

# VSI X.25 for OpenVMS Accounting Guide

**Operating System and Version:** VSI OpenVMS IA-64 Version 8.4-1H1 or higher  
VSI OpenVMS Alpha Version 8.4-2L1 or higher

**Software Version:** VSI X.25 for OpenVMS Version 2.1

---

# VSI X.25 for OpenVMS Accounting Guide



VMS Software

---

Copyright © 2025 VMS Software, Inc. (VSI), Boston, Massachusetts, USA

## Legal Notice

Confidential computer software. Valid license from VSI required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for VSI products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. VSI shall not be liable for technical or editorial errors or omissions contained herein.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group.

# Table of Contents

<b>Preface .....</b>	<b>v</b>
1. About VSI .....	v
2. Intended Audience .....	v
3. Document Structure .....	v
4. Related Documents .....	v
5. OpenVMS Documentation .....	viii
6. VSI Encourages Your Comments .....	viii
7. Terminology .....	viii
8. Conventions .....	ix
<b>Chapter 1. Introduction to X.25 Accounting .....</b>	<b>1</b>
1.1. The X.25 Accounting Utility .....	2
1.2. Resources Needed to Run the X.25 Accounting Utility .....	2
1.2.1. File Space .....	2
1.2.2. Factors Influencing Processing Time .....	2
1.3. Producing Accounting Records .....	2
1.4. Processing VAX P.S.I. V4.3 Accounting Records (OpenVMS VAX Only) .....	3
<b>Chapter 2. Using the X.25 Accounting Utility .....</b>	<b>5</b>
2.1. Starting and Exiting From the Utility .....	5
2.2. Command Qualifiers .....	5
2.3. Information Provided by the X.25 Accounting Utility .....	6
2.3.1. Information About the Process Using X.25 for OpenVMS .....	6
2.3.2. Information on How X.25 for OpenVMS Is Used .....	7
2.3.3. X.25 Accounting and X.29 Incoming Calls .....	10
2.4. Specifying the Output Format .....	10
2.4.1. Listing Output .....	11
2.4.1.1. Brief Listing Format .....	11
2.4.1.2. Full Listing Format .....	11
2.4.1.3. Summary Listing Format .....	11
2.4.1.4. Directing the ACCOUNTING/X25 Listing Output .....	12
2.4.2. Binary Output Files .....	12
2.4.2.1. Directing the ACCOUNTING/X25 Binary Output .....	12
2.5. Selecting Records .....	13
2.6. Sorting Records .....	13
2.7. Using DCL Symbols .....	14
2.8. Error Messages .....	14
2.9. Summary of the ACCOUNTING/X25 Command .....	14
2.9.1. Format .....	14
2.9.2. Summary of the ACCOUNTING/X25 Command Qualifiers .....	15
2.9.3. Restrictions .....	15
<b>Chapter 3. The X.25 Accounting Utility Command Qualifiers .....</b>	<b>19</b>
/ACCOUNT .....	19
/BEFORE .....	19
/BINARY .....	20
/BRIEF .....	20
/FULL .....	21
/IDENT .....	21
/LOG .....	22
/NODE .....	22

/OUTPUT .....	23
/OWNER .....	24
/PRIORITY .....	24
/REJECTED .....	25
/REMOTE_ID .....	25
/REPORT .....	26
/SINCE .....	27
/SORT .....	27
/SUMMARY .....	28
/TERMINAL .....	29
/TITLE .....	30
/TYPE .....	30
/UIC .....	31
/USER .....	32
<b>Appendix A. X.25 Accounting Record Formats .....</b>	<b>33</b>
A.1. Record Format .....	33
A.2. Accounting Record Types .....	35
A.3. Accounting Packets .....	35
A.3.1. General Format of Information Packets .....	35
A.3.2. File Name Packet (ACR\$K_FILENAME) .....	37
A.3.3. Identification Packet (ACR\$K_ID) .....	37
A.3.4. X.25 Packet (ACR\$K_PSI) .....	39
<b>Appendix B. A Sample X.25 Accounting Program .....</b>	<b>43</b>

# Preface

## 1. About VSI

VMS Software, Inc. (VSI) is an independent software company licensed by Hewlett Packard Enterprise to develop and support the OpenVMS operating system.

## 2. Intended Audience

This manual is intended for anyone who has the task of managing X.25 for OpenVMS to monitor its usage and performance.

The manual assumes that you understand and have experience with the following:

- Data communications
- Packet switching
- Local Area Networks
- Wide Area Networks
- OpenVMS operating system
- DECnet-Plus
- X.25 for OpenVMS functionality

In addition, the manual assumes that you have read the *VSI DECnet-Plus for OpenVMS Introduction and User's Guide* [<https://docs.vmssoftware.com/vsi-openvms-decnet-plus-introduction-and-user-s-guide/>].

## 3. Document Structure

This manual is divided into three chapters and two appendices:

- Chapter 1, *Introduction to X.25 Accounting* provides an overview of the X.25 Accounting utility.
- Chapter 2, *Using the X.25 Accounting Utility* describes how to use the X.25 Accounting utility.
- Chapter 3, *The X.25 Accounting Utility Command Qualifiers* details each of the qualifiers that can be used with the X.25 Accounting utility.
- Appendix A, *X.25 Accounting Record Formats* details the structure of the data records written to the X.25 Accounting log file.
- Appendix B, *A Sample X.25 Accounting Program* provides a brief description of the sample user accounting program installed in the `SYS$EXAMPLES` directory.

## 4. Related Documents

The following sections describe VSI DECnet-Plus for OpenVMS, VSI X.25 for OpenVMS, and VSI OpenVMS manuals that either directly describe the X.25 for OpenVMS software or provide related information.

## VSI DECnet-Plus for OpenVMS Documentation

The following DECnet-Plus manuals contain information useful to X.25 for OpenVMS managers, users, and programmers:

- *VSI OpenVMS DECnet-Plus Introduction and User's Guide* [<https://docs.vmssoftware.com/vsi-openvms-decnet-plus-introduction-and-user-s-guide>]

This manual provides general information on DECnet-Plus and describes the concept of packet switching data networks.

- *VSI OpenVMS DECnet-Plus Installation and Configuration* [<https://docs.vmssoftware.com/vsi-openvms-decnet-plus-installation-and-configuration>]

This manual describes how to install and configure VSI DECnet-Plus for OpenVMS software. For OpenVMS IA-64 and OpenVMS Alpha systems, this manual also describes how to install X.25 for OpenVMS software. Details on configuring X.25 for OpenVMS on OpenVMS IA-64 and OpenVMS Alpha systems are provided in the *VSI X.25 for OpenVMS Configuration Guide* [<https://docs.vmssoftware.com/vsi-x-25-for-openvms-configuration-guide>]. For OpenVMS VAX systems, this manual also describes how to install and configure the X.25 functionality provided by VSI DECnet-Plus for OpenVMS VAX.

- *VSI DECnet-Plus Network Management Guide* [<https://docs.vmssoftware.com/vsi-decnet-plus-network-management-guide>]

This manual provides conceptual and task information about managing and monitoring a DECnet-Plus network. In addition, the manual devotes a section to the management of X.25 entities used by DECnet operating over X.25 data links.

- *VSI OpenVMS DECnet-Plus Network Control Language Reference* [<https://docs.vmssoftware.com/vsi-openvms-decnet-plus-network-control-language-reference>]

This manual provides detailed information on the Network Control Language (NCL), which is used to manage X.25 for OpenVMS management entities.

## VSI X.25 for OpenVMS Documentation

The following manuals make up the X.25 for OpenVMS documentation set:

- *VSI X.25 for OpenVMS Configuration Guide* [<https://docs.vmssoftware.com/vsi-x-25-for-openvms-configuration-guide>] (OpenVMS IA-64 and OpenVMS Alpha)

This manual explains how to configure X.25 for OpenVMS software on OpenVMS IA-64 and OpenVMS Alpha systems.

- *VSI X.25 for OpenVMS Security Guide*

This manual describes the X.25 Security model and how to set up, manage, and monitor X.25 Security to protect your X.25 for OpenVMS system from unauthorized incoming and outgoing calls.

- *VSI X.25 for OpenVMS Problem Solving Guide* [<https://docs.vmssoftware.com/vsi-x-25-for-openvms-problem-solving-guide>]

This manual provides guidance on how to analyze and correct X.25-related and X.29-related problems that may occur while using the X.25 for OpenVMS software. In addition, the manual describes loopback testing for LAPB data links.

- *VSI X.25 for OpenVMS Programming Guide*

This manual describes how to write X.25 and X.29 programs to perform network operations.

- *VSI X.25 for OpenVMS Programming Reference*

This manual provides reference information for X.25 and X.29 programmers. It is a companion manual to the VSI X.25 for OpenVMS Programming.

- *VSI X.25 for OpenVMS Accounting*

This manual describes how to use X.25 Accounting to obtain performance records and information on how X.25 is being used on your system.

- *VSI X.25 for OpenVMS Installation Guide* [<https://docs.vmssoftware.com/vsi-x-25-for-openvms-installation-guide/>]

This manual describes how to install VSI X.25 for OpenVMS V2.1 HPE servers running the OpenVMS operating system. This guide is intended for system managers who are responsible for installing VSI X.25 for OpenVMS V2.1.

- *VSI X.25 for OpenVMS Management Guide* [<https://docs.vmssoftware.com/vsi-x-25-management-guide/>]

This manual provides information applicable to the X.25 functionality provided by VSI X.25 for OpenVMS and VSI DECnet-Plus for OpenVMS VAX.

- *VSI X.25 for OpenVMS Utilities Guide*

This manual describes how to use and manage X.25 Mail and how to use and manage a host-based PAD to connect to a remote system. It also describes how to manage the X.29 communication links used for both of these functions. In addition, this manual explains how to use OpenVMS DCL **SET TERMINAL/X29** commands to manage remote host-based or network PADs.

## VSI OpenVMS Documentation

The following OpenVMS manuals contain information useful to X.25 for OpenVMS managers, users, and programmers:

- *VSI OpenVMS User's Manual* [<https://docs.vmssoftware.com/vsi-openvms-user-s-manual/>].
- *VSI OpenVMS DCL Dictionary Part One* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m>] and *Part Two* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-n-z>].
- *VSI OpenVMS System Management Utilities Reference Manual, Volume I* [<https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-1>] and *Volume II* [<https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-ii-m-z>].
- *HP OpenVMS System Services Reference Manual Part One* [<https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-a-getuai/>] and *Part Two* [<https://docs.vmssoftware.com/vsi-openvms-system-services-reference-manual-getutc-z/>].
- *VSI OpenVMS Guide to System Security* [<https://docs.vmssoftware.com/vsi-openvms-guide-to-system-security>].

## 5. OpenVMS Documentation

The full VSI OpenVMS documentation set can be found on the VMS Software Documentation webpage at <https://docs.vmssoftware.com>.

## 6. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: <docinfo@vmssoftware.com>. Users who have VSI OpenVMS support contracts through VSI can contact <support@vmssoftware.com> for help with this product.

## 7. Terminology

The terminology used in the VAX P.S.I. product has been replaced by the terminology used in the X.25 for OpenVMS product. Table 1, “X.25 Terminology” shows the correlation between VAX P.S.I. terms and their X.25 for OpenVMS counterparts.

**Table 1. X.25 Terminology**

VAX P.S.I.	X.25 for OpenVMS
VAX P.S.I.	X.25 for OpenVMS VAX
Access system	X.25 Client system
Native system	X.25 Direct Connect system
Multihost system	X.25 Connector system
Gateway system	X.25 Connector system

In addition to the terms shown in the previous table, the X.25 for OpenVMS documentation set uses the following standard terms for client systems, server systems, relay systems, and the X.25 for OpenVMS management entities that represent these systems:

**Table 2. X.25 for OpenVMS Client/Server Terminology**

Client system	A client system of an X.25 Connector system (and therefore a client of the X25 Server management module on the X.25 Connector system.)
Relay Client system	A client system of an X.25 Relay system (and therefore a client of the X25 Relay management module on the X.25 Relay system.)
Relay-Client	A shorthand term for an X25 RELAY CLIENT management entity on an X.25 Relay system that contains management information about an actual Relay Client system.
Relay system	An X.25 Direct Connect or Connector system with the X.25 Relay module enabled.
Server Client system	Another term for a Client system.
Server-Client	A shorthand term for an X25 SERVER CLIENT management entity on an X.25 Connector system that contains management information about one or more actual X.25 Client systems.



For more information about clients, servers, and relays in X.25 for OpenVMS, refer to the *VSI X.25 for OpenVMS Configuration Guide* [<https://docs.vmssoftware.com/vsi-x-25-for-openvms-configuration-guide>] and the *VSI X.25 for OpenVMS Management Guide* [<https://docs.vmssoftware.com/vsi-x-25-management-guide>].

## 8. Conventions

The following conventions are used in the X.25 for OpenVMS documentation set:

Convention	Meaning
UPPERCASE and lowercase	<p>The OpenVMS operating system does not differentiate between lowercase and uppercase characters. Literal strings that appear in text, examples, syntax descriptions, and function descriptions can be entered using uppercase characters, lowercase characters, or a combination of both.</p> <p>In running text, uppercase characters indicate OpenVMS DCL commands and command qualifiers; Network Control Language (NCL) commands and command parameters; other product-specific commands and command parameters; network management entities; OpenVMS system logical names; and OpenVMS system service calls, parameters, and item codes.</p> <p>Leading uppercase characters, such as Protocol State, indicate management entity characteristics and management entity event names. Leading uppercase characters are also used for the top-level management entities known as modules.</p>
<code>system output</code>	This typeface is used in interactive and code examples to indicate system output. In running text, this typeface is used to indicate the exact name of a device, directory, or file; the name of an instance of a network management entity; or an example value assigned to a DCL qualifier or NCL command parameter.
<b>user input</b>	In interactive examples, user input is shown in <b>bold monospaced</b> print.
\$	In this manual, a dollar sign (\$) is used to represent the default OpenVMS user prompt.
Ctrl/x	In procedures, a sequence such as Ctrl/x indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
<b>Return</b>	In procedures, a key name is shown enclosed to indicate that you press the corresponding key on the keyboard.
<i>italic text</i>	Italic text indicates variables or book names. Variables include information that varies in system input and output. In discussions of event messages, italic text indicates a possible value of an event argument.
<b>bold text</b>	Bold text indicates an important term or important information.
( )	In a command definition, parenthesis indicate that you must enclose the options in parenthesis if you choose more than one. Separate the options using commas.

<b>Convention</b>	<b>Meaning</b>
{ }	In a command definition, braces are used to enclose sets of values. The braces are a required part of the command syntax.
[ ]	In a command definition, square brackets are used to enclose parts of the command that are optional. You can choose one, none, or all of the options. The brackets are not part of the command syntax. However, brackets are a required syntax element when specifying a directory name in an OpenVMS file specification.

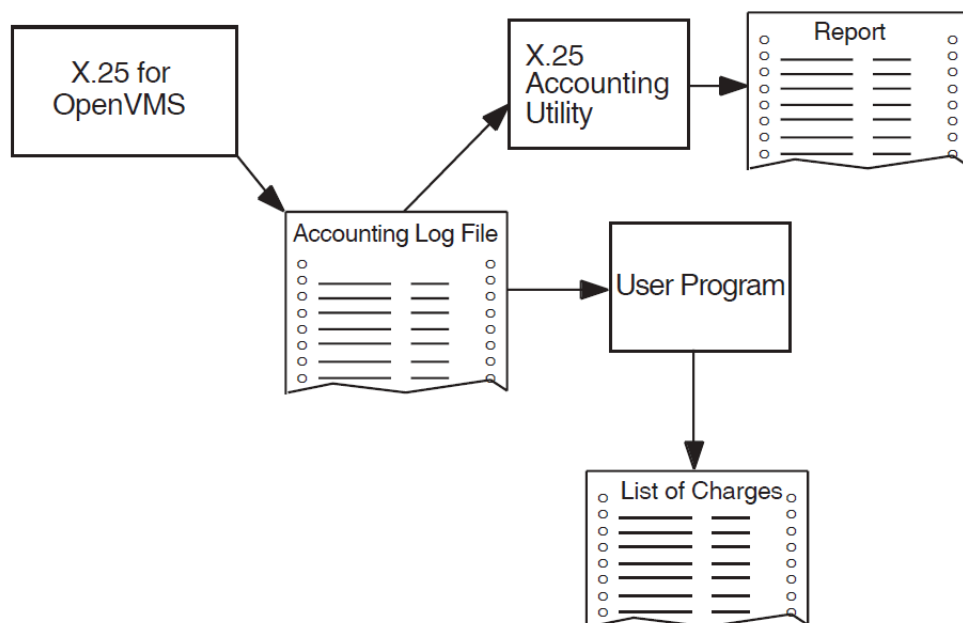
# Chapter 1. Introduction to X.25 Accounting

When you use VSI X.25 for OpenVMS, you can record details of the way in which it is used. You can use these details to charge users for X.25 resources or to record system performance. 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).
- All calls and attempted calls (including failed outgoing access) and all access to Permanent Virtual Circuits (PVCs), on Direct Connect, Connector, and Client systems.

Note that incoming X.29 calls do not show information about the user who is logged in, because this information is not available. When you have produced a set of accounting records, you can run the X.25 Accounting utility to produce a report. Alternatively, you can process the records with your own program. Figure 1.1, “Accounting Options” illustrates the options that the X.25 Accounting utility gives you.

**Figure 1.1. Accounting Options**



Details of generating the accounting records are given in Section 1.3, “Producing Accounting Records”. The accounting record formats are described in Appendix A, *X.25 Accounting Record Formats*.

An example user accounting program is provided in the `SYS$EXAMPLES:` directory. The program can be used without modification or can be adapted to suit your own requirements. See Appendix B, *A Sample X.25 Accounting Program* for a brief description of the example.

## 1.1. The X.25 Accounting Utility

The X.25 Accounting utility was derived from the OpenVMS Accounting utility (ACCOUNTING) to provide a means of processing X.25 records and packets. All OpenVMS Accounting command qualifiers are therefore valid qualifiers of the X.25 Accounting utility. However, since X.25 does not use all of the fields available with OpenVMS, some of the qualifiers will produce no output (refer to Chapter 3, *The X.25 Accounting Utility Command Qualifiers*).

The X.25 Accounting utility produces four types of output:

- A brief listing of selected records
- A full listing of selected records
- A summary report of selected items from selected records
- A binary copy of selected and/or rejected records These can be displayed on a terminal or written to a file.

X.25 Accounting 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 X.25 Accounting requires read access to the input accounting file.

## 1.2. Resources Needed to Run the X.25 Accounting Utility

This section describes the system resources required to run the X.25 Accounting utility.

### 1.2.1. File Space

When output from the X.25 Accounting utility is directed to a file, large quantities of disk space can be consumed in a short time. If your disk quota is exceeded during execution of an X.25 accounting 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 you require.

### 1.2.2. 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.

## 1.3. Producing Accounting Records

To generate accounting records, the following command procedure must be run:

```
SYS$MANAGER:X25$ACCOUNTING.COM      (OpenVMS I64 and OpenVMS Alpha)
SYS$MANAGER:PSIACCOUNTING.COM       (OpenVMS VAX)
```

This command procedure also lets you specify where the records are written.

To use the command procedure, you need the SYSPRV privilege.

To run the command procedure:

1. Enable your terminal as follows:

```
$ 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.

2. Enter the command:

```
$ @SYS$MANAGER:X25$ACCOUNTING parameter (OpenVMS I64 and OpenVMS  
Alpha)  
$ @SYS$MANAGER:PSIACCOUNTING parameter (OpenVMS VAX)
```

where *parameter* is one of the commands **ON**, **OFF**, **OPEN**, **CLOSE**, or **SHOW**. The commands have the following meanings:

#### **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.)

#### **OFF**

Stop accounting record production and close the output file.

#### **OPEN *file-name***

Close any current file, then start producing accounting records and write them to the file *file-name*.

#### **CLOSE**

Stop accounting record production and close the output file. This is the same as OFF.

#### **SHOW**

Display an OPCOM message giving the current state of X.25 Accounting and the current output file name.

---

### **Note**

- The **ON** and **OPEN** commands 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.
- For a call to be written to the accounting file, X.25 Accounting must be ON when the virtual circuit is cleared. If X.25 Accounting is OFF when the virtual circuit is cleared, any calls in progress will not be logged.

---

## **1.4. Processing VAX P.S.I. V4.3 Accounting Records (OpenVMS VAX Only)**

Accounting records produced by X.25 for OpenVMS VAX have a different format to those produced by VAX P.S.I. V4.3. However, you can process accounting records produced by VAX P.S.I. V4.3 using X.25 for OpenVMS VAX.

By default, X.25 for OpenVMS VAX does not process accounting records produced by VAX P.S.I. V4.3. To process such records, redefine the logical name PSIACC as follows:

```
DEFINE PSIACC SYS$SYSTEM:PSIACC_V43.EXE
```

To return to processing accounting records produced by X.25 for OpenVMS VAX, redefine the logical name PSIACC as follows:

```
DEFINE PSIACC SYS$SYSTEM:PSIACC.EXE
```

# Chapter 2. Using the X.25 Accounting Utility

This chapter provides an overview of the X.25 Accounting utility provided with VSI X.25 for OpenVMS.

## 2.1. Starting and Exiting From the Utility

To invoke the X.25 Accounting utility, enter the following DCL command:

```
$ ACCOUNTING/X25 [file-spec[, . . . ]] (OpenVMS I64 and OpenVMS Alpha)
$ ACCOUNTING/PSI [file-spec[, . . . ]] (OpenVMS VAX)
```

where *file-spec[, . . . ]* specifies one or more accounting files to be processed.

---

### Note

For compatibility with X.25 for OpenVMS VAX, the following format is also supported on OpenVMS IA-64 and OpenVMS Alpha systems:

```
ACCOUNTING/PSI [file-spec[, . . . ]]
```

---

Each time you issue the command it executes a single accounting request. Generally, each request runs until it is complete.

If you want to terminate the command before it is complete, press **Ctrl/Y** and type EXIT. These actions ensure that the command terminates normally, and all the files are handled correctly.

Any OpenVMS user with sufficient privilege to access the accounting file may use the X.25 Accounting utility.

---

### Note

Throughout the rest of this document the command **ACCOUNTING/X25** is used in example commands. If using the X.25 Accounting utility on an OpenVMS VAX system, you should use **ACCOUNTING/PSI** in place of **ACCOUNTING/X25** when entering a command.

---

## 2.2. Command Qualifiers

When you issue the **ACCOUNTING/X25** command, you can add qualifiers to:

- Select records from the accounting log file.
- Specify whether the output goes to the SYS\$OUTPUT device or to a separate output file.
- Specify whether the output will be in binary or ASCII format.

Section 2.4, “Specifying the Output Format”, Section 2.5, “Selecting Records”, and Section 2.6, “Sorting Records” describe these qualifiers. Full details about each of the available command qualifiers are provided in Chapter 3, *The X.25 Accounting Utility Command Qualifiers*.

## 2.3. Information Provided by the X.25 Accounting Utility

This section briefly describes the information provided by the X.25 Accounting utility. All the fields described in the following sections can be displayed using the Full Listing facility (**/FULL** qualifier). The fields are divided into two groups:

- Fields that provide information about the process that is using X.25 for OpenVMS (refer to Section 2.3.1, “Information About the Process Using X.25 for OpenVMS”).
- Fields that provide information about how X.25 for OpenVMS was used (refer to Section 2.3.2, “Information on How X.25 for OpenVMS Is Used”).

---

### Note

For calls handled internally (for example, incoming X.29 calls), the information about the process using X.25 for OpenVMS refers to the SYSTEM, and therefore the information in the fields described in Section 2.3.1, “Information About the Process Using X.25 for OpenVMS” is not useful.

For details of how to use the X.25 Accounting utility to obtain information about the process that is making X.29 calls, refer to Section 2.3.3, “X.25 Accounting and X.29 Incoming Calls”.

---

### 2.3.1. Information About the Process Using X.25 for OpenVMS

The X.25 Accounting utility provides the following information about the process using X.25 for OpenVMS:

Account	A field defined by the system manager, to accumulate data on the use of X.25 for OpenVMS resources.
Elapsed time	The difference between the Start time and the Finish time (if the connection was successful).
Final status code	The code associated with the final (exit) completion status.
Final status text	The final (exit) completion status of the process or image.
Finish time	The time at which the virtual circuit was terminated.
Job name	The name assigned to the entry in the batch queue. Applicable only to printing.
Owner ID	The process name.
Priority	The priority of the process (at the time the call was initiated).
Privilege	The privileges of the process.
Process ID	The unique OpenVMS identifier of a process.
Processor time	The processor time used during the current user session.
Queue entry	The name assigned to the entry in the print queue.
Queue name	The name of the print queue.
Start time	The time at which the virtual circuit was established. This field contains a value only if the connection was successful.
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 (UIC) of the process.
Username	The name that the user typed to log on to the system.

## 2.3.2. Information on How X.25 for OpenVMS Is Used

The X.25 Accounting utility provides the following information about how X.25 for OpenVMS is being used:

### Accept facilities

A copy of the facilities field in the call accept packet (refer to your PSDN documentation). The field is displayed in hexadecimal notation.

### Bytes sent/received

The counts of the data bytes sent/received. These are used for charging purposes.

### Association

For an incoming SVC, this is the name of the X25 ACCESS FILTER entity that matched the call.

For an outgoing call, it is the name of the X25 ACCESS TEMPLATE entity used in making the call.

For a PVC, this field is blank.

### Calling facilities

A copy of the facilities field in the calling packet (refer to your PSDN documentation). The field is displayed in hexadecimal notation.

### Circuit type

The type of virtual circuit used. The types are as follows:

- Outgoing or incoming
- Switched Virtual circuit (SVC) or Permanent Virtual Circuit (PVC)
- Unsuccessful
- X.25 Client (PSI-access) or Connector (Multi-host) system
- X.29 (incoming)

### Clearing cause

Information from the clear packet sent by the PSDN about the virtual circuit clearance (see your PSDN documentation).

### Clearing facilities

A copy of the facilities field in the clear packet (refer to your PSDN documentation). This field is displayed in hexadecimal notation. For some PSDNs, it may contain all the charging information you require.

### Clearing reason

An explanatory text about virtual circuit clearance.

### **CUG number**

This is the number for a Closed User Group (CUG) used. This number is assigned by the PSDN when you subscribe to the CUG.

### **DTE Class**

The X25 ACCESS DTE CLASS entity used for the SVC or PVC.

### **Diagnostic**

Diagnostic information from the clear packet sent by the PSDN (see your PSDN documentation).

### **Facilities**

The facilities requested for the virtual circuit. These are as follows:

- Reverse charging
- Fast select
- CUG
- Bilateral CUG (group of two DTEs)

### **Group**

Name of the Closed User Group (CUG) used.

### **Inc[oming] and out[going] packet size**

The incoming and outgoing packet size requested from the PSDN. This may affect the charging rate.

### **Inc[oming] and out[going] throughput class**

The incoming and outgoing throughput classes requested from the PSDN. This may affect the charging rate.

### **Inc[oming] and out[going] window size**

The incoming and outgoing window size requested from the PSDN. This may affect the charging rate.

### **Local DTE**

For outgoing SVCs, this is the address of the DTE on which the call was made, and for incoming SVCs it is the address of the DTE on which the call arrived.

For PVCs, the field is blank.

### **Local NSAP**

If the NSAP is specified, this is:

- For an outgoing SVC, the NSAP address of the caller.
- For an incoming SVC, the NSAP address being called.

### **LCN**

The Logical Channel Number (LCN) associated with the virtual circuit.

**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.

**Network device**

The NW or NV device unit reference number or name.

**Original DTE**

For a redirected SVC, this is the address of the DTE called by the original call packet.

**Packets sent/received**

The counts of the data packets sent/received.

**Protocol ID**

The first four bytes of call data.

**PVC Name**

The name of the PVC used for the virtual circuit.

**Redirect reason**

For an incoming SVC, this indicates whether the call was redirected and, if it was, provides the reason as follows:

- Busy
- Out of order
- Systematic redirection

**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 by the PSDN.

For Permanent Virtual Circuits (PVCs), the field is blank.

**Remote ID**

The identifier of the user at the remote node. (For a Client system, this will always be the Connector node.)

**Remote node name**

The name of the remote node involved in using a Client or Connector system.

**Remote NSAP**

If the NSAP is specified, this is:

- For an outgoing SVC, the NSAP address being called.
- For an incoming SVC, the NSAP address of the caller.

### Segments sent/received

The counts of the data segments sent/received. A segment is a charging unit, and 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.

### Source

The name of the X25 ACCESS PORT entity associated with the virtual circuit at the source node.

## 2.3.3. X.25 Accounting and X.29 Incoming Calls

For X.29 incoming access, the X.25 Accounting utility gives `SYSTEM` as the user name because it is an X.25 housekeeping process that accepts the call. This process runs under the account name of the process that started X.25 for OpenVMS. This is usually `SYSTEM`.

You can correlate the X.25 for OpenVMS and OpenVMS accounting records to find out the user name of the X.29 incoming call. Note, however, that if virtual terminals have been enabled, it is only possible to correlate the records using the time and process identification (PID).

Enter the following command to display the name of the NV terminal used in the call:

```
$ ACCOUNTING/X25/FULL/USER=SYSTEM
```

You can then match this information with the entry in OpenVMS accounting. For example, if the above command displayed `NVA6` as the terminal used in the call, enter the following command to find the user name 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 user name before being cleared.

## 2.4. Specifying the Output Format

The following options are available for the output from the **ACCOUNTING/X25** command:

- Produce Listing (ASCII) format on your `SYS$OUTPUT` device. If the display device is a terminal, it should be defined as a VT-100 or later device.

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 troubleshooting exercise. No historical record of output is kept.

- Direct Listing (ASCII) format to an output file. Section 2.4.1.4, “Directing the ACCOUNTING/X25 Listing Output” describes how to direct the output to an alternative terminal or to an output file.
- Direct binary format to a file that you can process later using other accounting commands. Section 2.4.2.1, “Directing the ACCOUNTING/X25 Binary Output” describes how to use qualifiers to the **ACCOUNTING/X25** command to specify binary format and direct it to an output file.

Section 2.4.1, “Listing Output” describes how to use qualifiers to the **ACCOUNTING/X25** command to specify the kind of listing format you require. If you do not use any qualifiers to specify the output

format you require, the default is the brief listing format described in Section 2.4.1.1, “Brief Listing Format”.

## 2.4.1. Listing Output

Listing output consists of screen images. Any terminal supported by OpenVMS 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 correct dimensions.)

Alternatively, listing output can be routed to a file for subsequent printing. Section 2.4.1.4, “Directing the ACCOUNTING/X25 Listing Output” describes how to do this.

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.

### 2.4.1.1. 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.

For example, to obtain a brief listing of the information in the default accounting file, enter:

```
$ ACCOUNTING/X25
```

Since no command qualifiers have been named, the X.25 Accounting utility applies the following defaults to the **ACCOUNTING/X25** command:

- **/NOFULL** = a brief listing
- *input file* = SYS\$MANAGER:PSIACCOUNTING.DAT
- **/OUTPUT** = current SYS\$OUTPUT device

The following command provides a brief listing from the binary input file MYFILE.

```
$ ACCOUNTING/X25 MYFILE
```

### 2.4.1.2. Full Listing Format

The full listing format provides all the data for all selected records in the accounting file being processed.

For example, to obtain a full listing of all the records in the default accounting file, enter:

```
$ ACCOUNTING/X25/FULL
```

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. The default response for this prompt is "No", in which case the first screen image of the next record is displayed. This does not apply if the qualifier **/OUTPUT=file** is used.

### 2.4.1.3. 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 period requested. The statistics in the summary output are either totals or maximum values of the report items (refer to the **/REPORT** qualifier for details).

To generate the summary output, use the **/SUMMARY** qualifier. For example, to obtain a summary of the accounting records for the current day, enter:

```
$ ACCOUNTING/X25/SUMMARY=DAY /REPORT
```

Note that in the above example, the default value for the **/REPORT** qualifier is used, which totals the records in the file.

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.

A list of the fields that can be specified with the **/SUMMARY** qualifier is given in Chapter 3, *The X.25 Accounting Utility Command Qualifiers*.

#### 2.4.1.4. Directing the ACCOUNTING/X25 Listing Output

The output from the X.25 Accounting utility can be routed to any supported terminal device (VT-100 or later) or to a file. The following command sends an ASCII listing of all records in the accounting file to the file `ACCOPY.LIS`. This file can then be printed out on a hardcopy device.

```
$ ACCOUNTING/X25/OUTPUT=ACCOPY
```

### 2.4.2. Binary Output Files

A binary output file is itself an accounting log file that is created when an accounting 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/X25** command with different selection criteria. These files provide you with a set of preselected 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 matching the selection criteria are directed to this file.

A complete description of the record formats of the log file is given in Appendix A, *X.25 Accounting Record Formats*.

By default, output is directed to the `SYS$OUTPUT` device. However, it is more useful to specify a separate output file. Section 2.4.2.1, “Directing the ACCOUNTING/X25 Binary Output” describes how to do this.

#### 2.4.2.1. Directing the ACCOUNTING/X25 Binary Output

Whenever you need to capture X.25 accounting data for future use, you can direct the binary output to a file. For example, you might want to gather routine performance data for long-term analysis. X.25 accounting data can be recorded on a routine basis and summarized to gather data about X.25 for OpenVMS resource utilization over long periods of time.

To direct the output to a file, use the qualifier **/OUTPUT**, as follows:

```
$ ACCOUNTING/X25/BINARY/OUTPUT=RECORDS.DAT
```

This example directs binary accounting records to the file `RECORDS.DAT`.

## 2.5. Selecting Records

The qualifiers of the **ACCOUNTING/X25** command allow you to select records from the accounting log file. You select records based on fields in the records and their values. When you select records, you have the option of storing the records you did not select in a binary output file, which you can process later. To store the records you did not select, use the **/REJECTED** qualifier. If you do not use this qualifier, you will lose the records you did not select.

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 the use of these selection qualifiers.

```
$ ACCOUNTING/X25/SINCE=15-DEC-2003 -  
_ $ /BEFORE=22-JAN-2004:23:59:99/ACCOUNT=MANUFA/NODE=OSCAR
```

This command selects and lists in brief format only those records for the account `MANUFA` on node `OSCAR` for the specified days.

The following command rejects all accounting records that do not specify user `RAMI` in the User field. The rejected records are stored in binary format in the file `REJ_RECORDS.DAT` in the current directory.

```
$ ACCOUNTING/X25/USER=RAMI/REJECTED=REJ_RECORDS.DAT
```

For full details of the selection qualifiers, refer to Chapter 3, *The X.25 Accounting Utility Command Qualifiers*.

## 2.6. Sorting Records

The X.25 Accounting 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/X25/SORT=(ACCOUNT,USER,-PRIORITY)
```

This command sorts all records in the current accounting file by:

- Account in ascending order
- User name in ascending order
- Final priority in descending order

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/X25/ACCOUNT=MAIL/SORT=USER
```

This command selects all records for the account `MAIL` and then sorts them by user name, producing a brief listing.

A list of the fields that can be specified with the **/SORT** qualifier is given in Chapter 3, *The X.25 Accounting Utility Command Qualifiers*.

## 2.7. Using DCL Symbols

It may be convenient to establish DCL symbols for frequently used combinations, as in the following example:

```
$ MY_GROUP == "/USER=(MARY, TOM, DICK, HARRY, BARNEY, ALICE) "  
$ ACCOUNTING/X25 'MY_GROUP'
```

This example shows how you can define the DCL symbol `MY_GROUP` and use it as a parameter to the **ACCOUNTING/X25** command.

## 2.8. Error Messages

The *HP OpenVMS System Messages and Recovery Procedures Reference Manual* lists the messages generated by the OpenVMS Accounting Utility (**ACCOUNTING**) and provides explanations and suggested user actions. The error messages associated with a particular **ACCOUNTING** qualifier also apply when **ACCOUNTING/X25** uses the same qualifier.

## 2.9. Summary of the ACCOUNTING/X25 Command

This section contains a summary of the **ACCOUNTING/X25** command format and its command qualifiers. A complete description of all the command qualifiers can be found in Chapter 3, *The X.25 Accounting Utility Command Qualifiers*.

### 2.9.1. Format

The format of the DCL command to invoke the X.25 Accounting utility is:

```
$ ACCOUNTING/X25 [file-spec[, . . . ]] (OpenVMS I64 and OpenVMS Alpha)  
$ ACCOUNTING/PSI [file-spec[, . . . ]] (OpenVMS VAX)
```

where *file-spec[, . . . ]* specifies one or more accounting files to be processed.



If you specify more than one file name, separate one file name from the next with a comma. 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.

## Note

For compatibility with X.25 for OpenVMS VAX systems, the following format is also supported on OpenVMS IA-64 and OpenVMS Alpha systems:

```
ACCOUNTING/PSI [file-spec[, . . . ]]
```

## 2.9.2. Summary of the ACCOUNTING/X25 Command Qualifiers

Summaries of the qualifiers for the **ACCOUNTING/X25** command are provided in the following tables:

- Table 2.1, “Selection Qualifiers for the ACCOUNTING/X25 Command” describes the selection qualifiers for the **ACCOUNTING/X25** command.
- Table 2.2, “Other Qualifiers for the ACCOUNTING/X25 Command” lists the other qualifiers for the **ACCOUNTING/X25** command.
- Table 2.3, “OpenVMS ACCOUNTING Command Qualifiers” describes qualifiers for the OpenVMS **ACCOUNTING** command that are allowed with the **ACCOUNTING/X25** command. However, since they are selection qualifiers and X.25 records do not use these fields, they have no effect if they are entered.

## 2.9.3. Restrictions

Some of the qualifiers detailed in Table 2.1, “Selection Qualifiers for the ACCOUNTING/X25 Command” and Table 2.2, “Other Qualifiers for the ACCOUNTING/X25 Command” are subject to restrictions. These restrictions are as follows:

- The **/BRIEF**, **/FULL**, **/SUMMARY** and **/BINARY** qualifiers are mutually exclusive.
- The **/REPORT** qualifier requires the **/SUMMARY** qualifier.
- The **/SORT** qualifier may only be used for brief or full listings, not for summary reports.
- 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.

**Table 2.1. Selection Qualifiers for the ACCOUNTING/X25 Command**

Qualifiers	Defaults
<code>/[NO]ACCOUNT= { ["-",]account-name ["-",]account-name, . . . }</code>	<code>/NOACCOUNT</code>
<code>/[NO]ADDRESS= { ["-",]node-address ["-",]node-address, . . . }</code>	<code>/NOADDRESS</code>

Qualifiers	Defaults
/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, . . . }	/NOPRIORITY
/[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

Table 2.2. Other Qualifiers for the ACCOUNTING/X25 Command

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= <i>title</i>	/NOTITLE

Table 2.3. OpenVMS ACCOUNTING Command Qualifiers

Qualifiers	Defaults
The following OpenVMS <b>ACCOUNTING</b> command qualifiers are also allowed with the <b>ACCOUNTING/X25</b> command. However, since they are selection qualifiers and X.25 accounting	

Qualifiers	Defaults
records do not use the specified fields, specifying any value besides the [NO] option results in a blank display (that is, no records selected).	
/[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



# Chapter 3. The X.25 Accounting Utility Command Qualifiers

This chapter describes all the X.25 Accounting utility command qualifiers and provides examples of their use. The qualifiers are described in alphabetical order and the conventions described in the introductory text of this manual are observed.

The DCL rules for entering commands, described in the *VSI OpenVMS DCL Dictionary Part One* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/>] and *Part Two* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-n-z/>], also apply to the X.25 Accounting utility command qualifiers.

## /ACCOUNT

**/ACCOUNT** — Controls whether only those records matching the specified account names are selected. The account names specified must match the account name as specified in the user authorization file.

### Format

```
/ACCOUNT= ( [ "-" , ] account-name , . . . )
```

```
/NOACCOUNT
```

"-" specifies that all records are selected **except** those matching the specified account names.

*account-name* specifies the account names 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 one account name from the next with a comma and enclose the list of account names in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the account names specified in the list.

If you omit the qualifier or specify **/NOACCOUNT**, the account name is not used to select records.

### Examples

```
1. $ ACCOUNTING/X25/ACCOUNT= (ROBERT, RUTH)
```

The command in this example selects records matching the accounts ROBERT and RUTH.

```
2. $ ACCOUNTING/X25/ACCOUNT= ("-", ROBERT)
```

The command in this example selects records for all accounts except ROBERT.

## /BEFORE

**/BEFORE** — Controls whether only those records dated earlier than the specified time are selected. (The date of the records is the time that the virtual circuit was closed or terminated.)

## Format

**/BEFORE [= time]**

*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 *VSI OpenVMS DCL Dictionary Part One* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/>] and *Part Two* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-n-z/>].

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/X25/BEFORE=5-NOV-2004
```

The command in this example selects all records dated earlier than November 5, 2004.

## /BINARY

**/BINARY** — Controls whether output is formatted in binary or in ASCII format.

## Format

**/BINARY**

**/NOBINARY**

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/X25/OUTPUT=MYACC.DAT/BINARY/USER=JONES
```

The command in this example writes accounting data for user JONES in binary format to the file MYACC.DAT.

## /BRIEF

**/BRIEF** — Controls whether a brief format is used in ASCII displays.

## Format

**/BRIEF**

By default, records are displayed in the brief format. You must specify **/FULL** to have the complete (full) contents of each selected record displayed.

The **/BRIEF**, **/FULL**, **/SUMMARY**, and **/BINARY** qualifiers are mutually exclusive.

## Example

```
$ ACCOUNTING/X25/BRIEF
```

The command in this example displays a brief listing of the contents of each selected record.

Date / Time	Type	Subtype	Username	ID	Source	Status
25-JAN-2005 16:54:41	FILE			00000000		00000000
25-JAN-2005 16:55:31	PSI	CIRCUIT	SYSTEM	00000113		00000000
25-JAN-2005 17:09:57	PSI	CIRCUIT	SYSTEM	00000113		00000000
25-JAN-2005 17:11:41	PSI	CIRCUIT	SYSTEM	00000113		00000000
25-JAN-2005 17:11:41	PSI	CIRCUIT	SYSTEM	0000015E	RTA2:	00000000
25-JAN-2005 18:05:34	PSI	CIRCUIT	SYSTEM	0000015E	RTA2:	00000000
25-JAN-2005 18:07:07	PSI	CIRCUIT	SYSTEM	0000015E	RTA2:	00000000
25-JAN-2005 19:16:59	PSI	CIRCUIT	SYSTEM	00000162	NVA19:	00000000
25-JAN-2005 19:17:05	PSI	CIRCUIT	SYSTEM	00000113		00000000

## /FULL

**/FULL** — Controls whether a full format is used in ASCII displays.

## Format

```
/FULL
```

```
/NOFULL
```

By default, records are displayed in the brief format. You must specify **/FULL** to have the complete (full) contents of each selected record displayed.

The **/BRIEF**, **/FULL**, **/SUMMARY**, and **/BINARY** qualifiers are mutually exclusive.

If you specify **/NOFULL** or omit the qualifier, records are displayed in the brief format.

## Example

```
$ ACCOUNTING/X25/FULL
```

The command in this example displays a full listing of the contents of each selected record.

## /IDENT

**/IDENT** — Controls whether only those records matching the specified process identifier numbers are selected.

## Format

```
/IDENT= (["-",] process-id, . . . )
```

```
/NOIDENT
```

"-" specifies that all records are selected **except** those matching the specified process identifier numbers (process IDs).

*process-id*, . . . specifies the process IDs 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 one process ID from the next with a comma and enclose the list of process IDs in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the process IDs specified in the list.

If you specify **/NOIDENT** or omit the qualifier, the process ID is not used to select records.

## Examples

1. \$ ACCOUNTING/X25/IDENT=(25634,045A6B)

The command in this example selects records matching the process IDs specified.

2. \$ ACCOUNTING/X25/IDENT=(-,2B758)

The command in this example selects all records except those matching the specified process ID.

## /LOG

**/LOG** — Controls whether informational messages (input file names, selected record counts, rejected record counts) are displayed to SYS\$OUTPUT.

## Format

**/LOG**

**/NOLOG**

By default, messages are not displayed. If more than one input file is specified in an **ACCOUNTING/X25** command with the **/LOG** qualifier, the displayed log contains one message for each file and the total count of selected and rejected records.

## Example

\$ ACCOUNTING/X25/LOG

The command in this example displays accounting records along with any informational messages such as selected record counts and rejected record counts.

## /NODE

**/NODE** — Controls whether only those records matching the specified remote node names are selected. The node name is a unique identifier for DECnet-Plus nodes and is only relevant for virtual circuits that use a Client system.

## Additional Information

The **/NODE** qualifier can be used to specify:



- A Connector node for calls to and from a Client system
- A Client system for calls made through the (named) Connector node

For other virtual circuits the name field is blank.

## Format

`/NODE= ( [ "-" , ] node-name , . . . )`

`/NONODE`

"-" specifies that all records are selected **except** those matching the specified node names.

*node-name*, . . . specifies the node names 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 one node name from the next with a comma and enclose the list of nodes in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the node names specified in the list.

If you omit the qualifier or specify `/NONODE`, the node name is not used to select records.

## Example

```
$ ACCOUNTING/X25/NODE=SAKURA
```

The command in this example selects accounting records for the node SAKURA.

## /OUTPUT

`/OUTPUT` — Specifies the output file to which the output is sent.

## Format

`/OUTPUT [= file-spec]`

`/NOOUTPUT`

*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, 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`.

If you omit the qualifier, output is directed to the current `SYS$OUTPUT` device.

## Examples

```
1. $ ACCOUNTING/X25/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.

## 2. \$ ACCOUNTING/X25/OUTPUT=STAT

The command in this example generates a brief ASCII listing of all accounting records from the currently open accounting file and outputs them to the file `STAT.LIS`. Notice that the default file type for ASCII output is `.LIS`.

## /OWNER

`/OWNER` — Controls whether only those records matching the specified owner process identifier numbers are selected.

### Format

```
/OWNER= ( ["-",] owner-process-id, . . . )
```

```
/NOOWNER
```

`"-"` specifies that all records are selected **except** those matching the specified owner process IDs.

*owner-process-id*, . . . specifies the owner process identification numbers (owner process IDs) used to select records. Owner process IDs are only present in subprocesses to specify the process IDs 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 one process ID from the next with a comma and enclose the list of process IDs in parentheses.

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/X25/OWNER=22C04F56
```

The command in this example selects records that match the specified owner process ID of `22C04F56`.

## /PRIORITY

`/PRIORITY` — Controls whether only those records matching the specified base process priority are selected.

### Format

```
/PRIORITY= ( ["-",] priority, . . . )
```

```
/NOPRIORITY
```

`"-"` specifies that all records are selected **except** those matching the specified base process 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 one priority from the next with a comma and enclose the list of priorities in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the process base priorities specified in the list.

If you specify **/NOPRIORITY** or omit the qualifier, the process base priority is not used to select records.

## Examples

1. \$ ACCOUNTING/X25/PRIORITY=3

The command in this example selects records that match a base priority of 3.

2. \$ ACCOUNTING/X25/PRIORITY=("-", 3)

The command in this example selects all records except those that match a base priority of 3.

## /REJECTED

**/REJECTED** — Controls whether unselected records are output to a specified file. These records are always in binary format.

## Format

**/REJECTED** [= *file-spec*]

**/NOREJECTED**

*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.

If you specify **/NOREJECTED** or omit the qualifier, unselected records are not output.

## Example

\$ ACCOUNTING/X25/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

**/REMOTE\_ID** — Controls whether only those records matching the specified remote IDs are selected. The remote ID is a system-dependent identifier for DECnet-Plus users and is only relevant for virtual circuits that use a Client system. For other virtual circuits, the remote ID field is null.

## Format

**/REMOTE\_ID=** (["-"], *remote-id*, . . . )

**/NOREMOTE\_ID**

"-" specifies that all records are selected **except** those matching the specified remote IDs.

*remote-id*, . . . specifies the remote IDs 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 one remote ID from the next with a comma and enclose the list of remote IDs in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the remote IDs specified in the list.

If you omit the qualifier or specify **/NOREMOTE\_ID**, the remote ID is not used to select records.

**Example**

```
$ ACCOUNTING/X25/REMOTE_ID=RID01
```

The command in this example outputs all records except for those associated with remote ID RID01.

**/REPORT**

**/REPORT** — Controls whether a specified item is included in a summary report. One column is generated on the summary report for each item specified. The **/REPORT** qualifier requires the **/SUMMARY** qualifier. Refer to the description of the **/SUMMARY** qualifier.

**Format**

```
/REPORT [= report-item, ... ]
```

```
/NOREPORT
```

*report-item*, . . . specifies the report items 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 one item from the next with a comma and enclose the list of items in parentheses.

The columns on the summary report appear in the same left-to-right sequence as the sequence in which you specify items in the list of report items.

If you specify **/REPORT** without a value (or if you specify **/SUMMARY** and do not specify **/REPORT**) then **/REPORT=RECORDS** is assumed.

**Example**

```
$ ACCOUNTING/X25/SUMMARY/REPORT=ELAPSED
```

The command in this example produces a summary report of call elapsed time.

## /SINCE

**/SINCE** — Controls whether only those records dated later than a specified time are selected. (The date on the records is the time that access to X.25 was terminated.)

### Format

**/SINCE** [= *time*]

**/NOSINCE**

*time* specifies the time used to select records. Only 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 described in the *VSI OpenVMS DCL Dictionary Part One* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-a-m/>] and *Part Two* [<https://docs.vmssoftware.com/vsi-openvms-dcl-dictionary-n-z/>].

If you specify **/SINCE** without a 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/X25/SINCE=4-DEC-2004
```

The command in this example selects records dated on or later than December 4, 2004.

## /SORT

**/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*, . . . )]

**/NOSORT**

- specifies that the sort field is used as a descending key. By default, the sequence is ascending.

*sort-item*, . . . specifies the sort items used to select records. At least one sort item must be specified. If you specify more than one sort item, separate one item from the next with a comma and enclose the list of items 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

Keyword	Meaning
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

By default, keys are assumed to be ascending. 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 identification codes (UIC) was requested and (-) specified, then the items would be sorted such that the highest identification number would be at the start of the list and the lowest at the end of the list.

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/X25/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 any of the three fields are blank.

## /SUMMARY

/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, . . . ]
```

/NOSUMMARY

*summary-item*, . . . specifies the summary items used to select records. You can specify any of the following summary items:

Keyword	Outputs
ACCOUNT	Account name from the UAF
DATE	YYYY MMM DD (year, month, date)
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 (file, PSI)
UIC	User identification code
USER	User name from the 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 one item from the next with a comma and enclose the list of items in parentheses.

The summarized items are sorted in ascending order and listed in the same left-to-right sequence as the sequence in which you specify 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/X25/SUMMARY=ACCOUNT
```

The command in this example generates a summary report by account name.

## /TERMINAL

**/TERMINAL** — Controls whether only those records matching the specified terminal names are selected. Terminal names are associated with interactive processes.

## Format

```
/TERMINAL= ( ["-",] terminal-name, . . . )
```

```
/NOTERMINAL
```

"-" specifies that all records are selected **except** those matching the specified terminal names.

*terminal-name*, . . . specifies the terminal names used to select records.

When you specify **/TERMINAL**, you must specify at least one terminal name. Specify terminal names as standard device names and include the colon (:), for example, `TTA6:`.

If you specify more than one terminal name, separate one name from the next with a comma and enclose the list of names in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the terminal names specified in the list.

If you specify **/NOTERMINAL** or omit the qualifier, the terminal name is not used to select records.

## Examples

1. `$ ACCOUNTING/X25/TERMINAL=TTB3:`

The command in this example selects records that match the terminal name TTB3:.

2. `$ ACCOUNTING/X25/TERMINAL=("-", TTB3:)`

The command in this example selects all records except those that match the terminal name TTB3:.

## /TITLE

**/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**

*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 lowercase letters, you must enclose the title in quotation marks (""). Limit the title to 34 characters or less for a terminal with a width of 80 characters.

## Example

```
$ ACCOUNTING/X25/SUMMARY=ACCOUNT/TITLE="June 2004 Accounting Report"
```

The command in this example generates a summary report and writes the following title at the top of the report:

```
June 2004 Accounting Report
```

## /TYPE

**/TYPE** — Controls whether only those records matching the specified record type are selected.

## Format

**/TYPE= (["-",] *record-type*, . . . )**

**/NOTYPE**



"-" specifies that all records are selected **except** those matching the specified record types.

*record-type*, . . . specifies the record types used to select records. You can specify the following record types:

Keyword	Meaning
FILE	Accounting file forward and backward pointers
PSI	X.25 virtual circuit termination

When you specify **/TYPE**, you must specify at least one record type. If you specify more than one record type, separate one type from the next with a comma and enclose the list of types in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the record types specified in the list.

If you specify **/NOTYPE** or omit the qualifier, the record type is not used to select records.

## Examples

1. \$ ACCOUNTING/X25/TYPE=PSI

The command in this example selects records that match the record type PSI.

2. \$ ACCOUNTING/X25/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

**/UIC** — Controls whether only those records matching the specified user identification code (UIC) are selected.

## Format

**/UIC=** ([ "-", ] *uic*, . . . )

**/NOUIC**

"-" specifies that all records are selected **except** those matching the specified UICs.

*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 one UIC from the next with a comma and enclose the list of UICs in parentheses.

Specify the UIC in the format:

[*g,m*]

where:

- *g* is an octal number in the range 0 to 377 representing the group number

- *m* is an octal number in the range 0 to 377 representing the member number

Square brackets ([ ]) or angle brackets (<>) are required in the UIC specification.

You can specify the wildcard character (\*) in either the group or member fields of the UIC specification, or in both.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the UICs specified in the list.

If you specify **/NOUIC** or omit the qualifier, the UIC is not used to select records.

## Example

```
$ ACCOUNTING/X25/UIC=( [360, *] )
```

The command in this example selects records that match UICs having a group number of 360.

## /USER

**/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 , ... )
```

**/NOUSER**

"-" specifies that all records are selected **except** those matching the specified user names.

*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 one user name from the next with a comma and enclose the list of user names in parentheses.

If the first keyword in the list is a minus sign enclosed in quotation marks ("-"), all records are selected **except** those matching the user names specified in the list.

If you specify **/NOUSER** or omit the qualifier, the user name is not used to select records.

## Examples

1. \$ ACCOUNTING/X25/USER=SASHA

The command in this example selects records that match the user name SASHA.

2. \$ ACCOUNTING/X25/USER= ("-" , SASHA)

The command in this example selects all records except those that match the user name SASHA.

# Appendix A. X.25 Accounting Record Formats

This chapter describes the structure of the data records written to the X.25 Accounting log file `SYS$MANAGER:PSIACCOUNTING.DAT`.

These records are generated by the completion of any X.25 virtual circuit access. This includes the following:

- Incoming or outgoing X.25 calls
  - Routing circuits
  - Transport (CONS) connections
  - User applications
- Incoming or outgoing X.29 calls
- PVC 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`.

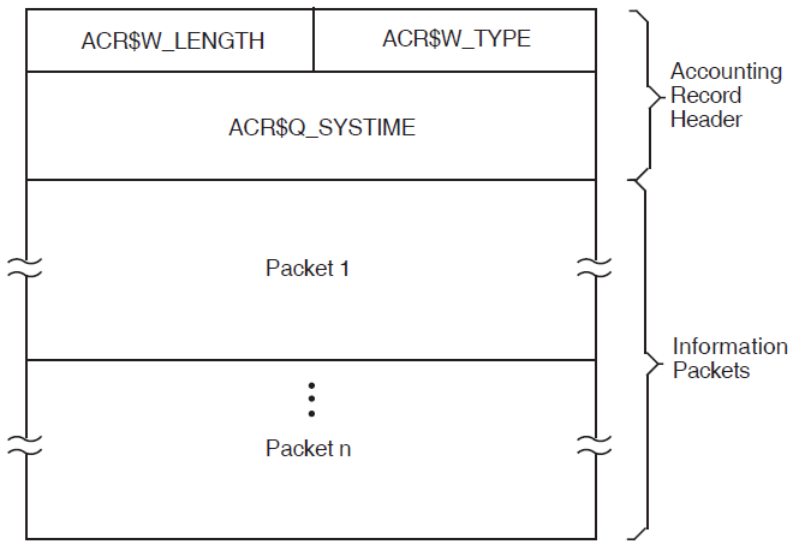
## A.1. Record Format

An X.25 for OpenVMS accounting record follows the format of the standard OpenVMS accounting records with the addition of a new record type and a new information packet. For a description of the accounting records, refer to the *VSI OpenVMS System Management Utilities Reference Manual Volume I* [[https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-1/#ACC\\_U](https://docs.vmssoftware.com/vsi-openvms-system-management-utilities-reference-manual-volume-i-a-1/#ACC_U)].

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 A.1, “Accounting Record Format” illustrates the general format of the accounting record, and Table A.1, “Fields in Accounting Record Header” describes the fields contained in this record. The type field in the accounting record header (described in Table A.1, “Fields in Accounting Record Header”) is subdivided into five further fields. Table A.2, “ACR\$W\_TYPE Fields in an Accounting Record Header” describes these fields.

**Figure A.1. Accounting Record Format**



ZK-0957-GE

**Table A.1. Fields in Accounting Record Header**

Symbolic Offset	Contents
ACR\$W_TYPE	Identifies the type of record. This field is subdivided into five fields, as described in Table A.2, "ACR\$W_TYPE Fields in an Accounting Record Header". (1 word)
ACR\$W_LENGTH	Total length of the record. (1 word)
ACR\$Q_SYSTIME	System time (64-bit absolute time). (1 quad word)

**Table A.2. ACR\$W\_TYPE Fields in an Accounting Record Header**

Symbolic Offset	Contents
ACR\$V_PACKET	Identifies this header as a record header. This bit is set to 0. (1 bit)
ACR\$V_TYPE	Identifies the type of the record. The three record types used by X.25 for OpenVMS are described in Section A.2, "Accounting Record Types". (7 bits)
ACR\$V_SUBTYPE	Indicates the subtype of the record. The only valid subtype for X.25 for OpenVMS is ACR\$K_PSI_VCT (Virtual Circuit Termination). (4 bits)
ACR\$V_VERSION	Indicates the accounting format with which the record is associated. The current value is ACR\$K_CURRVER (which is equated to ACR\$K_VERSION3). (3 bits)
ACR\$V_CUSTOMER	Identifies whether the record was written by VSI software or by customer software. In the X.25 for OpenVMS accounting file, this bit is always set to 0 to indicate that the record was written by VSI software. (1 bit)

## A.2. Accounting Record Types

Accounting record types identify the type of event that caused the record to be logged. The three accounting record types that can occur in the X.25 accounting file are shown in Table A.3, “Accounting Record Types”. Each type of accounting record requires a defined set of information packets. The table shows the information packets contained in each type of record.

**Table A.3. Accounting Record Types**

Symbol	Event	Information Packets
ACR\$K_FILE_BL	The accounting file was opened. The packet contains a backward link to the previous file. This record is present only if a previous accounting file was open.	ACR\$K_FILENAME
ACR\$K_FILE_FL	The accounting file was closed. The packet contains a forward link to the next file. This record is present only if a new accounting file will be opened after this file is closed.	ACR\$K_FILENAME
ACR\$K_PSI	An X.25 for OpenVMS event occurred.	ACR\$K_ID ACR\$K_PSI

## A.3. Accounting Packets

The header, in each of the three type of information packets, defines the type of packet as follows:

- Identification packet (ACR\$K\_ID)
- File Name packet (ACR\$K\_FILENAME)
- X.25 Data packet (ACR\$K\_PSI)

Section A.3.1, “General Format of Information Packets” describes the general format of an information packet. Section A.3.3, “Identification Packet (ACR\$K\_ID)” and Section A.3.4, “X.25 Packet (ACR\$K\_PSI)” describe the format of each type of information packet.

### A.3.1. General Format of Information Packets

Each information packet 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.

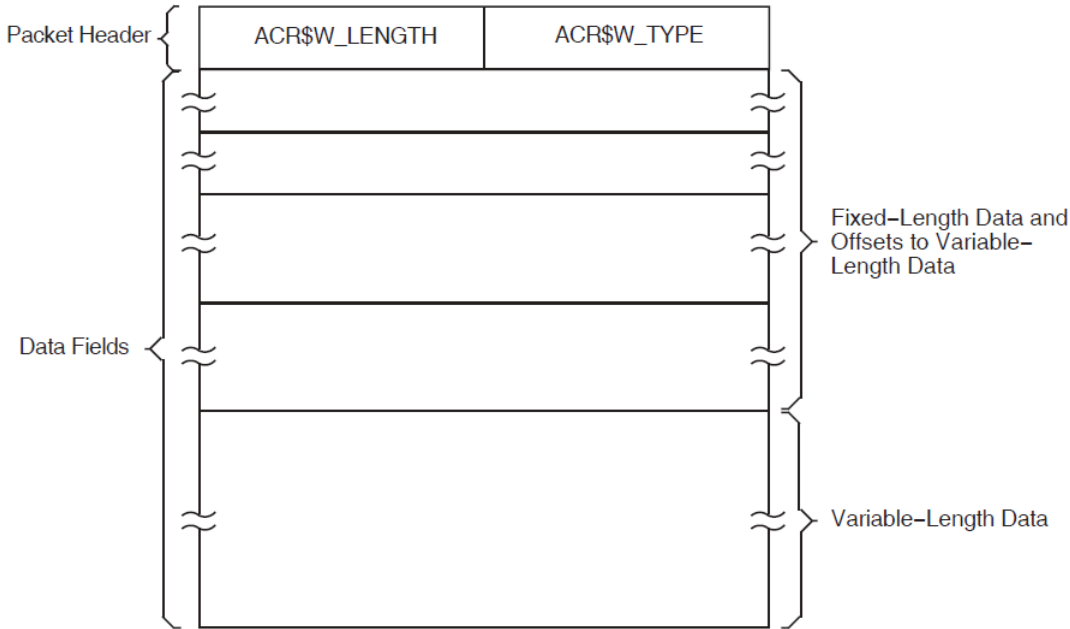
All variable-length data is represented as counted strings. Variable-length data follow the last fixed-length data field in the packet. Figure A.2, “Format of an Information Packet Format” shows the general format of an information packet. An information packet may not have values in all of its data fields. In addition, counted strings can be single-byte strings with the byte having a value of zero (indicating zero-length data).

Refer to Section A.3.3, “Identification Packet (ACR\$K\_ID)” and Section A.3.4, “X.25 Packet (ACR\$K\_PSI)” for complete descriptions of the data fields contained in each information packet.

All information packets start with a packet header. The packet header uses the same symbolic offsets as the first longword of the record header. Figure A.3, “Format of the Information Packet Header”

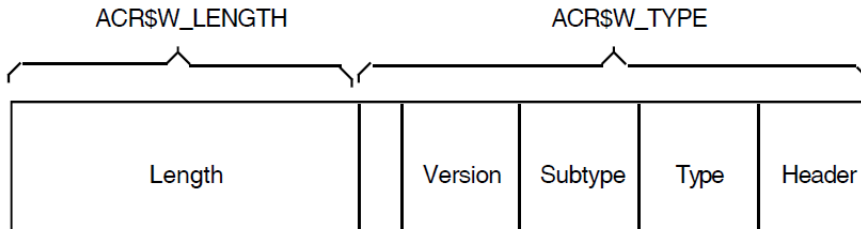
illustrates the information packet header. Table A.4, “Fields in an Information Packet Header” describes the fields in this header. The ACR\$W\_TYPE field in the information packet header is subdivided into five fields. Table A.5, “ACR\$W\_TYPE Fields in an Information Packet” describes these fields.

**Figure A.2. Format of an Information Packet Format**



ZK-0958-GE

**Figure A.3. Format of the Information Packet Header**



**Table A.4. Fields in an Information Packet Header**

Symbolic Offset	Description
ACR\$W_TYPE	Identifies the type of the packet. This field is subdivided into five fields, as described in Table A.5, “ACR\$W_TYPE Fields in an Information Packet”. (1 word)
ACR\$W_LENGTH	Total length of the packet, in bytes. (1 word)

**Table A.5. ACR\$W\_TYPE Fields in an Information Packet**

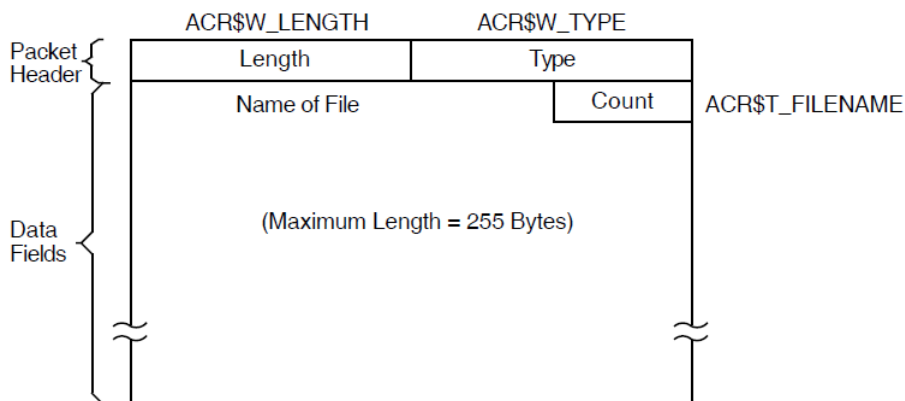
Symbolic Offset	Description
ACR\$V_PACKET	Identifies this header as a packet header. This bit is set to 1. (1 bit)
ACR\$V_TYPE	Identifies the type of the packet. There are three packet types. These packet types are described in Section A.3.3, “Identification Packet

Symbolic Offset	Description
	(ACR\$K_ID)” and Section A.3.4, “X.25 Packet (ACR\$K_PSI)”. (7 bits)
ACR\$V_SUBTYPE	Indicates the subtype of the packet. The only valid subtype for X.25 for OpenVMS is ACR\$K_PSI_VCT (Virtual Circuit Termination). (4 bits)
ACR\$V_VERSION	Indicates the accounting format with which the packet is associated. The current value is ACR\$K_CURRVER (which is equated to ACR\$K_VERSION3). (3 bits)
ACR\$V_CUSTOMER	Identifies whether the packet was written by VSI software or by customer software. In the X.25 for OpenVMS accounting file, this bit is always set to 0 to indicate that the packet was written by VSI software. (1 bit)

### A.3.2. File Name Packet (ACR\$K\_FILENAME)

The file name packet contains the name of an accounting file. Figure A.4, “Format of a File Name Packet” shows the format of the file name packet. Table A.6, “Data Fields in a File Name Packet” describes the fields contained in the packet. Refer to Section A.1, “Record Format” for information on the packet header.

**Figure A.4. Format of a File Name Packet**



ZK-0964-GE

**Table A.6. Data Fields in a File Name Packet**

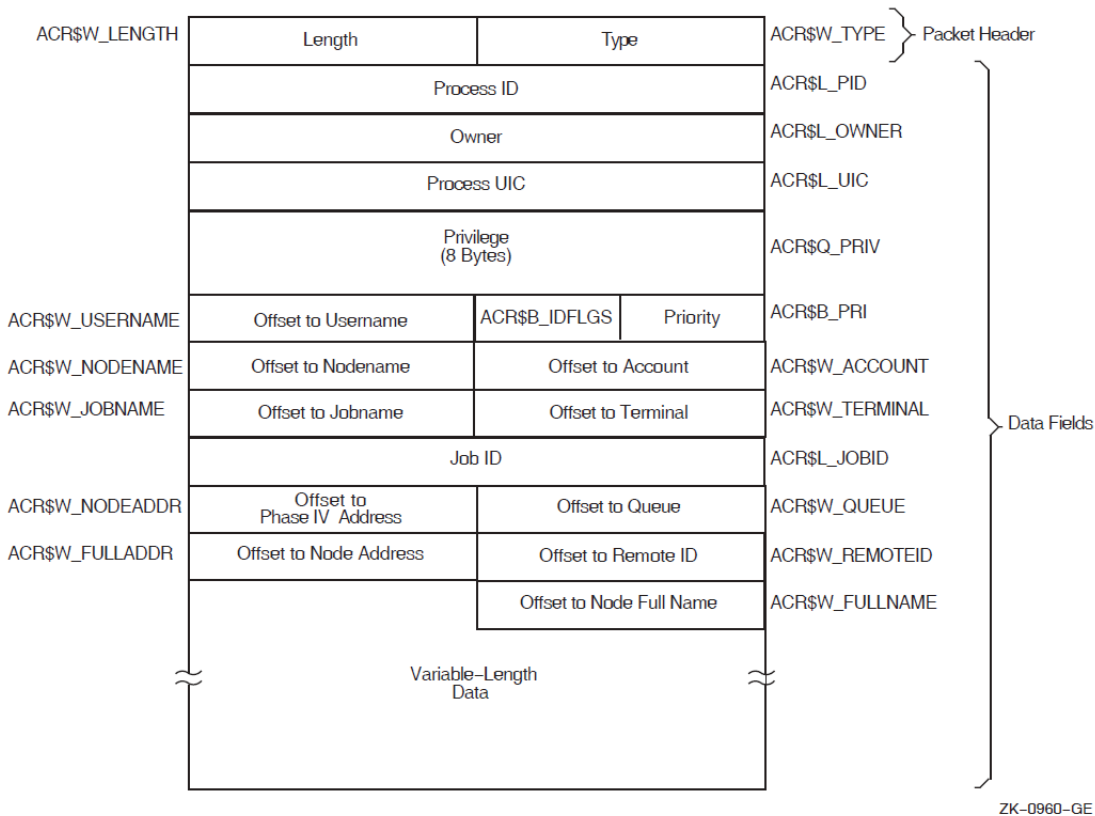
Symbolic Offset	Description
ACR\$T_FILENAME	Name of the file (counted ASCII string that gives the full file specification).

### A.3.3. Identification Packet (ACR\$K\_ID)

The identification packet identifies the process that caused the record to be logged.

Figure A.5, “Format of an Identification Packet” shows the format of the identification packet. Table A.7, “Data Fields in an Information Packet” describes the fields contained in the packet. See Section A.1, “Record Format” for information on the packet header.

**Figure A.5. Format of an Identification Packet**



**Table A.7. Data Fields in an Information Packet**

Symbolic Offset	Description
ACR\$L_PID	Process identification of the process. (longword)
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)
ACR\$L_UIC	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)
ACR\$Q_PRIV	Privileges held by the process. (quadword)
ACR\$B_PRI	Base priority of the process. (byte)
ACR\$B_IDFLGS	Flags byte. The ACR\$W_FULLADDR and ACR\$W_FULLNAME fields are present if the low bit is set. (byte)
ACR\$W_USERNAME	Offset to a counted ASCII string containing the user name of the process. (word)
ACR\$W_ACCOUNT	Offset to a counted ASCII string containing the account name of the process. (word)
ACR\$W_NODENAME	Offset to a counted ASCII string containing the Phase IV node name of the remote process. This is only relevant if the virtual circuit used a Connector or Client system. (word)
ACR\$W_TERMINAL	Offset to a counted ASCII string containing the terminal name of the process. (word)



Symbolic Offset	Description
ACR\$W_JOBNAME	Offset to a counted ASCII string containing the job name of the process. For X.25 for OpenVMS, this is always a single-byte string with a count of zero. (word)
ACR\$L_JOBID	Identification of the print or batch job. Always zero for X.25 for OpenVMS records. (longword)
ACR\$W_NODEADDR	Offset to a counted binary string containing the Phase IV remote node address. For X.25 for OpenVMS, this field is always set to zero. (word)
ACR\$W_QUEUE	Offset to a counted ASCII string containing the name of the queue with which a batch or print job is associated. For X.25 for OpenVMS, this is always a single-byte string with a count of zero. (word)
ACR\$W_REMOTEID	Offset to a counted ASCII string containing the remote ID of the remote process (varies with PSDN implementation and use). This is only relevant if the virtual circuit used a Connector or Client system. (word)
ACR\$W_FULLADDR	Offset to a counted binary string containing the complete remote node network address. On a DECnet-Plus system, this is the remote node's NSAP. This is only relevant if the virtual circuit used a Connector or Client system. (word)
ACR\$W_FULLNAME	Offset to a counted ASCII string containing the complete remote node name. On a DECnet-Plus system, this is the remote node's full name. This is only relevant if the virtual circuit used a Connector or Client system. (word)

### A.3.4. X.25 Packet (ACR\$K\_PSI)

The X.25 packet contains data on the usage of X.25 for accounting purposes. Figure A.6, "Format of an X.25 Packet" shows the format of the packet. Table A.8, "Data Fields in an X.25 Packet" describes the fields contained in the packet. Refer to Section A.1, "Record Format" for information on the packet header.

**Figure A.6. Format of an X.25 Packet**

ACR\$W_PSI_LENGTH	Length	Type	ACR\$W_PSI_TYPE
ACR\$W_LCN	LCN	Virtual circuit type	ACR\$W_PSI_VCTYPE
ACR\$Q_START_TIME	Start time (quadword)		
ACR\$W_PSI_ASSOCIATION	Association	Source name	ACR\$W_PSI_SOURCE
ACR\$W_PSI_LDTE	Local DTE	Remote DTE	ACR\$W_PSI_RDTE
ACR\$W_PSI_PVC_NAME	PVC name	DTE class name	ACR\$W_PSI_DTECLASS
ACR\$W_PSI_LOCAL_NSAP	Local NSAP	Remote NSAP	ACR\$W_PSI_REMOTE_NSAP
ACR\$Q_PSI_BYTES_TX	Data bytes sent (quadword)		
ACR\$Q_BYTES_RX	Data bytes received (quadword)		
ACR\$Q_PSI_SEGMENTS_TX	Data segments sent (quadword)		
ACR\$Q_PSI_SEGMENTS_RX	Data segments received (quadword)		
ACR\$Q_PSI_PACKETS_TX	Data packets sent (quadword)		
ACR\$Q_PSI_PACKETS_RX	Data packets received (quadword)		
ACR\$Q_PSI_MESSAGES_TX	Data messages sent (quadword)		
ACR\$Q_PSI_MESSAGES_RX	Data messages received (quadword)		
ACR\$W_PSI_OUT_THRU	Out. throughput class	Inc. throughput class	ACR\$W_PSI_INC_THRU
ACR\$W_PSI_OUT_WINDOW	Out. window size	Inc. window size	ACR\$W_PSI_INC_WINDOW
ACR\$W_PSI_OUT_PSIZE	Out. packet size	Inc. packet size	ACR\$W_PSI_INC_PSIZE
ACR\$W_PSI_CUG	CUG number	Facilities requested	ACR\$W_PSI_FAC_REQ
ACR\$W_PSI_CLEAR_FACS	Network clear facs	Originally called DTE	ACR\$W_PSI_ORIG_DTE
ACR\$W_PSI_ACCEPT_FACS	Accept facilities	Call facilities	ACR\$B_PSI_CALL_FACS
ACR\$B_PSI_DIAG	Redirect reason	Clearing diagnostic	ACR\$B_PSI_REASON
ACR\$B_PSI_REDIRECT		Clearing cause	ACR\$W_PSI_CAUSE
ACR\$W_PSI_PROTOCOL	Protocol ID	Unit name	ACR\$W_PSI_UNIT
		CUG name	ACR\$W_PSI_GROUP
	Variable-length data		

**Table A.8. Data Fields in an X.25 Packet**

Symbolic Offset	Description
ACR\$W_PSI_VCTYPE	<p>Identifies the type of virtual circuit. (word)</p> <p>The following flags are defined:</p> <ul style="list-style-type: none"> <li>● ACR\$V_PSI_OUT - Outgoing switched virtual circuit.</li> <li>● ACR\$V_PSI_PVC - Permanent virtual circuit.</li> <li>● ACR\$V_PSI_X29 - Virtual circuit used for X.29 access.</li> <li>● ACR\$V_PSI_ACCESS - Virtual circuit used by X.25 Client.</li> </ul>

Symbolic Offset	Description
	<ul style="list-style-type: none"> <li>● ACR\$V_PSI_FAIL - Access failed (for example, the call was rejected).</li> <li>● ACR\$V_PSI_GATEWAY - Virtual circuit used by X.25 Connector system.</li> </ul> <p>If neither ACR\$V_PSI_OUT nor ACR\$V_PSI_PVC are set, then the virtual circuit was an incoming SVC.</p> <p>For virtual circuits that use a Connector system, ACR\$V_PSI_ACCESS is set in the record logged on the Client system and ACR\$V_PSI_GATEWAY is set in the record logged on the Connector system.</p>
ACR\$W_PSI_LCN	Logical channel number. (word)
ACR\$Q_PSI_START_TIME	System time at the start of the virtual circuit. This is not used for incoming calls that fail to get sent to any process. (quadword)
ACR\$W_PSI_SOURCE	Offset to a counted string containing the source name (an X25 ACCESS PORT entity). (word)
ACR\$W_PSI_ASSOCIATION	Offset to a counted string which contains the name of the Call Association. (word)
ACR\$W_PSI_RDTE	Offset to a counted string containing the remote DTE address. (word)
ACR\$W_PSI_LDTE	Offset to a counted string containing the local DTE address. (word)
ACR\$W_PSI_DTECLASS	Offset to a counted string that contains the name of the DTE class. (word)
ACR\$W_PSI_PVC_NAME	Offset to a counted string which contains the PVC name. (word)
ACR\$W_PSI_REMOTE_NSAP	Offset to a counted string which contains the NSAP address of the remote user. (word)
ACR\$W_PSI_LOCAL_NSAP	Offset to a counted string which contains the NSAP address of the local user. (word)
ACR\$Q_PSI_BYTES_TX	Data bytes sent. (quadword)
ACR\$Q_PSI_BYTES_RX	Data bytes received. (quadword)
ACR\$Q_PSI_SEGMENTS_TX	Data segments sent. (quadword)
ACR\$Q_PSI_SEGMENTS_RX	Data segments received. (quadword)
ACR\$Q_PSI_PACKETS_TX	Data packets sent. (quadword)
ACR\$Q_PSI_PACKETS_RX	Data packets received. (quadword)
ACR\$Q_PSI_MESSAGES_TX	Data messages sent. (quadword)
ACR\$Q_PSI_MESSAGES_RX	Data messages received. (quadword)
ACR\$W_PSI_INC_THRU	Incoming throughput class. (word)
ACR\$W_PSI_OUT_THRU	Outgoing throughput class. (word)
ACR\$W_PSI_INC_WINDOW	Incoming window size. (word)
ACR\$W_PSI_OUT_WINDOW	Outgoing window size. (word)
ACR\$W_PSI_INC_PSIZE	Incoming packet size. (word)

Symbolic Offset	Description
ACR\$W_PSI_OUT_PSIZE	Outgoing packet size. (word)
	ACR\$W_PSI_FAC_REQ Facilities requested flags. (word) Flags defined in this field are: <ul style="list-style-type: none"> <li>● ACR\$V_PSI_REVCHG - Reverse charging.</li> <li>● ACR\$V_PSI_FAST - Fast select.</li> <li>● ACR\$V_PSI_CUG - Closed user group.</li> <li>● ACR\$V_PSI_BCUG - Bilateral closed user group (this is set only if ACR\$V_PSI_CUG is also set).</li> <li>● ACR\$V_PSDI_FAST_RES - Fast select with restricted response</li> </ul>
ACR\$W_PSI_CUG_NUMBER	Closed User Group number. (word)
ACR\$W_PSI_ORIG_DTE	Offset to a counted string which contains the address of the originally called DTE. This string will be empty (count is zero) if the call was not redirected. (word)
ACR\$W_PSI_CLEAR_FACS	Offset to a counted string containing PSDN clearing facilities. (word)
ACR\$W_PSI_CALL_FACS	Offset to a counted string containing calling facilities. (word)
ACR\$W_PSI_ACCEPT_FACS	Offset to a counted string containing call accepting facilities. (word)
ACR\$B_PSI_REASON	Reason for clearing. (byte)
ACR\$B_PSI_CAUSE	Cause for clearing. (byte)
ACR\$B_PSI_DIAG	Clearing diagnostic. (byte)
ACR\$B_PSI_REDIRECT	Call redirect reason. (byte)  The following values are possible: <ul style="list-style-type: none"> <li>● ACR\$V_PSI_BUSY - The number was busy.</li> <li>● ACR\$V_PSI_OUT_OF_ORDER - The number was out of order.</li> <li>● ACR\$V_PSI_SYSTEMATIC - The number was systematically redirected.</li> </ul> If this field is zero, the call was not redirected.
ACR\$W_PSI_UNIT	Offset to a counted string containing NW or NV unit name. (word)
ACR\$W_PSI_PROTOCOL	Offset to a counted string containing the protocol ID field. This is the first four bytes of call user data. (word)
ACR\$W_PSI_GROUP	Offset to a counted string containing the Closed User Group name. (word)

For additional information about the data fields in the X.25 packet, refer to Section 2.3.2, "Information on How X.25 for OpenVMS Is Used".

# Appendix B. A Sample X.25 Accounting Program

This appendix contains information about a sample program that analyzes X.25 Accounting records.

The program files are on the distribution kit and are copied to the `SYS$EXAMPLES :` directory. They are:

<code>SYS\$EXAMPLES : X25\$CHARGING . C</code>	Program source code (C) (IA-64 and Alpha)
<code>SYS\$EXAMPLES : PSI\$CHARGING . C</code>	Program source code (C) (VAX)
<code>SYS\$EXAMPLES : PSI\$CHARGING . EXE</code>	Executable image (VAX)
<code>SYS\$LIBRARY : PSILIB . H</code>	Symbol definition header file

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.

