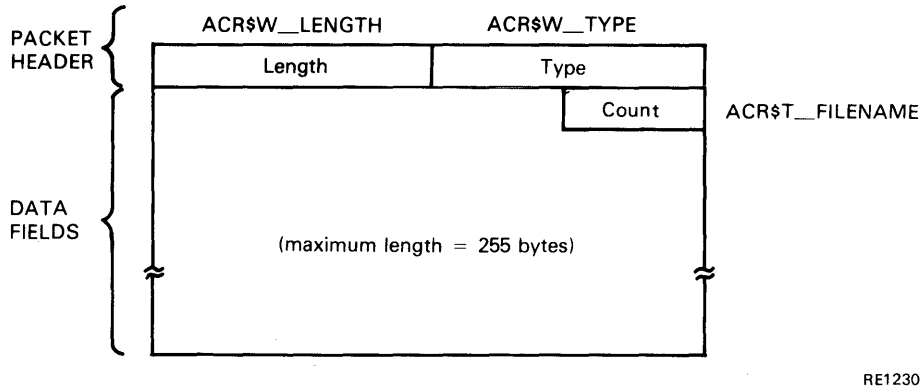


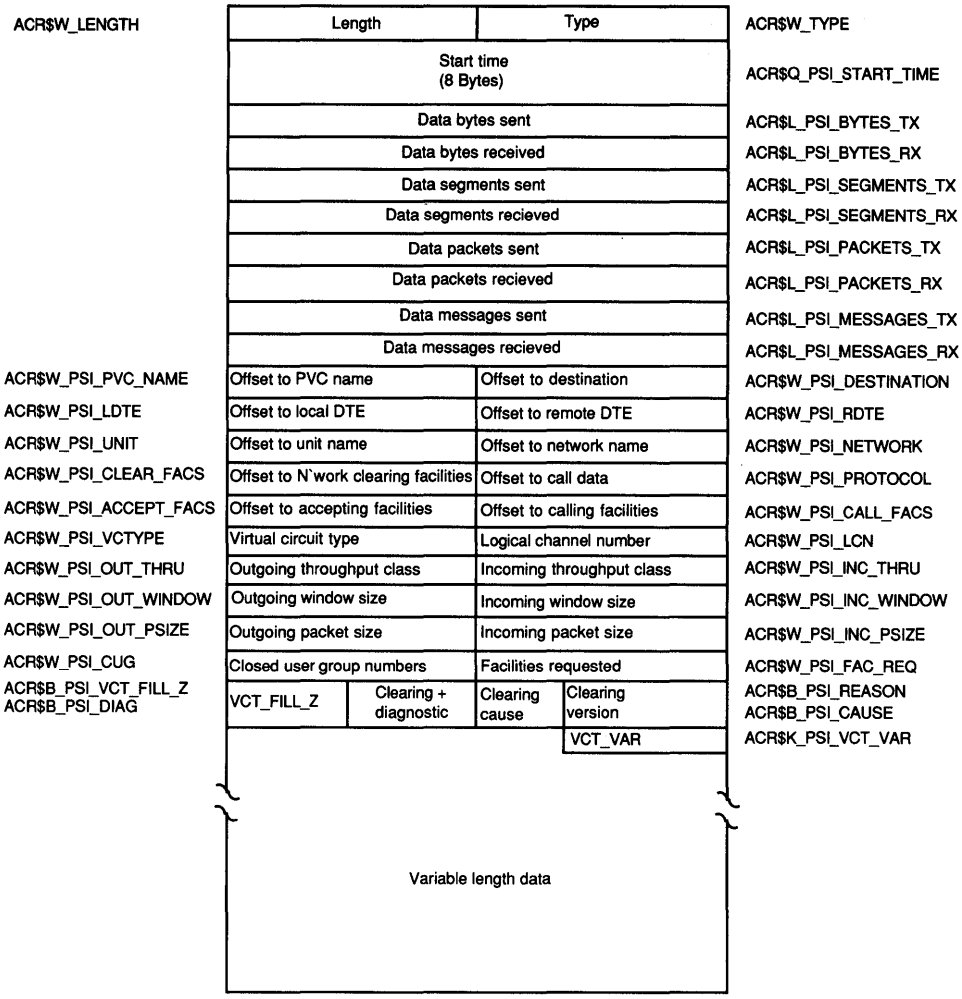
Table 11–7 Field Descriptions for ACR$K_FILENAME

Field	Symbolic Offset	Contents
file name	ACR$T_FILENAME	Name of the file. (counted ASCII string)

11.3.4 Packet Type ACR$K_PSI (PSI Packet)

The VAX PSI packet contains data on the usage of VAX PSI for accounting purposes. Figure 11–6 depicts the organization of the packet; Table 11–8 describes the fields contained in the packet. See Section 11.1 for more information on the packet header.

Figure 11–6 Block Diagram for ACR$KPSI



RE6781

Table 11–8 Field Descriptions for ACR$K_PSI

Field	Symbolic Offset	Contents
start-time	ACR$Q_PSI_START_TIME	System time at the start of the circuit. This will not be used for incoming calls that fail to get sent to any process as, in this case, the circuit never started. (quadword)
bytes tx	ACR$L_PSI_BYTES_TX	Data bytes sent. (longword)
bytes rx	ACR$L_PSI_BYTES_RX	Data bytes received. (longword)
segments tx	ACR$L_PSI_SEGMENTS_TX	Data segments sent. (longword)
segments rx	ACR$L_PSI_SEGMENTS_RX	Data segments received. (longword)
packets tx	ACR$L_PSI_PACKETS_TX	Data packets sent. (longword)
packets rx	ACR$L_PSI_PACKETS_RX	Data packets received. (longword)
messages tx	ACR$L_PSI_MESSAGES_TX	Data messages sent. (longword)
messages rx	ACR$L_PSI_MESSAGES_RX	Data messages received. (longword)
name	ACR$W_PSI_PVC_NAME	Offset to counted string containing the PVC name. (word)
destination	ACR$W_PSI_DESTINATION	Offset to counted string containing the destination name. (word)
ldte	ACR$W_PSI_LDTE	Offset to counted string containing the local DTE address. (word)
rdte	ACR$W_PSI_RDTE	Offset to counted string containing the remote DTE address. (word)
unit	ACR$W_PSI_UNIT	Offset to counted string containing NW or NV unit name. (word)
network	ACR$W_PSI_NETWORK	Offset to counted string containing the network name. (word)
clear facs	ACR$W_PSI_CLEAR_FACS	Offset to counted string containing network clearing facilities. (word)
protocol	ACR$W_PSI_PROTOCOL	Offset to counted string containing the protocol ID field (word). This is the first four bytes of call user data.
accept facs	ACR$W_PSI_ACCEPT_FACS	Offset to counted string containing call accepting facilities. (word)
call facs	ACR$W_PSI_CALL_FACS	Offset to counted string containing calling facilities. (word)

Table 11–8 (Cont.) Field Descriptions for ACR$K_PSI

Field	Symbolic Offset	Contents
vctype	ACR$W_PSI_VCTYPE	Indicates the type of virtual circuit (word). The following flags are defined: 1. ACR$V_PSI_OUT - Outgoing switched virtual circuit. 2. ACR$V_PSI_PVC - Permanent virtual circuit. 3. ACR$V_PSI_X29 - Circuit used for X29 access (only relevant for incoming circuits). 4. ACR$V_PSI_ACCESS - Circuit used PSI-ACCESS. 5. ACR$V_PSI_FAIL - Access failed (for example, the call was rejected). 6. ACR$V_PSI_GATEWAY - The circuit used PSI as a gateway (Multi-host node). If neither ACR$V_PSI_OUT or ACR$V_PSI_PVC are set, then the circuit was an incoming SVC. For circuits that use either an X.25 gateway or VAX PSI Multi-host configuration, ACR$V_PSI_ACCESS will be set in the record logged on the PSI-ACCESS node and ACR$V_PSI_GATEWAY will be set in the record logged on the VAX PSI Multi-host node.
lcn	ACR$W_PSI_LCN	Logical channel number. (word)
out thru	ACR$W_PSI_OUT_THRU	Outgoing throughput class. (word)
inc thru	ACR$W_PSI_INC_THRU	Incoming throughput class. (word)
out window	ACR$W_PSI_OUT_WINDOW	Outgoing window size. (word)
inc window	ACR$W_PSI_INC_WINDOW	Incoming window size. (word)
out psize	ACR$W_PSI_OUT_PSIZE	Outgoing packet size. (word)
inc psize	ACR$W_PSI_INC_PSIZE	Incoming packet size. (word)
cug	ACR$W_PSI_CUG_NUMBER	Closed user group number. (word)

Table 11–8 (Cont.) Field Descriptions for ACR$K_PSI

Field	Symbolic Offset	Contents
fac req	ACR$W_PSI_FAC_REQ	Facilities requested flags (word). Flags defined in this field are: 1. ACR$V_PSI_REVCHG - Reverse charging. 2. ACR$V_PSI_FAST - Fast select. 3. ACR$V_PSI_CUG - Closed user group. 4. ACR$V_PSI_BCUG - Bilateral closed user group (this is set only if ACR$V_PSI_CUG is also set).
diag	ACR$B_PSI_DIAG	Clearing diagnostic. (byte)
cause	ACR$B_PSI_CAUSE	Cause for clearing. (byte)
reason	ACR$B_PSI_REASON	Reason for clearing. (byte)

IV
Reference Information

This part of the manual consists of three appendices:

- Appendix A - Contains a sample accounting program that analyzes VAX PSI accounting records.
- Appendix B - Contains the VAX PSI Installation Checkout Procedure (ICP) for previous versions of VAX PSI (VAX PSI V4.1 and earlier). You can use ICP to verify that the VAX PSI or VAX PSI Access software has been installed correctly on your system.
- Appendix C - Contains information about flow control parameter negotiation, and explains how VAX PSI negotiates packet and window sizes with the PSDN.

A

A Sample PSI Accounting Program

This appendix contains a sample program that analyzes PSI Accounting records. The program files are on the distribution kit and are copied to SYS$EXAMPLES. They are:

SYS$EXAMPLES:PSI$CHARGING.MAR	Program source code (Macro)
SYS$EXAMPLES:PSI$CHARGING.EXE	Executable image
SYS$LIBRARY:PSILIB.MLB	Symbol Definition Macros

The program reads records from the file PSIACCOUNTING.DAT, which should be in your default directory.

The program writes its output to the file PSIACCOUNTING.LIS, which it creates in your default directory.

You may wish to modify the program to suit your own requirements. You do this as follows:

1. Edit the source code.
2. Assemble and link the program in the normal way.
3. Run the executable image.

```
.TITLE        PSI_CHARGING - PSI CHARGING EXAMPLE
.IDENT        /4.0-00/

;
;                       COPYRIGHT (c) 1985 BY
;      DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS 01754
;
; THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND  COPIED
; ONLY  IN  ACCORDANCE  WITH  THE  TERMS  OF  SUCH  LICENSE AND WITH THE
; INCLUSION OF THE ABOVE COPYRIGHT NOTICE.  THIS SOFTWARE OR  ANY  OTHER
; COPIES  THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
; OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE  IS  HEREBY
; TRANSFERRED.
;
; THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE  WITHOUT  NOTICE
; AND  SHOULD  NOT  BE  CONSTRUED  AS  A COMMITMENT BY DIGITAL EQUIPMENT
; CORPORATION.
;
; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR  RELIABILITY  OF  ITS
; SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
;

;++
;
; Facility:
;
;       VAX PSI Example program
;
; Abstract:
;
;       This program is an example of a charging program to
;       analyse VAX PSI accounting records
;
; Environment:
;
;       VAX/VMS native, user mode
;
; Date:
;
;       2-Aug-1985
;
; Edit        Date        Reason
; ----        ----        ------
;
;--
        .PAGE
        .SBTTL        DEFINITIONS AND MACROS
        .ENABLE       SUPPRESSION
;
; Include:
;
        .LIBRARY      'SYS$LIBRARY:PSILIB'
```

```
;
;       Macro library calls
;

        $PSIDEF         ; Define VAX PSI symbols
        $SSDEF          ; Define status codes
        $ACRDEF         ; Accounting record definitions

        .PAGE
        .SBTTL          COSTS
        .PSECT          COSTS,NOWRT,NOEXE,LONG
;+
;
; These  locations  define the chargeable cost  (in arbitrary units)  of
; use  of  an  X.25 network. They are pairs of F-floating  numbers.  The
; first  number in the pair relates to successful calls, the  second  to
; unsuccessful calls.
;
; Note  that  there could be different costs for different  networks and
; there  could  be  different rates depending on other  parameters  (for
; example, window size). However, this program would have to be modified
; to deal with these.
;
; Information  on  different costs is available in  the  PSI  accounting
; record.
;
; The  algorithm  to which the numbers relate is described in  the  code
; below.
;
;-
BYTE_COST:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
SEGMENT_COST:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
; For example
;       .F_FLOATING         15E-5   ; Successful
;       .F_FLOATING         15E-5   ; Unsuccessful
PACKET_COST:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
MESSAGE_COST:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
SECOND_COST:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
; For example
;       .F_FLOATING         25E-4   ; Successful
;       .F_FLOATING         25E-4   ; Unsuccessful
```

```
BYTE_MINIMUM:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
SEGMENT_MINIMUM:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
; For example
;       .F_FLOATING         20      ; Successful
;       .F_FLOATING         16      ; Unsuccessful
PACKET_MINIMUM:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful
MESSAGE_MINIMUM:
        .F_FLOATING         0       ; Successful
        .F_FLOATING         0       ; Unsuccessful

        .PAGE
        .SBTTL          LOCAL DATA
        .PSECT          RWDATA,WRT,NOEXE,LONG
;
; Local data
;

;
; RMS control blocks - these are first to make them longword aligned
;

;
;       Input FAB
;
;       The input file is PSIACCOUNTING.DAT in the current
;       default directory.
;
ACC_FAB:
        $FAB            FAC=GET,-                   ; Only GETs required
                        FNM=<PSIACCOUNTING.DAT>,-   ; File name
                        FOP=SQO                     ; Sequential access only

;
;       Input RAB
;
ACC_RAB:
        $RAB            FAB=ACC_FAB,-               ; Associated FAB
                        ROP=RAH,-                   ; Read ahead
                        UBF=ACC_REC,-               ; User buffer address
                        USZ=ACC_REC_LEN             ; User buffer size
```

```
;
;       Output FAB
;
;       The output file is PSIACCOUNTING.LIS in the current
;       default directory.
;
OUT_FAB:
        $FAB        FAC=PUT,-                   ; Access for $PUTs
                    FNM=<PSIACCOUNTING.LIS>,-   ; File name
                    FOP=<TEF,-                  ; Truncate at eof
                    SQO>,-                      ; Sequential access only
                    RAT=CR,-                    ; CR carriage control
                    RFM=VAR                     ; Variable length
                                                ; records
;
;       Output RAB
;
OUT_RAB:
        $RAB        FAB=OUT_FAB,-               ; Associated FAB
                    RBF=OUT_REC,-               ; Output buffer
                    RSZ=OUT_REC_LEN,-           ; Record size
                    ROP=WBH                     ; Write behind

;
;       Accounting record buffer
;
ACC_REC_LEN =       512
ACC_REC:
        .BLKB       ACC_REC_LEN

;
;       Output record buffer
;
OUT_REC:
        .BLKB       132
OUT_REC_LEN = .-OUT_REC

;
;       FAO data areas
;
FAO_OUTLEN:         ; Length of FAO output
        .WORD         0
FAO_OUT_DESC:       ; FAO output descriptor
        .LONG         OUT_REC_LEN
        .ADDRESS OUT_REC
FAO_PRMLST:         ; FAOL parameter list
        .BLKL         30

;
;       Work area
;
;       The sets of three longwords are nonchargeable, chargeable
;       and total respectively.  The whole area is zeroed between
;       records.
;
WORK:               ; Start of area to be zeroed
```

```
BYTES:
        .BLKL       3
SEGMENTS:
        .BLKL       3
PACKETS:
        .BLKL       3
MESSAGES:
        .BLKL       3
COST:
        .BLKL       1               ; F-floating cost
TIME:
        .BLKQ       1               ; Elapsed time
SECONDS:
        .BLKL       1               ; As above but in seconds

WORK_LEN = .-WORK                   ; Length of area to be zeroed

;
;       Totals area
;
TOT_REC:            ; Records used
        .LONG         0
TOT_CHG_CALLS:      ; Chargeable calls
        .LONG         0
TOT_NCHG_CALLS:     ; Nonchargeable calls
        .LONG         0
TOT_CHG_BYTES:      ; Chargeable bytes
        .LONG         0
TOT_NCHG_BYTES:     ; Nonchargeable bytes
        .LONG         0
TOT_CHG_SEGMENTS:   ; Chargeable segments
        .LONG         0
TOT_NCHG_SEGMENTS:  ; Nonchargeable segments
        .LONG         0
TOT_CHG_PACKETS:    ; Chargeable packets
        .LONG         0
TOT_NCHG_PACKETS:   ; Nonchargeable packets
        .LONG         0
TOT_CHG_MESSAGES:   ; Chargeable messages
        .LONG         0
TOT_NCHG_MESSAGES:  ; Nonchargeable messages
        .LONG         0
TOT_TIME:           ; Elapsed time
        .QUAD         0
TOT_COST:           ; Cost
        .F_FLOATING   0
```

```
;
; Strings
;
        .PSECT          RODATA,NOWRT,NOEXE,LONG
HEADER1:
        .ASCII          /                       ELAPSED  /
        .ASCII          /< - - -  CHARGEABLE   - - - > /
        .ASCII          /< - -  NON-CHARGEABLE   - - > /
        .ASCII          /< - - - - - - TOTAL - - - - - > /
        .ASCII          //
HEADER1_LEN = .-HEADER1
HEADER2:
        .ASCII          /USERNAME     ACCOUNT  TIME     /
        .ASCII          / BYTES   SEGS.  PACKS. MESS.  /
        .ASCII          / BYTES   SEGS.  PACKS. MESS.  /
        .ASCII          / BYTES   SEGS.  PACKS. MESS.  /
        .ASCII          / COST/
HEADER2_LEN = .-HEADER2

HEADER3:
        .REPEAT         130
        .ASCII          /-/
        .ENDR
HEADER3_LEN = .-HEADER3

FAO_DESC:
        .ASCID          /!12AC !8AC !8%T !8SL !6SL !6SL !6SL !8SL
                         !6SL  !6SL !6SL !8SL !6SL !6SL !6SL !5SL.!2ZL/
TOTAL_DESC:
        .ASCID          /!21<TOTAL  !4SL  (!4SL)!> !8%T !8SL !6SL !6SL
                         !6SL !8SL !6SL !6SL !6SL !8SL !6SL !6SL !6SL
                         !5SL.!2ZL/
```

```
        .PAGE
        .SBTTL          MAIN ROUTINE
        .PSECT          CODE,NOWRT,EXE,LONG
;++
;
; Functional Description:
;
;       The program opens the input and output files and prints
;       the headers. It then loops through each record printing
;       the  fields  and calculating the cost.  At the end, the
;       total line is printed.
;
; Input parameters:
;
;       None
;
; Implicit inputs:
;
;       None
;
; Output parameters:
;
;       None
;
; Implicit outputs:
;
;       None
;
; Completion codes:
;
;       Any error status returned from a system service or an
;       RMS service SS$_NORMAL.
;
; Side effects:
;
;       None
;
;--
        .PAGE
        .SBTTL          INITIALISATION

        .ENTRY          PSI_CHARGING,^M<>

;+
; Entry point
;-

;+
;       Open accounting file
;-
        $OPEN           FAB=ACC_FAB
        BSBW            ERROR

        $CONNECT -
        RAB=ACC_RAB
        BSBW            ERROR
```

```
;+
;       Open output file
;-
        $CREATE         FAB=OUT_FAB
        BSBW            ERROR

        $CONNECT -
        RAB=OUT_RAB
        BSBW            ERROR

;+
;       Write headings
;-
        MOVAB           HEADER1,OUT_RAB+RAB$L_RBF          ; Start of buffer
        MOVW            #HEADER1_LEN,OUT_RAB+RAB$W_RSZ     ; Length of buffer
        $PUT            RAB=OUT_RAB                        ; Write
        BSBW            ERROR

        MOVAB           HEADER2,OUT_RAB+RAB$L_RBF          ; Start of buffer
        MOVW            #HEADER2_LEN,OUT_RAB+RAB$W_RSZ     ; Length of buffer
        $PUT            RAB=OUT_RAB                        ; Write
        BSBW            ERROR

        MOVAB           HEADER3,OUT_RAB+RAB$L_RBF          ; Start of buffer
        MOVW            #HEADER3_LEN,OUT_RAB+RAB$W_RSZ     ; Length of buffer
        $PUT            RAB=OUT_RAB                        ; Write
        BSBW            ERROR

        .PAGE
        .SBTTL          RECORD PROCESSING
;+
;       Main loop
;-
NEXT_RECORD:
        $GET            RAB=ACC_RAB                ; Read record
        BLBS            R0,20$                     ; If LBS - read was ok
        CMPL            R0,#RMS$_EOF               ; Is it end of file ?
        BNEQ            10$                        ; If EQL - print totals
        BRW             PRINT_TOTAL
10$:
        BSBW            ERROR                      ; Show the error

20$:
```

```
;+
;       Check if this is a PSI record
;-
        CMPZV           #ACR$V_VERSION,#ACR$S_VERSION,-
                        ACC_REC+ACR$W_TYPE,#ACR$K_VERSION3
                                                ; Is this a format
                                                ; I understand ?
        BNEQ            NEXT_RECORD             ; If NE - no
        CMPZV           #ACR$V_TYPE,#ACR$S_TYPE,-
        ACC_REC+ACR$W_TYPE,#ACR$K_PSI           ; Is this a PSI record ?
        BNEQ            NEXT_RECORD             ; If NE - no
        CMPZV           #ACR$V_SUBTYPE,#ACR$S_SUBTYPE,-
                        ACC_REC+ACR$W_TYPE,#ACR$K_PSI_VCT
                                                ; Is it a circuit
                                                ; termination ?
        BNEQ            NEXT_RECORD             ; If NE - no

;+
;       It is a PSI circuit termination record
;
;       Find the start of the PSI circuit termination packet
;-
        MOVZWL          ACC_REC+ACR$W_LENGTH,R8 ; Length of record
        ADDL2           #ACC_REC,R8             ; End of the record
        MOVAB           ACC_REC+ACR$K_HDRLEN,R7 ; Start of first packet
30$:    CMPZV           #ACR$V_TYPE,#ACR$S_TYPE,-
                        ACR$W_TYPE(R7),#ACR$K_PSI ; PSI packet ?
        BNEQ            40$                     ; If NE - no, try again
        MOVAB           (R7),R9                 ; Save start of PSI
                                                ; packet
        BRB             50$
40$:    CMPZV           #ACR$V_TYPE,#ACR$S_TYPE,-
                        ACR$W_TYPE(R7),#ACR$K_ID ; Id packet ?
        BNEQ            50$                     ; If NE - no, try again
        MOVAB           (R7),R10                ; Save start of ID
                                                ; packet
50$:    MOVZWL          ACR$W_LENGTH(R7),R1     ; Length of this packet
        ADDL2           R1,R7                   ; Try next packet
        CMPL            R7,R8                   ; End of record ?
        BLSS            30$                     ; If LS - not end of
                                                ; record, try again

        MOVAB           ACC_REC,R8              ; Get start of record
                                                ; in register
;+
;       R8 = start address of record
;       R9 = start address of PSI packet
;       R10 = start address of ID packet
;-
```

```
;+
;       Decide whether call is chargeable
;
;       The agorithm used here is that the call is chargeable if
;       it  is an outgoing SVC without reverse  charging  or  an
;       incoming SVC with reverse charging. PVCs are not treated
;       as chargeable  (in this example, there is a fixed charge
;       for a PVC and all data is free).
;
;-
        CLRL            R7                              ; Assume not chargeable
        BBS             #ACR$V_PSI_PVC,ACR$W_PSI_VCTYPE(R9),PVC
                                                        ; Branch if the
                                                        ; circuit is a PVC
        BBS             #ACR$V_PSI_OUT,ACR$W_PSI_VCTYPE(R9),OUTGOING
                                                        ; Branch if the
                                                        ; circuit is outgoing
; The circuit must be incoming
        BBS             #ACR$V_PSI_REVCHG,ACR$W_PSI_FAC_REQ(R9),CHARGE
; Chargeable if "reverse charging"
        BRB             NOCHARGE

OUTGOING:
        BBC             #ACR$V_PSI_REVCHG,ACR$W_PSI_FAC_REQ(R9),CHARGE
; Chargeable if not "reverse charging"
        BRB             NOCHARGE

CHARGE:
        INCL            R7                              ; R7 indicates whether
                                                        ; chargeable

PVC:                                                    ; PVC is not chargeable

NOCHARGE:

;+
;       Fill in work area from record
;-
        MOVC5   #0,#0,#0,#WORK_LEN,WORK                 ; Zero work area

        MOVL    ACR$L_PSI_BYTES_TX(R9),BYTES[R7]; Bytes transmitted
                                                        ; Index with R7 to fill
                                                        ; in chargeable or
                                                        ; nonchargeable area
        ADDL2   ACR$L_PSI_BYTES_RX(R9),BYTES[R7]; Bytes received
        ADDL3   BYTES,BYTES+4,BYTES+8                   ; Total

        MOVL    ACR$L_PSI_SEGMENTS_TX(R9),SEGMENTS[R7]
        ADDL2   ACR$L_PSI_SEGMENTS_RX(R9),SEGMENTS[R7]
        ADDL3   SEGMENTS,SEGMENTS+4,SEGMENTS+8

        MOVL    ACR$L_PSI_PACKETS_TX(R9),PACKETS[R7]
        ADDL2   ACR$L_PSI_PACKETS_RX(R9),PACKETS[R7]
        ADDL3   PACKETS,PACKETS+4,PACKETS+8

        MOVL    ACR$L_PSI_MESSAGES_TX(R9),MESSAGES[R7]
        ADDL2   ACR$L_PSI_MESSAGES_RX(R9),MESSAGES[R7]
        ADDL3   MESSAGES,MESSAGES+4,MESSAGES+8
```

```
;       Elapsed time in seconds
        TSTL    ACR$Q_PSI_START_TIME(R9)         ; Valid start time ?
        BNEQ    10$                              ; If NEQ - then yes
        TSTL    ACR$Q_PSI_START_TIME+4(R9)       ; Test other longword
        BEQL    20$                              ; If EQL - time not
                                                 ; valid, duration zero,
                                                 ; leave TIME and
                                                 ; SECONDS as 0
10$:
        MOVQ    ACR$Q_SYSTIME(R8),TIME           ; Get finish time
        SUBL    ACR$Q_PSI_START_TIME(R9),TIME    ; Subtract low longword
        SBWC    ACR$Q_PSI_START_TIME+4(R9),TIME+4
                                                 ; Subtract high longword
        EDIV    #10000000,TIME,SECONDS,R0        ; Divide to get seconds
        TSTL    R0                               ; Any remainder ?
        BEQL    20$                              ; If EQL - no
        INCL    SECONDS                          ; Round up
20$:

;+
;       Work out cost
;
;       The algorithm used here assumes that there is duration
;       element and a data volume element which has a  minimum
;       which  depends on whether the call was  successful. To
;       simplify  matters, this example assumes the rate  does
;       not  vary  with  the time of day.
;
;       If necessary, that information could be obtained  from
;       the ACR$Q_PSI_START_TIME field.
;
;       This algorithm is based on the tariff used by  British
;       Telecom at the time of writing and the  simplification
;       above is appropriate for lines with high usage  (which
;       get charged at a flat rate).
;-

        BLBS    R7,25$                   ; If LBC, the call is
                                         ; not chargeable
        BRW     FORMAT
25$:

        CLRL    R1                       ; Assume call did not fail
        BBC     #ACR$V_PSI_FAIL,ACR$W_PSI_VCTYPE(R9),40$
                                         ; If bit clear, call did
                                         ; not fail
        INCL    R1                       ; Use other table
40$:

;+
;       Duration element
;-
```

```
        BBS     #ACR$V_PSI_PVC,ACR$W_PSI_VCTYPE(R9),30$
                                        ; No duration charge for PVCs
        CVTLF   SECONDS,R0              ; Chargeable seconds
        MULF2   SECOND_COST[R1],R0      ; Cost of time
        ADDF2   R0,COST
30$:

;+
;       Volume element
;-
        CVTLF   BYTES+4,R0              ; Chargeable bytes
        CMPF    R0,BYTE_MINIMUM[R1]     ; Compare with the correct
                                        ; minimum
        BGEQ    50$                     ; If GEQ, then use R0
        MOVF    BYTE_MINIMUM[R1],,R0    ; Use minimum
50$:    MULF2   BYTE_COST[R1],R0        ; Cost of bytes
        ADDF2   R0,COST

        CVTLF   SEGMENTS+4,R0           ; Chargeable segments
        CMPF    R0,SEGMENT_MINIMUM[R1]  ; Compare with the correct
                                        ; minimum
        BGEQ    60$                     ; If GEQ, then use R0
        MOVF    SEGMENT_MINIMUM[R1],R0  ; Use minimum
60$:    MULF2   SEGMENT_COST[R1],R0     ; Cost of segments
        ADDF2   R0,COST

        CVTLF   PACKETS+4,R0            ; Chargeable packets
        CMPF    R0,PACKET_MINIMUM[R1]   ; Compare with the correct
                                        ; minimum
        BGEQ    70$                     ; If GEQ, then use R0
        MOVF    PACKET_MINIMUM[R1],R0   ; Use minimum
70$:    MULF2   PACKET_COST[R1],R0      ; Cost of packets
        ADDF2   R0,COST

        CVTLF   MESSAGES+4,R0           ; Chargeable messages
        CMPF    R0,MESSAGE_MINIMUM[R1]  ; Compare with the correct
                                        ; minimum
        BGEQ    80$                     ; If GEQ, then use R0
        MOVF    MESSAGE_MINIMUM[R1],R0  ; Use minimum
80$:    MULF2   MESSAGE_COST[R1],R0     ; Cost of messages
        ADDF2   R0,COST
```

```
;+
;       Format the record
;-
FORMAT:
        MOVAB   FAO_PRMLST,R6           ; Get parameter list address
        MOVZWL  ACR$W_USERNAME(R10),R1  ; Offset of username
                                        ; (ID packet)
        ADDL3   R10,R1,(R6)+            ; Add base of ID packet and
                                        ; put address in param list
        MOVZWL  ACR$W_ACCOUNT(R10),R1   ; Offset of account
        ADDL3   R10,R1,(R6)+            ; Address of account
        MOVAB   TIME,(R6)+              ; Elapsed time
        MOVL    BYTES+4,(R6)+           ; Chargeable bytes
        MOVL    SEGMENTS+4,(R6)+        ; Chargeable segments
        MOVL    PACKETS+4,(R6)+         ; Chargeable packets
        MOVL    MESSAGES+4,(R6)+        ; Chargeable messages
        MOVL    BYTES,(R6)+             ; Nonchargeable bytes
        MOVL    SEGMENTS,(R6)+          ; Nonchargeable segments
        MOVL    PACKETS,(R6)+           ; Nonchargeable packets
        MOVL    MESSAGES,(R6)+          ; Nonchargeable messages
        MOVL    BYTES+8,(R6)+           ; Total bytes
        MOVL    SEGMENTS+8,(R6)+        ; Total segments
        MOVL    PACKETS+8,(R6)+         ; Total packets
        MOVL    MESSAGES+8,(R6)+        ; Total messages

;       The cost must be spilt into two integers (longword)
;       for fao to format as nnn.nn. The first integer
;       is the integral part.
        CVTFL   COST,R1                 ; Integer part of cost
        MOVL    R1,(R6)+                ; In parameter list
;       The second integer is the integral part of the result
;       of multiplying the fractional part by 100
        CVTLF   R1,R1                   ; Integer part of cost
        SUBF3   R1,COST,R1              ; Get fractional part of cost
        MULF2   #1E2,R1                 ; Multiply by 100
        CVTRFL  R1,(R6)+                ; In parameter list (rounded)

        $FAOL_S CTRSTR=FAO_DESC,-       ; Control string
                OUTLEN=FAO_OUTLEN,-     ; Output string length
                OUTBUF=FAO_OUT_DESC,-   ; Output string descriptor
                PRMLST=FAO_PRMLST       ; Parameter list
        BSBW    ERROR

        MOVAB   OUT_REC,OUT_RAB+RAB$L_RBF       ; Fill in record
        MOVW    FAO_OUTLEN,OUT_RAB+RAB$W_RSZ    ; buffer address
                                                ; and size

;+
;       Print the record
;-
        $PUT    RAB=OUT_RAB
        BSBW    ERROR
```

```
;+
;       Increment totals
;-
        BLBC    R7,90$                  ; If LBC - not chargeable
        INCL    TOT_CHG_CALLS
90$:
        INCL    TOT_REC

        ADDL2   BYTES,TOT_NCHG_BYTES    ; Non chargeable bytes
        ADDL2   BYTES+4,TOT_CHG_BYTES   ; Chargeable bytes

        ADDL2   SEGMENTS,TOT_NCHG_SEGMENTS
                                        ; Non chargeable segments
        ADDL2   SEGMENTS+4,TOT_CHG_SEGMENTS
                                        ; Chargeable segments
        ADDL2   PACKETS,TOT_NCHG_PACKETS
                                        ; Non chargeable packets
        ADDL2   PACKETS+4,TOT_CHG_PACKETS
                                        ; Chargeable packets
        ADDL2   MESSAGES,TOT_NCHG_MESSAGES
                                        ; Non chargeable messages
        ADDL2   MESSAGES+4,TOT_CHG_MESSAGES
                                        ; Chargeable messages

        ADDL2   TIME,TOT_TIME           ; Elapsed time
        ADWC    TIME+4,TOT_TIME+4

        ADDF2   COST,TOT_COST           ; Cost

        BRW     NEXT_RECORD

        .PAGE
        .SBTTL  TOTAL PROCESSING
;+
;       Format and print the total line
;-
PRINT_TOTAL:

;+
;       Close input file
;-
        $CLOSE  FAB=ACC_FAB
        BSBW    ERROR

;+
;       Print separator line
;-

        MOVAB   HEADER3,OUT_RAB+RAB$L_RBF       ; Start of buffer
        MOVW    #HEADER3_LEN,OUT_RAB+RAB$W_RSZ  ; Length of buffer
        $PUT    RAB=OUT_RAB                     ; Write
        BSBW    ERROR
```

```
;+
;       Move fields into the work area and calculate totals
;-
        MOVL    TOT_NCHG_BYTES,BYTES            ; Nonchargeable bytes
        MOVL    TOT_CHG_BYTES,BYTES+4           ; Chargeable bytes
        ADDL3   BYTES,BYTES+4,BYTES+8           ; Total bytes

        MOVL    TOT_NCHG_SEGMENTS,SEGMENTS      ; Nonchargeable segments
        MOVL    TOT_CHG_SEGMENTS,SEGMENTS+4     ; Chargeable segments
        ADDL3   SEGMENTS,SEGMENTS+4,SEGMENTS+8  ; Total segments

        MOVL    TOT_NCHG_PACKETS,PACKETS        ; Nonchargeable packets
        MOVL    TOT_CHG_PACKETS,PACKETS+4       ; Chargeable packets
        ADDL3   PACKETS,PACKETS+4,PACKETS+8     ; Total packets

        MOVL    TOT_NCHG_MESSAGES,MESSAGES      ; Nonchargeable messages
        MOVL    TOT_CHG_MESSAGES,MESSAGES+4     ; Chargeable messages
        ADDL3   MESSAGES,MESSAGES+4,MESSAGES+8  ; Total messages

        MOVF    TOT_COST,COST                   ; Total cost
        MOVQ    TOT_TIME,TIME                   ; Total time

;+
;       Assemble the FAO parameter list
;-
        MOVAB   FAO_PRMLST,R6                   ; Get parameter list
                                                ; address
        MOVL    TOT_REC,(R6)+                   ; Total calls
        MOVL    TOT_CHG_CALLS,(R6)+             ; Chargeable calls
        MOVAB   TIME,(R6)+                      ; Elapsed time
        MOVL    BYTES+4,(R6)+                   ; In FAO list
        MOVL    SEGMENTS+4,(R6)+                ; Chargeable segments
        MOVL    PACKETS+4,(R6)+                 ; Chargeable packets
        MOVL    MESSAGES+4,(R6)+                ; Chargeable messages
        MOVL    BYTES,(R6)+                     ; In FAO list
        MOVL    SEGMENTS,(R6)+                  ; Nonchargeable segments
        MOVL    PACKETS,(R6)+                   ; Nonchargeable packets
        MOVL    MESSAGES,(R6)+                  ; Nonchargeable messages
        MOVL    BYTES+8,(R6)+                   ; In FAO list
        MOVL    SEGMENTS+8,(R6)+                ; Total segments
        MOVL    PACKETS+8,(R6)+                 ; Total packets
        MOVL    MESSAGES+8,(R6)+                ; Total messages
;       The cost must be spilt into two integers (longword) for fao to
;       format as nnn.nn.
;       The first integer is the integral part.
        CVTFL   COST,R1                         ; Integer part of cost
        MOVL    R1,(R6)+                        ; In parameter list
;       The second integer is the integral part of the result of
;       multiplying the fractional part by 100
        CVTLF   R1,R1                           ; Integer part of cost
        SUBF3   R1,COST,R1                      ; Get fractional part
                                                ; of cost
        MULF2   #1E2,R1                         ; Multiply by 100
        CVTFL   R1,(R6)+                        ; In parameter list
```

```
        $FAOL_S CTRSTR=TOTAL_DESC,-             ; Control string
                OUTLEN=FAO_OUTLEN,-             ; Output string length
                OUTBUF=FAO_OUT_DESC,-           ; Output string
                                                ; descriptor
                PRMLST=FAO_PRMLST               ; Parameter list
        BSBW    ERROR

        MOVAB   OUT_REC,OUT_RAB+RAB$L_RBF       ; Fill in record buffer
                                                ; address
        MOVW    FAO_OUTLEN,OUT_RAB+RAB$W_RSZ    ; and size

;+
;       Print the record
;-
        $PUT    RAB=OUT_RAB
        BSBW    ERROR

;+
;       Close the output file
;-
        $CLOSE  FAB=OUT_FAB
        BSBW    ERROR

        $EXIT_S CODE=#SS$_NORMAL

        .PAGE
        .SBTTL  ERROR CHECKING SUBROUTINES

;+
;       If any errors occur, the program exits with the error status
;-
ERROR:                                  ; Check system service status
                                        ; (in R0)
        BLBC    R0,ERR                  ; If low bit clear then error
                                        ; occurred
        RSB                             ; No errors so return

ERR:
        $EXIT_S -                       ; Error occurred so exit
        CODE = R0                       ;  with status

        .END    PSI_CHARGING            ; Start of main routine
```

B

The VAX PSI Installation Checkout Procedure

This appendix explains how to run the old Installation Checkout Procedure (ICP).

NOTE

- Use the old ICP to test the configuration of your VAX PSI system if your system OR any remote system you want to use for the test has VAX PSI V4.1 (or earlier) installed.
- Use the new Configuration Test Program (CTP) to test the configuration of your VAX PSI system if your system AND any remote system you want to use for the test have VAX PSI V4.2 (or later) installed.

  See Chapter 3 for information about the CTP.

B.1 Introduction

The ICP allows you to check that the VAX PSI or VAX PSI Access software has been installed and configured correctly on your system. When in use, the ICP sets up your terminal as a network operator's console and turns on logging of all PSI-related events (in order to provide as much information as possible in the case of errors). For details of event logging, refer to the *VAX P.S.I Problem Solving Guide.*

The ICP is made up of an automated DCL command procedure called PSI_ICP and the programs PSI$XTR (X.25 Test Receiver) and PSI$XTS (X.25 Test Sender). PSI$XTR and PSI$XTS are used to verify that the installation can set up a call and transfer data between the local node and another DTE in the network. The other DTE can be a remote machine running PSI$XTR under either VAX PSI or RSX-11 PSI. (This can be one of your own machines or any other suitable machine.)

In order to run the ICP, you must subscribe to Open User Group and have at least one Switched Virtual Circuit (SVC). Alternatively, if you have two SVCs, you can run PSI$XTR on your system and loop the test through the PSDN (and Multi-host node if necessary).

NOTE

This chapter also explains how to check installation of the X.29 software (see Section B.5).

B.2 Preparing to Run PSI_ICP.COM

Before running PSI_ICP.COM, make sure that VAX PSI is installed, configured and running, and that at least one DTE is up. If you are checking an Access system, also ensure that at least one Multi-host PSI system on your DECnet network is reachable and ready for use.

Check that a DTE is running by using the following command:

```
NCP>SHOW MODULE X25-PROTOCOL KNOWN DTES
```

If you are running the ICP on an Access system, the Multi-host PSI system is ready for use when the Multi-host PSI system software has been installed and its ICP has been executed successfully. To find out whether the Multi-host PSI system is reachable and that at least one DTE is up on that system, use the command:

```
NCP>TELL node-id SHOW MODULE X25-PROTOCOL KNOWN DTES
```

where *node-id* is the name of the node running Multi-host PSI.

If a DTE is up, it appears in the summary display produced by the command with a state and substate of ON-RUNNING.

If no DTEs are up, wait for two minutes before issuing the command again. If the display shows that there are still no DTEs up, check both the DTE address and the physical connection to the PSDN. If everything appears to be in order, confirm with the network authority that the DCE is operational. If you need to investigate the problem further, refer to the *VAX P.S.I. Problem Solving Guide.*

If you are checking an Access system, make sure that the specified destination parameters are correct by issuing the following command:

```
NCP> TELL node-id SHOW MODULE X25-SERVER DESTINATION x
```

where *node-id* is the name of the node running Multi-host PSI, and *x* is the target system.

Check the displayed details against the information supplied in the *VAX P.S.I. Installation Procedures* and amend as necessary.

Finally, ensure that the user account specified to execute PSI$XTR will be able to create the necessary log files in its default directory, that is, that the protection rights on the directory permit Owner-Write access. In addition, check the disk quota to make sure that there are at least 500 blocks remaining to allow for the creation of the log files.

B.3 Starting PSI_ICP.COM

To start PSI_ICP.COM, issue the following command:

```
$ @SYS$TEST:PSI_ICP.COM
```

The procedure displays the following message:

```
This command file runs the PSI ICP. It assumes that at least one DTE
is up and running.
Network name  (string):
```

Insert the name of your network. (If the network you wish to use is defined as the default, you may enter <RET>.) The procedure then asks:

```
Do you want to run the ICP to the local node [YES]?
```

If you want to loop the test back to your local node, answer YES. The procedure then asks you for the local DTE address (see Section B.3.1).

If you want to run the ICP between your local node and a remote node running PSI$XTR, answer NO and refer to Section B.3.2.

NOTE

For details of how PSI_ICP.COM completes, refer to Section B.4, which also describes the two log files created by the procedure.

B.3.1 Running the ICP to the Local DTE

If you answered YES to the local node question, the procedure asks:

```
Local DTE address  (DTE-address):
```

Specify the DTE address as a string of from 1 to 15 digits. Refer to the *Public Network Information* manual for details of the DTE address format for your network. The procedure displays the following message:

```
The next series of questions provide information used to set up the
DESTINATION used in the test. The subaddress is optional.

Subaddress  (2-4 digits):
Username  (string):
Password  (string):
Verification  (string):
```

Enter the subaddress of the local DTE to be used in the test. Answer the other questions as if you were logging on to a terminal. Enter a valid current account name and the associated password that will be used to run PSI$XTR. (You will also be asked to verify the password.)

B.3.2 Running the ICP to a Remote DTE

If you answered NO to the local node question, the procedure displays the following message:

```
The following questions determine the DTE address and subaddress of
the remote test machine.

Remote DTE address (DTE-address):

Remote subaddress (2-4 digits):
```

The remote DTE must be a machine running PSI$XTR under either VAX PSI or RSX-11 PSI. This can be one of your own machines or any other suitable machine, but you must subscribe to the Open User Group and have at least one SVC.

Specify the remote DTE address as a string of from 1 to 15 digits. Refer to the *Public Network Information* manual for details of the DTE address format for your network. Specify the subaddress, if required, as a string of from 2 to 4 digits.

Before you can execute the ICP, you must set up a destination and an object on the remote machine that is to receive the test. The following are examples of the NCP commands you use to set up a destination and object at the remote node:

```
NCP>SET OBJECT PSI$XTR FILE SYS$TEST:PSI$XTR USER FRED -
    PASSWORD ICPTEST NUMBER 0
NCP>SET MODULE X25-SERVER DESTINATION PSI$XTR OBJECT -
    PSI$XTR SUBADDRESS 30
```

B.4 Completing PSI_ICP.COM

Once you have specified the appropriate DTE address, the procedure displays:

```
Starting PSI_ICP test.
```

NOTE

Disregard any information or warning messages issued while the ICP is running.

Before the test completes, the procedure displays a set of statistics describing what happened during the test. The following is an example of a set of test statistics:

```
Number of Transmits    =  250
Number of Receives     =  234
Interrupts completed   =  0
Interrupts received    =  0
Interrupts confirmed   =  0
Resets completed       =  0
Resets received        =  1
```

When the test has finished, the procedure displays either:

```
VAX PSI V4.1 ICP completed successfully.
```

or:

```
%NONAME-W-NOMSG, Message number 00000000
VAX PSI V4.1 ICP finished with errors.
```

PSI_ICP.COM creates two log files showing the progress of the ICP. After the test has finished, the procedure displays:

Two test log files have been generated. They are X25TEST.LOG in your current directory, and X25TEST.LOG in the PSI$XTR default directory.

With both VAX PSI and VAX PSI Access, if you run the ICP to a remote node, PSI$XTR is created on that node. The PSI$XTR default directory is the SYS$LOGIN directory of the account specified by the PSI$XTR object.

If you are running the ICP to the local DTE, the PSI$XTR default directory is in the default directory for the account you specified.

If the ICP completes with errors, check the log files by either printing them or displaying them at a terminal. Look for information and warning messages in the log files which indicate protocol errors. The log files will inform you whether or not the ICP was successful. If the ICP was successful, the log files will show that at least 200 data transmits and 200 data receives took place during the 5 minute duration of the ICP.

You can use the Trace utility to help you diagnose problems. See the *VAX P.S.I. Problem Solving Guide* for details.

You are advised to take advantage of the extensive status and error counters maintained by VAX PSI to help diagnose problems. If you are checking a Native/Multi-host mode installation, issue the following command:

```
NCP>SHOW LINE... line-id COUNTERS
```

You may also need to test the physical line using line-level loopback tests. Refer to the *VMS Network Control Program Manual* for details.

Refer to your PSDN authorities and to your software support specialist if you need further assistance.

B.4.1 Checklist of Possible Errors

PSI_ICP.COM may complete with errors for many different reasons. This section provides a checklist of the most likely reasons.

NOTE

For help with problem diagnosis and solutions, refer to the *VAX P.S.I. Problem Solving Guide.*

1. You may have entered an incorrect DTE address or subaddress. Check these values. If they are incorrect, try the test again.
2. You may have installed the software incorrectly. Check the Vector address, CSR address and adapter number. If the test fails again, contact your software support specialist.

3. In the case of an Access system, you may have made a mistake when installing the Access software. Check that you specified the network name and Multi-host PSI system name correctly. Check also that the Multi-host PSI system connects to the network you specified. If the test fails again, contact your software support specialist.
4. When checking an Access system - the DECnet or the Multi-host PSI system may not be up. Check that the Multi-host PSI system node is reachable with NCP commands. If not, wait until it becomes reachable, and try the test again.
5. The PSDN may not be up. Try the test again later, and if it fails, contact your software support specialist.
6. The device may not be connected to the modem or the modem may not be working correctly. Check the hardware and try the test again.
7. PSI$XTR may not be set up at the remote end (for example, no destination, task not installed and so on). Set up PSI$XTR correctly at the remote system, (see Section B.3.2) or contact your software support specialist.

B.5 Verifying an X.29 Installation

To verify that the system has been configured to support X.29, issue the SET HOST/X29 command. If the PSIPAD utility is not present, an appropriate error message will be issued. See the *VAX P.S.I. PAD and MAIL Utilities Manual* for details of this command.

NOTE

If you are using a machine other than a MicroVAX and you want to verify the X.29 installation from the console terminal, you may enter PAD command mode in one of two ways, as follows:

1. Turn the keylock rotary switch on the VAX processor control panel to the LOCAL DISABLE position, and then type <CTRL/P>
2. Type <CTRL/Y> twice.

DO NOT type <CTRL/P> without previously setting the processor to LOCAL DISABLE.

Do the following to test the X.29 installation:

1. Issue the SET HOST/X29 command to make a call from your system through the PSDN and back to your system.
2. Connect to the local DTE, specifying the correct call information, and log in to an account on your VAX system.

3. Engage the VAX in a simple dialogue by using a few standard VAX/VMS commands, such as:
   - DIRECTORY
   - SHOW SYSTEM
4. Log out.

See the technical handbook produced by the network authorities for your PSDN for more details on the above steps.

C

Flow Control Parameter Negotiation in VAX PSI

This appendix explains how VAX PSI uses flow control parameter negotiation to agree with your PSDN on the window and packet sizes to be used for both directions of data transfer between your DTE(s) and the network DCE.

C.1 Flow Control Parameter Negotiation

Flow control parameter negotiation is the way in which VAX PSI agrees on the parameters to be used for data transfer for the duration of an SVC connection between your DTE(s) and the network DCE. The parameters negotiated are the data packet and size and window size.

Note that flow control parameter negotiation is an optional facility offered by some PTT authorities and to which you can subscribe over a period of time. To find out if your PSDN offers flow control parameter negotiation, refer to the *Public Network Information* manual. Flow control parameter negotiation is not usually available for private packet switching networks.

If your PSDN offers this facility, and you subscribe to it, the PSDN and VAX PSI negotiate flow control parameters between the network DCE and your DTE(s) for each incoming and outgoing call.

Negotiation is achieved by using the negotiation facility fields in the Incoming Call Packet (ICP) or Call Request Packet (CRP) and the Call Accepted Packet (CAP) or Call Connected Packet (CCP).

For incoming calls, the network DCE sends an ICP to VAX PSI, and VAX PSI replies with a CAP. For outgoing calls, VAX PSI sends a CRP to the network DCE, which replies with a CCP.

In the following example, an outgoing call from VAX PSI to a network DCE is used to explain how negotiation takes place, using the CRP and CCP. A similar procedure is used for incoming calls from the network DCE to VAX PSI, using the ICP and CCP.

To request negotiation for an outgoing call, VAX PSI inserts its chosen parameter values and a facility code in the corresponding facility field of the CRP. The network DCE answers by either inserting parameter values it wishes to use in the CCP, or by accepting the values in the CRP. In the later case, the DCE does not insert values in the CCP.

Negotiation is always towards the default packet and window sizes. Generally, this is "downwards"; that is, the returned parameter values are either equal to or lower than the ones requested. This is because it is usual for the call initiator to request large packet and window sizes, and the call receiver to request smaller packet and window sizes.

Table C–1 shows the valid parameter requests for outgoing call negotiation. Note that the valid parameters are the same for incoming calls, except that the packet names are different. The CRP and CCP for outgoing calls become an ICP and CAP for incoming calls.

Table C–1 Valid Flow Control Parameter Sizes for Outgoing Calls

Call Request Packet (CRP)	Notes	Call Confirm Packet (CCP)
WINDOW(CRP) >= 2	(1)	**WINDOW**(CRP) >= **WINDOW**(CCP) **WINDOW**(CCP) >= 2
WINDOW(CRP) = 1	(2)	**WINDOW**(CCP) = 1 or 2
PACKET(CRP) >= 128	(3)	**PACKET**(CRP) >= **PACKET**(CCP) **PACKET**(CCP) >= 128
PACKET(CRP) < 128		128 >= **PACKET**(CCP) **PACKET**(CCP) >= **PACKET**(CRP)

Notes - see Table C–1

1. If negotiation is not requested in the CRP or ICP, the default window size is used. In Table C–1, the default window size is 2. This is the usual default window size for most PSDNs.
2. In Table C–1, the minimum window size for the CRP is 1. This is the usual minimum window size for most PSDNs.
3. If negotiation is not requested in the CRP or ICP, the default packet size is used. In Table C–1, the default packet size is 128. This is the usual default packet size for most PSDNs.

For more information about the default values for your PSDN, see the *Public Network Information* manual.

If flow control parameter negotiation is not requested, the default parameter values are used. VAX PSI and the network DCE MUST agree on these default values. Normally the default parameter values are PSDN specific, and are defined with the maximum parameter values in the network profile for each PSDN supported by VAX PSI. You can change these values using NCP.

NOTE

If you use NCP to change a parameter value, you must ensure that the value you select is valid for your network. See the *Public Network Information* manual for more information about the range of values you can select.

You must also ensure that the value you select is appropriate for flow control parameter negotiation between VAX PSI and the network DCE; that is, the value that you have agreed to use when 'you subscribed to your PSDN.

C.2 Determining Whether Your System Allows or Disallows Flow Control Parameter Negotiation

VAX PSI allows flow control parameter negotiation if your local DTE has a maximum default data window that is greater than the minimum value of the profile for the network with which the local DTE is associated. If these values are equal, then VAX PSI will not allow flow control parameter negotiation.

To check the minimum value of the profile for your network, refer to the *Public Network Information* manual. This manual shows the range of values for the packet and window sizes for each network supported by VAX PSI.

To check the default and maximum values of the packet and window sizes for your system, enter the following NCP command:

```
NCP>SHOW KNOWN DTE KNOWN NETWORK CHARACTERISTICS
```

This command displays characteristics of the local DTE(s) and network(s) in full, including the required parameters, as follows:

```
Known Module X25-Protocol Volatile Characteristics as of
  14-JAN-1988 12:20:20

DTE               = 234212345678
Network           = PSS
.
.
.
Default data      = 128
Default window    = 2
Maximum data      = 1024
Maximum window    = 7
.
.
.
```

The *Default data* and *Default window* parameters are copied from the profile *default-size* values when PSI is loaded onto your system.

The *Maximum data* and *Maximum window* parameters are copied from the profile *default-maximum-size* values when PSI is loaded onto your system.

C.2.1 Enabling Flow Control Parameter Negotiation for Your Local DTE(s)

To enable flow control parameter negotiation for your local DTE(s), increase the DTE *maximum data* and *maximum window* parameters to the highest values required; the maximum values are shown in the *Public Network Information* manual for your network. This will make the DTE *maximum data* and *maximum window* greater than the minimum values for the profile, so enabling flow control parameter negotiation.

C.2.2 Disabling Flow Control Parameter Negotiation for Your Local DTE(s)

To disable flow control parameter negotiation for your local DTE(s), decrease the DTE *maximum data* and *maximum window* parameters to the lowest values required; the minimum values are shown in the *Public Network Information* manual for your network.

C.3 Incoming and Outgoing Negotiation Requests

There are two types of negotiation request, incoming and outgoing.

C.3.1 Incoming Negotiation Requests

If flow control parameter negotiation is enabled, VAX PSI always answers incoming flow control negotiation requests and tries to agree on the values within a range defined in the X25-Protocol DTE database.

If flow control parameter negotiation is disabled, VAX PSI will always answer the ICP with flow control parameter negotiation facilities, using the default values; that is, VAX PSI will refuse to negotiate.

C.3.2 Outgoing Negotiation Requests

Outgoing flow control parameter negotiation may be requested by a VAX PSI user or application program (for example, DECnet in the case of DLM, or the host based PAD). Negotiation will or will not be requested in the Call Request Packet sent to the network DCE, depending on whether flow control parameter negotiation is enabled for the DTE.

If flow control parameter negotiation is enabled, and the values requested are within the defined range, VAX PSI will request flow control parameter negotiation in the CRP. If it is disabled, VAX PSI will not request flow control parameter negotiation in the CRP.

If VAX PSI sends a negotiation request to the DCE and this particular DCE does not support flow control parameter negotiation for that DTE (usually a DTE subscription option), then the call will be cleared with cause and diagnostic codes indicating a local procedure error or an invalid facility request.

Some DEC supplied applications (for example, PSI mail and DLMs) may request flow control parameter negotiation. If a particular DTE subscription does not include flow control parameter negotiation, calls may fail, as described above. If this occurs, you may change the DTE characteristics manually using NCP, to avoid VAX PSI requesting flow control parameter negotiation in the CRP.

Conversely, if your profile is set up so that flow control parameter negotiation is disabled, but your network does support flow control parameter negotiation and you have subscribed to it, then you may change the DTE characteristics using NCP to allow VAX PSI to request flow control parameter negotiation in the CRP.

Minimum value = maximum value

In this case, no negotiation will be requested. This is the proposed profile setting which effectively prevents outgoing negotiation.

Minimum value < maximum value

In this case, all negotiation requests will be attempted. VAX PSI will give you the best choice of values because the requested parameter is outside the valid range; that is, greater than the *maximum-value*.

Note that for the default values, VAX PSI will not fill the parameter and facility fields in the call packet.

C.4 How to Enable Flow Control Parameter Negotiation

C.4.1 Enabling Window Size Negotiation

To enable window size negotiation for a particular DTE, change the *maximum-window* parameter for the DTE in the X25-Protocol DTE database to the required value. The maximum window size allowed by your PSDN is shown in the relevant chapter of the *Public Network Information* manual.

For example, to set the *maximum-window* size to 5 for a DTE connected to the PSS network, the command is:

```
NCP>SET MODULE X25-P DTE 234212345678 NET PSS MAX WINDOW 5
```

This command allows all incoming and outgoing calls to and from DTE 234212345678 to negotiate a window size up to 5.

C.4.2 Disabling Window Size Negotiation

To disable window size negotiation for a particular DTE, change the *maximum-window* parameter for the DTE in the X25-Protocol database to the default value. This will stop negotiation for window sizes larger than the default value.

The default window size for your PSDN is shown in the relevant chapter of the *Public Network Information* manual.

C.4.3 Enabling Packet Size Negotiation.

To enable packet size negotiation for a particular DTE, change the *maximum-data* parameter for the DTE in the X25-Protocol DTE database to the required value. The maximum data size allowed by your PSDN is shown in the relevant chapter of the *Public Network Information* manual.

For example, to set the *maximum-data* size to 256 bytes for a DTE connected to the TRANSPAC network, the command is:

```
NCP>SET MODULE X25-P DTE 208012345678 NET TRANSPAC MAX DATA 256
```

This command allows all incoming and outgoing calls to and from DTE 208012345678 to negotiate a data size up to 256 bytes.

Note also that the line associated with the DTE must be able to use large frame sizes. You may need to change the *maximum-block* parameter of the associated line in the line database, remembering to allow for the packet header size. For example, the following command:

```
NCP>SET LINE DMF-0 NET TRANSPAC MAX BLOCK 261
```

allows frames of up to 261 bytes long for the line DMF-0, that is, 256 bytes for the packet and 5 bytes for the packet header.

C.4.4 Disabling Data Size Negotiation

To disable data size negotiation for a particular DTE, change the *maximum-data* parameter for the DTE in the X25-Protocol DTE database to the default value. This will stop negotiation for data sizes larger than the default.

The default data size for your PSDN is shown in the relevant chapter of the *Public Network Information* manual.

In addition, you must ensure that the line associated with the DTE is using the correct frame size; that is, the *maximum-block* size for the line is 5 bytes more than the default packet size.

C.5 Setting Up Flow Control Parameter Negotiation for DEC Supplied Application Programs.

Flow control parameter negotiation may increase the packet size and/or the window size for DEC supplied applications for a specific SVC, thus increasing throughput for the SVC. (Note that user applications can request flow control parameter negotiation, as described in the *VAX P.S.I. X.25 Programmer's Guide.*)

The following sections give details of the DEC supplied applications which can request flow control parameter negotiation.

C.5.1 Host Based PAD

Increasing the packet size does not usually improve the performance for a host based PAD because, typically, small data quantities are transferred at a time. However, increasing the window size may have significant effect on the performance, depending on the network and the remote host. You should test the effect of an increased window size in specific contexts. For example, the following command requests a window size of 7 for a specific call using the host based PAD:

```
NCP>SET HOST/X29/WINDOW_SIZE=7 234212345678
```

For more information about this type of command, see the *VAX P.S.I. PAD and MAIL Utilities Manual.*

C.5.2 PSI Mail

Increasing the packet and window sizes usually increases the PSI Mail throughput.

See the *VAX P.S.I. PAD and MAIL Utilities Manual* for more information about the use of the logical names PSI$MAIL_WINDOW_SIZE and PSI$MAIL_PACKET_SIZE.

C.5.3 DECnet Data Link Mapping (DLM) Circuits

Increasing the packet and window sizes usually increases the DLM circuits throughput. Ideally, the DLM packet size should match the *executor-buffer-size* parameter so that one executor buffer can be sent in one packet.

You can request particular packet and window sizes associated with a DLM SVC circuit through NCP circuit commands.

For example:

```
NCP>DEF CIR X25_USER OWNER EXECUTOR USAGE OUTGOING -
_ MAX DATA 512 MAX WINDOW 7
```

This NCP command is an example of requesting a packet size of 512 and a window size of 7 for the circuit X25_USER.

C.6 The Effects of Increased Packet and Window Sizes on Usage of VMS Resources

Increasing packet and window sizes will increase usage of non-paged dynamic pool. See the *VAX P.S.I. Installation Procedures* for more information, particularly the sections about Native mode installation, Access mode installation, and Combination mode installation.

Index

A

B

C

D

E

F

G

I

K

L

M

N

O

P

R

S

VAX P.S.I.
Management Guide
AA-HS82B-TE

READER'S COMMENTS

What do you think of this manual? Your comments and suggestions will help us to improve the quality and usefulness of our publications.

Please rate this manual:

	Poor				Excellent
Accuracy	1	2	3	4	5
Readability	1	2	3	4	5
Examples	1	2	3	4	5
Organization	1	2	3	4	5
Completeness	1	2	3	4	5

Did you find errors in this manual? If so, please specify the error(s) and page number(s).

_____

_____

_____

_____

General comments:

_____

_____

_____

_____

Suggestions for improvement:

_____

_____

_____

_____

Name _____ Date _____

Title _____ Department _____

Company _____ Street _____

City _____ State/Country _____ Zip Code _____

Digital Equipment Company Limited
Wide Area Communications Environment
PO Box 121 READING
Berkshire RG2 0TU
England