

HP TP Desktop Connector for ACMS

Gateway Management Guide

January 2006

This manual contains the information needed to manage the TP Desktop Connector gateway for ACMS on the system where the HP ACMS monitor resides. It also contains information on the network transports that can be used.

| | |
|-------------------------------------|--|
| Revision Update Information: | This is a revised manual. |
| Operating System: | OpenVMS Alpha Version 8.2 OpenVMS I64 Version 8.2-1 |
| Software Version: | <i>HP TP Desktop Connector for ACMS, Version 5.0</i> |

© Copyright 2006 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP 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 HP 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. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft and Windows are US registered trademarks of Microsoft Corporation.

Java is a US trademark of Sun Microsystems, Inc.

Printed in the US

Contents

| | |
|---|------|
| Preface | vii |
| 1 Introduction to Gateway Management for ACMS | |
| 1.1 Overview | 1-1 |
| 1.2 System Management with TP Desktop Connector Software | 1-2 |
| 1.3 Network Software | 1-3 |
| 2 Managing a TP Desktop Connector System | |
| 2.1 Downloading Application Information | 2-1 |
| 2.1.1 Software Version Synchronization | 2-1 |
| 2.2 Authorizing Users | 2-2 |
| 2.2.1 Providing OpenVMS Authorization on the Submitter Node | 2-2 |
| 2.2.2 Authorizing ACMS Users | 2-4 |
| 2.2.3 Authorizing Desktop Systems Terminals | 2-4 |
| 2.2.4 Authorizing Task Access on the Application Node | 2-4 |
| 2.2.4.1 Task Authorization on Submitter Nodes | 2-5 |
| 2.2.4.2 Task Authorization on Distributed Applications | 2-5 |
| 2.3 Controlling the TP Desktop Connector System | 2-6 |
| 2.3.1 Starting the System and Applications | 2-6 |
| 2.3.2 Displaying System Information | 2-6 |
| 2.3.3 Canceling Desktop Users and Tasks | 2-6 |
| 2.4 Controlling the Gateway | 2-6 |
| 2.4.1 Starting the Gateway | 2-7 |
| 2.4.2 Gateway Startup Parameter File | 2-8 |
| 2.4.3 Specifying Network Transports | 2-10 |
| 2.4.4 Specifying the TCP/IP Port Number on the Gateway | 2-11 |
| 2.4.5 Enabling Password Expiration | 2-11 |
| 2.4.6 Cyclic Redundancy Checking | 2-12 |
| 2.4.7 Stopping the Gateway Process | 2-13 |
| 2.5 Tuning the TP Desktop Connector System | 2-13 |

| | | |
|---------|--|------|
| 2.5.1 | Setting ACMS System Parameters | 2-13 |
| 2.5.2 | Tuning the TP Desktop Connector Gateway for ACMS | 2-15 |
| 2.5.3 | Overriding Default Process Quotas | 2-15 |
| 2.5.3.1 | General Tuning Guidelines | 2-17 |
| 2.5.3.2 | AST Limit | 2-18 |
| 2.5.3.3 | Buffered I/O Limit | 2-18 |
| 2.5.3.4 | Open File Limit | 2-18 |
| 2.5.3.5 | Buffered I/O Operations Limit | 2-19 |
| 2.5.3.6 | Page File | 2-19 |
| 2.5.3.7 | Working Set and Maximum Working Set | 2-19 |
| 2.5.4 | Tuning the Desktop Client Program Buffer Size | 2-20 |
| 2.5.4.1 | Setting ACMSDI_MAXBUF on Windows Clients | 2-20 |
| 2.5.4.2 | Setting ACMSDI_MAXBUF for OpenVMS Desktop Client Programs | 2-20 |
| 2.5.5 | Running Applications Simultaneously | 2-21 |
| 2.6 | Troubleshooting | 2-21 |

3 Using Transports

| | | |
|---------|---|-----|
| 3.1 | TP Desktop Connector Transport Methodology | 3-1 |
| 3.1.1 | Using Static-Link Libraries | 3-1 |
| 3.1.2 | Default Network Transports for Client API Libraries | 3-2 |
| 3.1.3 | Using Microsoft Windows Dynamic-Link Libraries | 3-2 |
| 3.2 | Using the DECnet Transport | 3-3 |
| 3.2.1 | Preparing the Gateway for DECnet | 3-3 |
| 3.2.1.1 | Activating the Gateway for DECnet | 3-3 |
| 3.2.2 | Preparing the Client for DECnet | 3-3 |
| 3.2.3 | Troubleshooting the DECnet Transport | 3-4 |
| 3.3 | Using the TCP/IP Transport | 3-5 |
| 3.3.1 | Preparing the Gateway for the TCP/IP Transport | 3-6 |
| 3.3.1.1 | Installation Requirements for the Gateway Using TCP/IP | 3-6 |
| 3.3.1.2 | Activating the Gateway on TCP/IP | 3-6 |
| 3.3.1.3 | Setting an Alternate Gateway TCP/IP Port Number | 3-6 |
| 3.3.2 | Preparing the Client for the TCP/IP Transport | 3-6 |
| 3.3.2.1 | Configuring the Libraries for TCP/IP | 3-7 |
| 3.3.2.2 | Using the TCP/IP Transport on Microsoft Windows | 3-7 |
| 3.3.3 | Building a Tru64 UNIX TP Desktop Connector Client that Uses TCP/IP | 3-7 |

| | | |
|---------|--|-----|
| 3.3.4 | Setting Up the Environment for TCP/IP | 3-7 |
| 3.3.4.1 | Specifying the Gateway TCP/IP Port Number on the Client | 3-8 |
| 3.3.4.2 | Setting an Alternate TCP/IP Port Number on Windows | 3-8 |
| 3.3.5 | Troubleshooting the TCP/IP Transport | 3-8 |

Index

Examples

| | | |
|-----|---------------------------------------|------|
| 2-1 | Gateway Startup Parameter File | 2-8 |
| 2-2 | Gateway Process Quota Data File | 2-16 |

Figures

| | | |
|-----|--|-----|
| 1-1 | TP Desktop Connector Configuration | 1-2 |
|-----|--|-----|

Tables

| | | |
|-----|--|------|
| 1 | Manual Structure | vii |
| 1-1 | Transports Supported by TP Desktop Connector Gateway ... | 1-4 |
| 2-1 | Gateway Keywords | 2-10 |
| 2-2 | Port Numbers | 2-11 |
| 2-3 | Variables Required for ACMSPARAM.COM | 2-14 |
| 2-4 | Gateway Process Quotas | 2-16 |
| 3-1 | Default Network Transports | 3-2 |

Preface

This manual provides reference information for administration of the TP Desktop Connector gateway for ACMS.

Intended Audience

This guide is intended for application programmers, application designers, and system managers.

Manual Structure

Table 1 describes the structure of this manual:

Table 1 Manual Structure

| Chapter | Description |
|-----------|--|
| Chapter 1 | Provides an introduction to gateway management tasks. |
| Chapter 2 | Outlines the essentials of managing a desktop client program and a TP system for desktop purposes. |
| Chapter 3 | Describes how to prepare the server and the client for use with the two transports that TP Desktop Connector supports. |

Related Documents

For information on developing HP ACMS applications, refer to the following manuals:

- ***HP TP Desktop Connector for ACMS Client Application Programming Guide***
Provides information for designing, coding, and implementing a TP Desktop Connector solution that uses the client services interface.

- ***HP TP Desktop Connector for ACMS Client Services Reference Manual***
Contains the services and commands needed to create and maintain desktop client programs that use the client services interface.
- ***HP TP Desktop Connector for ACMS Installation Guide***
Provides the steps needed to install a gateway on the OpenVMS system and a TP Desktop Connector API and sample software on the desktop client system.
- ***HP TP Desktop Connector Getting Started***
Introduces you to the *HP TP Desktop Connector* for ACMS software and provides information for building client applications that use the adapter interface. It also describes how to manage the client interface environment in which the applications run.
- ***HP ACMS for OpenVMS Getting Started***
Provides a high-level discussion and examples of the activities to develop, install, and run a complete HP ACMS application.

If you are new to programming with ACMS software, HP recommends reading the following books before using the *HP TP Desktop Connector* for ACMS product:

- ***HP ACMS for OpenVMS Writing Applications***
Describes procedures to follow using the Application Development Utility (ADU).
- ***HP ACMS for OpenVMS Writing Server Procedures***
Describes how to write and debug procedures for ACMS applications. Also supplies reference information for application and system programming services.

For additional information on ACMS software, refer to the following manuals:

- ***HP ACMS for OpenVMS Introduction***
Describes basic concepts and terms concerning the ACMS environment.
- ***HP ACMS for OpenVMS ADU Reference Manual***
Describes the details of the syntax for the definitions you create and the commands you use to build the run-time components.

For information on OpenVMS programming tools, refer to this document:

- *Using HP DECset*

Describes the OpenVMS programming environment, provides helpful hints about conducting a software project, and shows a case study of HP DECset tools. Provided with the HP DECset documentation set.

The HP ACMS documentation also describes how you can use the HP DECset tools to create an effective development environment.

Conventions

This guide uses the following conventions and symbols:

| | |
|--|--|
| user input | In examples, user input is highlighted with bold type. |
| \$ | The dollar sign indicates a generic command line prompt. This prompt may be different on your system. |
| Return | A key name in a box indicates that you press that key on the keyboard. |
| Ctrl/x | Press the Ctrl (control) key and hold it down while pressing the specified key (indicated here by x). |
| WORD | Uppercase text indicates OpenVMS data types, commands, keywords, logical names, and routines or services; C files and data structures; Microsoft Windows data structures; or HyperCard data types. |
| word | In format descriptions, lowercase words indicate parameters, variables, services, or procedures. |
| <i>italics</i> | <i>Italics</i> are used for emphasis and for parameters in text. Titles of manuals and product names are also italicized. |
| [] | In format descriptions, square brackets surround a choice of options; select none, one, several, or all of the choices. |
| . | A vertical ellipsis in an example means that information not directly related to the example has been omitted. |
| TP Desktop Connector gateway | Refers to the <i>HP TP Desktop Connector</i> for ACMS software. |
| Rdb | Refers to the TP Desktop Connector gateway for ACMS. |
| | Refers to the Oracle Rdb product. |

Windows

When used alone, Windows indicates any supported member of the family of Microsoft Windows operating systems. Where necessary, specific Windows operating systems are mentioned. For a list of Microsoft Windows operating systems supported by the TP Desktop Connector product, see the product's *Software Product Description* (SPD).

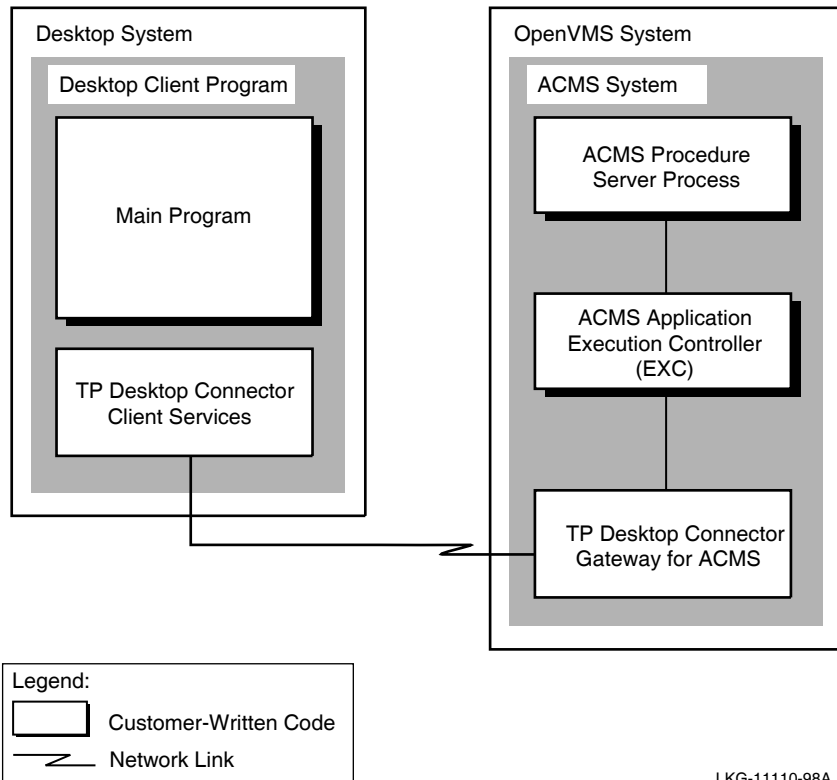
Introduction to Gateway Management for ACMS

This manual discusses the system management and administration of the TP Desktop Connector gateway for ACMS, called the gateway in this manual.

1.1 Overview

The gateway is a HP ACMS agent that communicates with other ACMS components for the desktop client program and sends the results of requests back to the desktop client. The Gateway client services call the gateway to execute requests. Access to a ACMS system through TP Desktop Connector software requires a software configuration such as the one shown in Figure 1-1.

Figure 1–1 TP Desktop Connector Configuration



LKG-11110-98A

As the manager of the system on which the gateway runs, you want to provide users with access to services. Also you want to operate the system for efficient performance.

1.2 System Management with TP Desktop Connector Software

This book describes how to perform system management with TP Desktop Connector software to authorize users, set up networks, and tune the gateway. It describes how to:

- Provide users with the access and authorization necessary for the applications, systems, and networks involved
- Cancel users and tasks
- Manage the gateway to ensure that the network is available
- Customize the gateway and system by setting parameters, user quotas, and specifying the transports to be initialized
- Set up the transports for the gateway and clients
- Determine password expiration information
- Apply cyclic redundancy checking when network or client application errors are a concern.
- Tune the ACMS system using the command procedure ACMSPARAM.COM to automatically calculate and specify system parameters suitable for use with TP Desktop Connector. Provide the accounts under which the TP Desktop Connector Process executes.

1.3 Network Software

The network software transmits messages between the desktop client program and the gateway over the network link as shown in Figure 1–1.

Although network software is required for TP Desktop Connector software to work, the programmer does not need to understand networking to develop desktop client programs. The network software is largely transparent to the desktop client program. The TP Desktop Connector client services shield the desktop client program from both the message protocol used by TP Desktop Connector software and the networking services used to implement that protocol.

To communicate, both the gateway and the desktop systems using TP Desktop Connector software must have the appropriate network software installed. For example, OpenVMS clients can use either a DECnet transport or a TCP/IP transport. The user or system manager must install this software and ensure that a network connection links the desktop client program to the gateway.

TP Desktop Connector software transparently handles all the network communications necessary to call ACMS tasks running on an OpenVMS system.

Table 1–1 lists the transports that the TP Desktop Connector gateway supports. For specific information about which transport versions are supported on each supported version on OpenVMS Alpha, see the *Software Product Description* (SPD).

Table 1–1 Transports Supported by TP Desktop Connector Gateway

| Clients | Transports |
|------------|----------------|
| Windows | TCP/IP |
| OpenVMS | DECnet, TCP/IP |
| Tru64 UNIX | TCP/IP |

Managing a TP Desktop Connector System

This chapter explains the tasks involved in managing a TP Desktop Connector system:

- Downloading application information from an OpenVMS system to the desktop system (see Section 2.1).
- Authorizing OpenVMS users of desktop client programs to be HP ACMS task submitters (see Section 2.2).
- Controlling the TP Desktop Connector system (see Section 2.3).
- Starting and stopping the TP Desktop Connector gateway for ACMS software (see Section 2.4).
- Tuning the ACMS system and gateway parameters to optimize performance (see Section 2.5).
- Troubleshooting on both the OpenVMS system and the desktop client system (see Section 2.6).

2.1 Downloading Application Information

This section provides information about synchronizing software versions on the host and desktop systems (see Section 2.1.1).

2.1.1 Software Version Synchronization

TP Desktop Connector software does not automatically download task information and forms code to the desktop. You must ensure that the version of the desktop client program running on the desktop system is consistent with the software on the ACMS application node. Refer to the recommended development procedures for version checking for in the ***HP TP Desktop Connector for ACMS Client Application Programming Guide***.

2.2 Authorizing Users

Authorizing users for TP Desktop Connector applications requires you to perform the following:

1. OpenVMS system authorization on the ACMS submitter node (see Section 2.2.1).
2. ACMS user authorization on the ACMS submitter node (see Section 2.2.2).
3. ACMS device authorization on the ACMS submitter node (see Section 2.2.3).
4. ACMS task authorization in the ACMS application definition (see Section 2.2.4).

2.2.1 Providing OpenVMS Authorization on the Submitter Node

The user of the application must be an authenticated OpenVMS user on the ACMS submitter node, that is, the node on which the gateway is running. You can authorize one account for all users or a different account for each user.

The account is not associated with a specific desktop system. A desktop user can have multiple OpenVMS accounts on a ACMS submitter node, or can have accounts on multiple systems, each system running a gateway. Authorization involves validating the user name and password sent from the desktop system to the submitter node.

When a desktop client program calls a TP Desktop Connector client service to sign a user in to the ACMS system, the gateway checks the user name and password. It ensures that all of the following are true, in the order listed:

1. The user name appears in SYSUAF.
2. The user name has not been disabled using the AUTHORIZE command `MODIFY user/FLAGS=DISUSER`.
3. The account has not expired.
4. The account has no secondary password set.

If the account has a secondary password set, the gateway rejects the sign-in.

5. At least one of the following conditions is met:
 - The user is permitted network access for the current time of the current type of day (primary or secondary).
 - The user name has been granted the ACMSDI\$CLIENT right.

6. The password passed on the call to the sign-in service matches the password stored in the SYSUAF file.
7. The password has not expired.

TP Desktop Connector software ignores the presence of an OpenVMS system password. Authorization can succeed even if a system password is set. TP Desktop Connector imposes no further OpenVMS requirements on the SYSUAF entries for user names specified in calls to the sign-in service.

The following command adds a new user name to SYSUAF allowing the new user access to TP Desktop Connector and the network:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> ADD RODRIGUEZ /NETWORK
UAF>
```

AUTHORIZE copies the settings for the user name DEFAULT to use for RODRIGUEZ, except as specified differently on the ADD command. When adding accounts, follow adequate security and password controls not shown in this example. For information on adding security and password controls, refer to the OpenVMS system management documentation.

To disallow network access for the user name, but still permit the user to sign in to the ACMS system from desktop client programs, enter the following commands:

```
UAF> ADD RODRIGUEZ /NONETWORK
.
.
.
UAF> GRANT /ID ACMSDI$CLIENT RODRIGUEZ
```

Use the TP Desktop Connector rights identifier ACMSDI\$CLIENT. The installation procedure creates the ACMSDI\$CLIENT rights identifier.

For security reasons, all types of invalid logins are reported to the desktop client program as ACMSDI_INVLOGIN. However, the protected SWL contains detailed information describing the OpenVMS authorization failure.

2.2.2 Authorizing ACMS Users

The user of the application must be an authorized ACMS user on the submitter node. Run the ACMS User Definition Utility on the submitter node and add the user account names. For example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN ACMSUDU
UDU> ADD RODRIGUEZ
UDU>
```

ACMS menus and other user attributes (except for *forms-print-file* and *forms-language*) defined in ACMSUDU are not applicable to desktop users.

Authorize as an agent the account under which the gateway runs:

```
UDU> ADD SYSTEM/AGENT
UDU>
```

If you use the system startup procedure to start the gateway, the OpenVMS account SYSTEM runs the gateway.

2.2.3 Authorizing Desktop Systems Terminals

All TP Desktop Connector users are signed in to ACMS using the NL: device specification. If you do not have the \$ALL or NL: device authorized, run the ACMS Device Definition Utility. For example:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN ACMSDDU
DDU> ADD NL
DDU> SHOW NL
.
.
.
DDU> EXIT
$
```

The device name NL authorizes all desktop devices.

2.2.4 Authorizing Task Access on the Application Node

Task authorization applies both when the user's ACMS application is running on the submitter node, and when the ACMS application is distributed.

2.2.4.1 Task Authorization on Submitter Nodes

On the submitter node, set the task access control for users in the application definition TASK ATTRIBUTES clause in one of the following ways:

- For each task, in the TASK CONTROL DEFAULTS clause
- For all tasks, in the TASK DEFAULTS clause

The user name specified in the access control list entry is the name under which the desktop user signs in to the ACMS system.

2.2.4.2 Task Authorization on Distributed Applications

The desktop client program as submitter needs authorization to allow it access to the task. The submitter must have a proxy account on the application node. On each application node, use either of the following methods to establish proxy.

- Change ACMS system parameters:

```
$ RUN ACMSGEN
ACMSGEN> SET NODE_NAME node
ACMSGEN> SET USERNAME_DEFAULT proxy_username
ACMSGEN> WRITE CURRENT
ACMSGEN>
```

To effect the change, stop and restart the ACMS system.

- Run the OpenVMS Authorize Utility. Do one of the following actions:

- To authorize all users:

```
UAF> ADD/PROXY node::* */DEFAULT
UAF>
```

All users from the specified submitter node have proxy access on the application node.

- To authorize individual users:

```
UAF> ADD/PROXY nodename::username_1 username_2
UAF>
```

The user running the account specified by *username_1* from the submitter node has individual proxy under *username_2*.

When the remote submitter node selects a task on the application node, the task's access control list (ACL) is searched to determine task access. For more information on task ACLs, refer to *HP ACMS for OpenVMS Managing Applications*.

2.3 Controlling the TP Desktop Connector System

The Sections 2.3.1 through 2.3.3 describe common operations involving control of the TP Desktop Connector system.

2.3.1 Starting the System and Applications

If all ACMS users are on desktop systems, the terminal subsystem is not required. Use the /NOTERMINALS option on startup:

```
$ ACMS/START SYSTEM/NOTERMINALS
$
```

This option saves process and memory usage, because the Terminal Subsystem Controller (TSC) process and the CP are not started.

2.3.2 Displaying System Information

TP Desktop Connector provides the ACMSDI\$GET_SUBMITTER_INFO service to supplement ACMS/SHOW USERS and ACMS/SHOW TASKS commands. Refer to *HP TP Desktop Connector for ACMS Client Services Reference Manual* for a description of the information returned by the service and the location of a sample program using the service.

2.3.3 Canceling Desktop Users and Tasks

To cancel TP Desktop Connector users, follow these steps:

1. Enter the ACMS/SHOW command with the ACMSDI\$GET_SUBMITTER_INFO service to identify desktop users and tasks.
2. Enter the ACMS/CANCEL USER command to specify a desktop user to cancel.
3. Enter the ACMS/CANCEL TASK command to specify a task to cancel.

The command procedure ACMS\$CANCEL.COM is not supported for desktop users. Because desktops appear to be inactive, the procedure cancels them at the specified interval. You can use ACMSDI\$CANCEL.COM to cancel inactive TP Desktop Connector users.

2.4 Controlling the Gateway

The following topics are discussed in this section:

- Starting the gateway process (see Section 2.4.1).
- Specifying a startup parameter file (see Section 2.4.2).
- Specifying network transports (see Section 2.4.3).

- Specifying the TCP/IP port number on the gateway (see Section 2.4.4).
- Enabling password expiration (see Section 2.4.5).
- Using cyclic redundancy checking (see Section 2.4.6).
- Stopping the gateway process (see Section 2.4.7).

2.4.1 Starting the Gateway

To start the TP Desktop Connector gateway for ACMS software, DECnet for OpenVMS software must be running. To check that DECnet software is running, use either NCP (Phase IV) or NCL (Phase V):

```
NCP> SHOW EXECUTOR
```

or

```
NCL> SHOW NODE 0 ALL STATUS
```

If the state is on, DECnet software is running.

To start the gateway process, use the command procedure found in the system startup directory to run under the SYSTEM user name.

The TP Desktop Connector startup command procedure executes from the SYSTEM user name. If you add the TP Desktop Connector startup as part of your system startup, this occurs automatically. If you need to start the TP Desktop Connector software on OpenVMS systems at times other than OpenVMS restart, try to run the procedure from the SYSTEM user name. You can accomplish this by either logging in to the OpenVMS system under that user name, or from a privileged account, setting your process privilege to CMKRNL and using the following command to restart the gateway:

```
$ @SYS$STARTUP:ACMSDI$SHUTDOWN
.
.
.
$ SUBMIT/USERNAME=SYSTEM/NOLOG ACMSDI$STARTUP.COM
```

The following example shows the interactive startup method:

```
$ @SYS$STARTUP:ACMSDI$STARTUP
%RUN-S-PROC_ID, identification of created process is 42600450
$
```

The following example shows the batch startup method:

```
$ SUBMIT/USER=SYSTEM/NOLOG SYS$STARTUP:ACMSDI$STARTUP
Job ACMSDI$STARTUP (queue SYS$BATCH, entry 1208) started on SYS$BATCH
$
```

This command procedure defines some logical names and issues a RUN command to start the gateway in a detached process with the correct privileges and a default set of qualifiers specifying the process quotas. (Refer to the logical name information for sample application code in ***HP TP Desktop Connector for ACMS Client Application Programming Guide***.) To alter the quotas, follow the guidelines given in Section 2.5.3.

If the ACMS system is stopped, users attempting to sign in on that submitter node are refused until the ACMS system is available. The ACMSDI\$SERVER process continues running. It detects when the ACMS system restarts and connects new users.

If a TP Desktop Connector application cannot sign in to the ACMS system, verify that the ACMSDI\$SERVER process is running under user name SYSTEM and that SYSTEM has been added to ACMSUDU.DAT as an agent. If the ACMSDI\$SERVER process is running under a user name other than SYSTEM, shut down TP Desktop Connector and restart it.

2.4.2 Gateway Startup Parameter File

TP Desktop Connector software provides the ACMSDI\$STARTUP command procedure for starting up the gateway. This command procedure can read an optional parameter file that you supply, containing customized settings such as user name quotas, designated transports to be initialized, integrity checking, in addition to OpenVMS parameters.

The parameter file defines logical names from keywords that determine the gateway characteristics. The logical names are defined in the process logical name table, for the process that invokes ACMSDI\$STARTUP.COM. If the keywords are not specified or are specified as null, the process logical names are not defined. For those keywords that take a value, use an equal sign (=) with or without spaces between the keyword and the value. Enclose multiple values in parentheses.

Example 2-1 is a sample parameter file.

Example 2-1 Gateway Startup Parameter File

```
!  
! What transports to start up  
!  
TRANSPORT = (DECNET,TCPIP)
```

(continued on next page)

Example 2–1 (Cont.) Gateway Startup Parameter File

```
!  
! To select the Port numbers and gateway host  
!  
TCPIP_PORT = 1023  
!  
! For Multiple Gateways for Debugging purposes  
!  
SERVER_NAME = DBG1  
DECNET_OBJECT = "TASK=DBG1"  
! To turn on integrity checking  
INTEGRITY_CHECK = YES  
!  
! To get warning messages  
!  
PASSWORD_EXP = 7  
!  
! OpenVMS Parameters  
!  
AST_LIMIT = 100  
BUFFER_LIMIT = 65535  
ENQUEUE_LIMIT = 200  
EXTENT = 2000  
FILE_LIMIT = 100  
IO_BUFFERED = 300  
IO_DIRECT = 100  
MAXIMUM_WORKING_SET = 750  
PAGE_FILE = 100000  
PRIORITY = 4  
QUEUE_LIMIT = 20  
WORKING_SET = 150
```

To specify the parameter file, use the DCL command in the following format:

```
SUBMIT SYS$STARTUP:ACMSDI$STARTUP /PARAM=SYS$STARTUP:filename
```

where:

filename is the name of the parameter file.

Table 2–1 contains the list of parameters that you can include in this file.

Table 2–1 Gateway Keywords

| Keyword | Meaning |
|-----------------|---|
| DECNET_OBJECT | Defines the object number to be used by the gateway for its DECnet object. The default is 87. |
| INTEGRITY_CHECK | Determines whether integrity checking is turned on. Specify either YES or NO. NO is the default. |
| PASSWORD_EXP | Defines the value (in days) to which a sign-in that detects an expiring password must be equal to or less than. |
| SERVER_NAME | Defines a gateway-name extension to distinguish between multiple gateways on the same node. The value is a name of up to 5 characters appended to ACMSDI\$SRV to create the process name. |
| TCPIP_PORT | Defines the number to be used by the gateway for its TCP/IP port. The default is 1023. |
| TRANSPORT | <p>Defines transports to be enabled when the gateway starts. Valid values are:</p> <ul style="list-style-type: none">• DECNET• TCPIP <p>The default is DECNET.</p> |

2.4.3 Specifying Network Transports

The gateway supports both methods of transports used by TP Desktop Connector clients: DECnet and TCP/IP. To specify which transports to activate, add the parameter TRANSPORT to the ACMSDI startup parameter file.

The following example activates both DECnet and TCP/IP transports:

```
TRANSPORT=(DECNET,TCPIP)
```

Note

DECnet object numbers are assigned by TP Desktop Connector. Do not change them.

See Chapter 3 for information on setting up the transport for your application.

2.4.4 Specifying the TCP/IP Port Number on the Gateway

TCP/IP uses numeric port numbers as endpoints of communication. The default port number is 1023, the highest unassigned privileged port number. Table 2–2 lists the three categories of port numbers.

Table 2–2 Port Numbers

| Category | Port Numbers |
|---------------------------|--------------|
| Reserved privileged ports | 0—255 |
| Privileged ports | 256—1023 |
| Nonprivileged ports | 1024—65535 |

When using TCP/IP as the network transport, specify a numeric port number as the object to which the gateway connects. To assign a TCP/IP port number to the gateway, specify the `TCPIP_PORT_NUMBER` in the startup parameter file, which is optionally passed to the ACMSDI startup command file. The following example assigns the port number 32767:

```
TCPIP_PORT_NUMBER=32767
```

2.4.5 Enabling Password Expiration

For further security checks, you can enable password expiration-checking by adding a new item, `PASSWORD_EXP`, to the `ACMSDI$STARTUP.COM` command file. This item defines the value (in days) to which a sign-in that detects an expiring password must be equal to or less than for the user to receive the status message `ACMSDI_PWDEXPIRED`. For example, if the value of `PASSWORD_EXP` is 7 and the sign-in detects that a password expires in 10 days, the sign-in call returns a normal success status. The gateway always checks the password expiration, regardless of whether the client uses the `ACMSDI_PWDEXPIRED` option. The default if you do not specify `ACMSDI_PWDEXPIRED` is 7 days.

For portable API desktop systems, the gateway checks the password expiration date only if the `ACMSDI_OPT_PWD_EXPIRING` flag is set in the sign-in options array. If you set the `ACMSDI_OPT_PWD_EXPIRING` flag, include a buffer address in the options array into which the client services can write the number of hours to expiration.

2.4.6 Cyclic Redundancy Checking

To help detect network transmission errors and some client application errors, you can turn on cyclic redundancy checking (CRC) on each message sent between the TP Desktop Connector client services and the gateway. CRC performs application data integrity checks that can prevent a gateway crash due to data corruption over the network.

In addition to preventing the gateway from crashing, CRC provides the following benefits:

- Prevents intentional modification of transaction data (for example, dollar fields or account fields)
- Provides confidence that the execution of critical transactions is correct in a client/server environment
- Provides better detection of message corruption than the network transport can provide

You turn on CRC on the gateway side by adding a parameter in the gateway startup parameter file:

```
INTEGRITY_CHECK=YES
```

The default is NO (off).

If you turn on CRC on the gateway side, CRC is performed for all clients sending messages to the gateway.

Note

Be aware that CRC is a CPU-intensive activity. Take this into consideration when deciding whether to turn integrity checking on.

If some client remote sites are more stable than others, you can be selective about which ones receive checking. To do this, leave CRC turned off by either not specifying it in the parameter file, or setting `INTEGRITY_CHECK=NO`.

On the desktop client side, define the logical name or environment variable `ACMSDI_INTEGRITY_CHECK = "YES"`. This activity turns CRC on only for the sessions coming from this particular desktop client system.

For example:

```
C> set acmsdi_integrity_check="yes"
```

Note

When deciding whether to turn CRC on, take into account that CRC is a CPU-intensive activity.

2.4.7 Stopping the Gateway Process

To stop the gateway process, use the DCL command procedure provided. For example:

```
$ @SYS$STARTUP:ACMSDI$SHUTDOWN
$
```

Any active desktop systems are disconnected and their work is aborted.

2.5 Tuning the TP Desktop Connector System

The following topics are described in this section.

- Tuning the TP Desktop Connector system by setting ACMS system parameters (see Section 2.5.1).
- Tuning the gateway process in relation to other processes on the TP Desktop Connector system (see Section 2.5.2).
- Overriding default process quotas (see Section 2.5.3).
- Setting a maximum buffer size for the client program (see Section 2.5.4).
- Running applications simultaneously (see Section 2.5.5).

2.5.1 Setting ACMS System Parameters

Used to tune the ACMS system, the ACMSPARAM.COM command procedure performs the following operations:

- Generates default values for all environment variables that are used to calculate ACMS system parameters (if values do not already exist).
- Calculates values for OpenVMS system parameters that are affected by ACMS.
- Calculates ACMS system parameters.

- Calculates quotas for the user names under which the ACMS Command Process (CP) executes. Also adds accounts and assigns privileges for these user names in the SYS\$SYSTEMLSYSUAF.DAT file.
- Writes the OpenVMS system parameters that need modification to a file.

The variable that specifies the number of desktop systems that submits to ACMSDI depends on where the gateway resides. If the gateway is on the local node, then AGENT_SUB_CNT indicates the number of desktop systems. If the gateway is on the remote node, then REMOTE_SUB_CNT indicates the number of desktop systems.

If the defaults given for these variables do not accurately reflect your system, edit ACMVARINI.DAT with corrected values and rerun ACMSPARAM.COM. For more information on variables and their effects on OpenVMS and ACMS system parameters, refer to *HP ACMS for OpenVMS Managing Applications*.

Consider the variables shown in Table 2–3 when setting ACMS system parameters for TP Desktop Connector usage.

Table 2–3 Variables Required for ACMSPARAM.COM

| Variable | |
|---------------------|--|
| AGENT_CNT | |
| <i>Description:</i> | Include in the number of agent processes one for the gateway. |
| <i>Default:</i> | 2 |
| <i>Affects:</i> | ACMSGEN parameter MSS_MAXOBJ |
| AGENT_SUB_CNT | |
| <i>Description:</i> | Include in the number of users selecting tasks in ACMS agent processes one per desktop user sign-in. |
| <i>Default:</i> | 10 |
| <i>Affects:</i> | ACMSGEN parameter MSS_MAXOBJ |
| REMOTE_NODE_CNT | |

(continued on next page)

Table 2–3 (Cont.) Variables Required for ACMSPARAM.COM

| Variable | |
|---------------------|---|
| <i>Description:</i> | Include the number of remote ACMS users connecting to the local ACC. If the gateway is on the local node, and there are no other remote ACMS systems, then REMOTE_NODE_CNT = 0. |
| <i>Default:</i> | 5 |
| <i>Affects:</i> | ACMSGEN parameter MSS_MAXOBJ. This number is used to calculate the ACC's process quotas. |
| REMOTE_SUB_CNT | |
| <i>Description:</i> | Include submitters from REMOTE_NODE_CNT. If the gateway is on the local node, then REMOTE_SUB_CNT = 0. |
| <i>Defaults:</i> | 20 * REMOTE_NODE_CNT |
| <i>Affects:</i> | ACMSGEN parameter MSS_MAXOBJ |

2.5.2 Tuning the TP Desktop Connector Gateway for ACMS

Because the size and nature of the load on the gateway varies widely from site to site, monitor the behavior of the gateway process and adjust its parameters and quotas according to the demand placed on it as you would any OpenVMS process. Section 2.5.3 describes the mechanism to override the default gateway process quotas.

2.5.3 Overriding Default Process Quotas

Override the default run-time process quotas for the gateway by supplying a file specification on the startup command line as invoked from system startup. For example:

```
$ @SYS$STARTUP:ACMSDI$STARTUP SYS$MANAGER:DESKTOP_ACMS_SERVER.DAT
```

To start the gateway from the DCL prompt, enter this command:

```
$ SUBMIT/USER=SYSTEM/NOLOG SYS$STARTUP:ACMSDI$STARTUP -
_$ /PARAMETER=(SYS$MANAGER:DESKTOP_ACMS_SERVER.DAT)
Job ACMSDI$STARTUP (queue SYS$BATCH, entry 1209) started on SYS$BATCH
$
```

Completely specify the data file, that is, device, directory, name, and type. If the specified data file cannot be accessed, the gateway uses default values for process quotas.

The data file contains lines in the following format:

value-name=desired-value

Example 2–2 shows a sample data file.

Example 2–2 Gateway Process Quota Data File

```
! Override default values for gateway start-up
!  
Page_file = 15000  
Working_set = 700           ! Required to support VERY_LARGE application
```

To comment a line, use an exclamation point (!). To make the text readable, use space and tab characters. Table 2–4 shows the values available for overriding.

Table 2–4 Gateway Process Quotas

| Process Quota | Summary Description | Alpha Default Value ¹ | I64 Default Value ¹ |
|---------------------|--|--|--------------------------------------|
| AST_LIMIT | Number of outstanding ASTs allowed | 250 | 250 |
| BUFFER_LIMIT | Maximum buffered I/O memory (bytes) | 65,536 | 65,536 |
| ENQUEUE_LIMIT | Maximum outstanding locks allowed | 2,000 | 2,000 |
| EXTENT | Maximum physical memory (pages) | 20,000 | 20,000 |
| FILE_LIMIT | Maximum open files and DECnet links | 100 | 100 |
| IO_BUFFERED | Maximum outstanding buffered I/O operations | 300 | 300 |
| IO_DIRECT | Maximum in progress direct (disk) I/O operations | 150 | 150 |
| MAXIMUM_WORKING_SET | Working set maximum size (pages) | 4,000 | 4,000 |

¹Information about tuning the quotas appears in Sections 2.5.3.1 through 2.5.3.7.

(continued on next page)

Table 2–4 (Cont.) Gateway Process Quotas

| Process Quota | Summary Description | Alpha Default Value ¹ | I64 Default Value ¹ |
|---------------|---|----------------------------------|--------------------------------|
| PAGE_FILE | Maximum reserved pages in page file | 100,000 | 100,000 |
| PRIORITY | Base process priority | 4 | 4 |
| QUEUE_LIMIT | Maximum outstanding timer queue entries | 20 | 20 |
| WORKING_SET | Number of working set pages | 2,000 | 2,000 |

¹Information about tuning the quotas appears in Sections 2.5.3.1 through 2.5.3.7.

For more information on the quotas, refer to the RUN command description in the OpenVMS DCL documentation. Sections 2.5.3.1 through 2.5.3.7 present some guidelines on tuning some of the quotas for the gateway.

2.5.3.1 General Tuning Guidelines

This section contains guidelines that you can use in deciding on initial values for quotas.

In setting initial values for the gateway quotas, assume that the gateway is working under the heaviest load you expect. Specifically, in the computations for buffered I/O limit or page file quotas, compute the quotas for the moment when:

- The maximum number of clients is signed in.
- Desktop client programs have task calls outstanding that represent the largest total passed workspace sizes.
- Desktop client programs have exchange steps in progress that represent the largest total exchange step workspace sizes.

The guidelines let you use as much information as you have about your workload. For example, you may know that although 40 desktop client programs are signed in, at most 30 have task calls in progress at the same time. Similarly, you may know that although 30 task calls may execute at the same time, at most, 20 have exchange steps in progress at the same time. You could then compute which specific concurrent task calls and exchange steps create the heaviest demand for buffered I/O and page file, and set the gateway quotas accordingly.

Alternatively, apply the guidelines assuming that the maximum number of desktop client programs are calling tasks at the same time, and that each task is executing exchange steps at the same time. This more conservative approach may overestimate the actual required quota values, but it is more tolerant of changes in your workload over time and is easier to analyze.

2.5.3.2 AST Limit

Set the limit on the outstanding ASTs (AST_LIMIT) at least as high as the number calculated from (2-1).

$$ast_limit = 13 + (4 \times active_client) \quad (2-1)$$

| | |
|----------------------|---|
| <i>active_client</i> | Number of concurrent desktop systems with active sign-ins |
|----------------------|---|

2.5.3.3 Buffered I/O Limit

Begin with the following value for the amount of buffered I/O calculated using (2-2):

$$buffer_limit = 500 + \sum_{clients} r_buffers + \sum_{calls} w_space_c + \sum_{x-step} w_space_x \quad (2-2)$$

| | |
|---------------------------|--|
| Σ <i>r_buffers</i> | Take the sum of all the sizes of receive buffers for all the desktop client programs |
|---------------------------|--|

| | |
|---------------------------|--|
| Σ <i>w_space_c</i> | Take the sum of all the sizes of passed workspaces for concurrent task calls |
|---------------------------|--|

| | |
|---------------------------|---|
| Σ <i>w_space_x</i> | Take the sum of all the sizes of workspaces for concurrent exchange steps |
|---------------------------|---|

2.5.3.4 Open File Limit

Allow the gateway an open file limit at least as high as the number calculated from (2-3):

$$file_limit = 12 + active_client \quad (2-3)$$

| | |
|----------------------|---|
| <i>active_client</i> | Number of concurrent desktop systems with active sign-ins |
|----------------------|---|

2.5.3.5 Buffered I/O Operations Limit

Allow a limit on the outstanding buffered I/O operations at least as high as the number calculated from (2-4):

$$io_buffered = 13 + (3 \times active_client) \quad (2-4)$$

active_client Number of concurrent desktop systems with active sign-ins

2.5.3.6 Page File

Requirements for page file space depend critically on the maximum workload presented to the gateway. A gateway process run with too small a page file quota exhausts its virtual memory as it tries to allocate space for internal structures. The structures are used to store the desktop client program workspaces to be transferred between the desktop system and the ACMS system.

Allow a page file space at least as high as the number calculated from (2-5):

$$page_file = 2500 + \sum_{clients} r_buffs + (10 \times a_client) + \sum_{calls} w_sp_c + \sum_{x-step} w_sp_x \quad (2-5)$$

Σr_buffs Take the sum of all the sizes of receive buffers for all the desktop client programs

a_client Number of concurrent desktop systems with active sign-ins

Σw_sp_c Take the sum of all the sizes of passed workspaces for concurrent task calls

Σw_sp_x Take the sum of all the sizes of workspaces for concurrent exchange steps

2.5.3.7 Working Set and Maximum Working Set

The optimal working set values for the gateway depend entirely on your workload. A gateway process running with working set values that are too small can begin paging heavily. Although this paging normally does not cause the gateway to fail, performance can degrade quickly, depending on patterns within your workload. Carefully monitor the gateway paging behavior and adjust the memory management quotas accordingly. Start with a WORKING_SET value of 250 pages and a MAXIMUM_WORKING_SET value of 500 pages.

2.5.4 Tuning the Desktop Client Program Buffer Size

Tuning the maximum message buffer size for your application allows you to control network traffic and optimize performance.

2.5.4.1 Setting ACMSDI_MAXBUF on Windows Clients

To define the maximum message buffer size on Windows systems, the TP Desktop Connector software provides the ACMSDI_MAXBUF environment variable that is analogous to the ACMSGEN parameter MSS_MAXBUF. If the parameter is set, the value determines the maximum size of a TP Desktop Connector message buffer. If the parameter is not set, the maximum message buffer size defaults to 1024 bytes for DECnet and TCP/IP transports. If application data exceeds the buffer size, the data is segmented and sent as multiple messages.

If memory is scarce on the desktop system, select a smaller maximum buffer size. A smaller maximum buffer size results in increased message traffic if the application data exceeds the maximum buffer size.

To set the ACMSDI_MAXBUF parameter on Windows systems, enter a command with the following syntax, or include it in the system startup file AUTOEXEC.BAT:

```
SET ACMSDI_MAXBUF=max_msg_size
```

The variable *max_msg_size* is the desired maximum message size in bytes, for example, 512. Include the command in AUTOEXEC.BAT to define the parameter at boot time. Alternatively, use a command file to set the parameter and invoke the application.

2.5.4.2 Setting ACMSDI_MAXBUF for OpenVMS Desktop Client Programs

For an OpenVMS desktop client program, set the TP Desktop Connector maximum message buffer size parameter by defining a logical name in a table available to the program. Use the following syntax:

```
DEFINE ACMSDI_MAXBUF max_msg_size
```

The gateway determines the desktop client program buffer size dynamically at run time as it reads messages.

2.5.5 Running Applications Simultaneously

The DECnet parameter MAXIMUM LINKS or MAXIMUM TRANSPORT CONNECTIONS on your desktop system controls how many copies of your desktop client program you can run simultaneously. Use either NCP (Phase IV) or NCL (Phase V) on your desktop system to set the value:

NCP DEFINE EXECUTOR MAX LINKS *integer-value*

or (for NSP links):

NCL SET NSP MAXIMUM TRANSPORT CONNECTIONS *integer-value*

or (for OSI links):

NCL SET OSI TRANSPORT MAXIMUM TRANSPORT CONNECTIONS *integer-value*

You need one link for each active file server and one link for each active desktop client program instance. Additional links are not required for multiple sign-ins started in a single desktop client program instance.

On Phase V systems, ensure that the MAXIMUM RECEIVE BUFFERS characteristic is equal to the following value:

MAXIMUM RECEIVE BUFFERS = MAXIMUM TRANSPORT CONNECTIONS * MAXIMUM WINDOWS

In addition, ensure that the MAXIMUM REMOTE NSAPS characteristic is equal to the following value:

MAXIMUM REMOTE NSAPS = MAXIMUM TRANSPORT CONNECTIONS + 1

2.6 Troubleshooting

Troubleshooting involves gathering information on the gateway system as well as on the desktop system. Follow these steps to gather information on the OpenVMS and desktop systems:

1. Before running your task, activate task audit logging on the gateway system with commands in the following format:

ACMS/MODIFY APPLICATION *application_name*/TASK=AUDIT

ACMS/SHOW APPLICATION *application_name*/TASK

This logs any application errors to the audit trail log (ATL).

2. On the desktop system, activate desktop client program error logging.

If the environment variable ACMSDI_LOG is defined, the TP Desktop Connector software writes any errors it encounters to an error log. The name of the error log file is determined by the value of ACMSDI_LOG for the following systems:

- Windows

To set the desktop client program error logging parameter on the Windows system, enter a command with the following syntax at the system prompt or in a command file:

```
SET ACMSDI_LOG=log_file_name
```

- OpenVMS

To set the parameter on the OpenVMS system, define a logical name using the following syntax:

```
DEFINE ACMSDI_LOG log_file_name
```

One error log file can exist for each invocation of a desktop client program while the parameter is defined. The first time a TP Desktop Connector client service is called, the log file is opened for write access. For each subsequent call to a TP Desktop Connector client service during an invocation of the desktop client program, the file is opened with append access.

3. Run the desktop client program to select the task.
4. Examine client.log and note the time of the error.

Ensure that the desktop system time is synchronized with the gateway system time. This helps ensure that the time stamps in the desktop client program error log are synchronized with the time stamps in the ACMS audit trail report (ATR) and software event log (SWL).

5. Examine the ATR:

```
$ MCR ACMSATR
ATR> LIST /SINCE=hh:mm
.
.
.
ATR>
```

6. Examine the SWL:

```
$ MCR SWLUP
SWLUP> LIST /SINCE=hh:mm/IMAGE=ACMSDI$SERVER
.
.
.
SWLUP>
```

The /IMAGE qualifier restricts the output to messages from the specified image.

7. Try running the task in the ACMS Task Debugger using the desktop client program to select the task (refer to ***HP TP Desktop Connector for ACMS Client Application Programming Guide*** and ***HP TP Desktop Connector for ACMS Client Services Reference Manual***).

Using Transports

This chapter discusses the TP Desktop Connector methodology for using the following transports available to TP Desktop Connector:

- DECnet
- TCP/IP

3.1 TP Desktop Connector Transport Methodology

This section outlines the transport methodologies that TP Desktop Connector applications use, giving a general understanding of why specific steps are taken in later sections.

3.1.1 Using Static-Link Libraries

TP Desktop Connector applications that use static-link libraries include:

- Tru64 UNIX for Alpha applications
- OpenVMS applications

The object libraries shipped for these clients include a transport object module. This transport object module is a layer between TP Desktop Connector and the transport. When the application is linked, the transport object becomes part of the client application executable. A single application executable can use only one transport. However, a particular application can be linked to produce different application executables that use different transports requiring different transport software.

To build an application that uses a transport other than the default transport, you must first remove the transport object from the TP Desktop Connector object library. You then place the desired transport object into the object library. You can now link an application using the chosen transport.

Certain transports have options that are not part of the standard transport software layer. These options typically define gateway connection parameters. You set these transport-specific options as environment variables when you run the application. You can set these environment variables either by the operating system or the program that uses the TP Desktop Connector API.

3.1.2 Default Network Transports for Client API Libraries

The client API static link libraries and samples, by default, use the network transports listed in Table 3–1. If a different transport is desired, the libraries must be reconfigured and the samples rebuilt to use that transport.

Table 3–1 Default Network Transports

| Desktop System | Client API Libraries | Samples |
|----------------|----------------------|---------|
| OpenVMS | DECnet | DECnet |
| Tru64 UNIX | TCP/IP | TCP/IP |

3.1.3 Using Microsoft Windows Dynamic-Link Libraries

Dynamic-link libraries (DLLs) differ from static-link libraries in that you link DLLs with applications at run time, rather than when you build the application.

If you are using a third-generation language (3GL) to code your application, you specify in your application the name of a DLL where a specific function is located.

Other methods exist for specifying the existence of DLLs, depending upon the application development environment. For example in Microsoft Visual Basic, you specify the DLL name as part of the routine declaration. At run time, Windows uses the DLL name specified by the application to locate the DLL and invoke the function. Windows searches the current directory and then all the directories listed in the PATH environment variable until a DLL with the required name is found. When Windows finds the DLL, it is loaded and you can invoke the functions provided by it.

Each TP Desktop Connector application using DLLs requires:

- TP Desktop Connector DLL
The TP Desktop Connector DLL has the entry points for the client services API.
- Network transport DLL
Copy acmsdiws.dll to acmsdi.dll for TCP/IP transport.

3.2 Using the DECnet Transport

TP Desktop Connector is designed to use DECnet by default. Therefore, only a few steps are involved in setting up the gateway and client systems to use DECnet as a transport.

3.2.1 Preparing the Gateway for DECnet

DECnet must be installed and running before you start the gateway.

3.2.1.1 Activating the Gateway for DECnet

The gateway must be activated to use the transports that its TP Desktop Connector clients will be using. There are two ways to activate the gateway for DECnet:

- Run the gateway startup procedure without using a startup parameter file, or do not specify any transport in the ACMSDI startup parameter file.

By default, the gateway starts with support for DECnet.

- Run the gateway startup procedure using a startup parameter file, specifying DECnet as a transport in the startup parameter file.

Use this approach if the gateway needs to support other transports in addition to DECnet. For example, if the gateway needs to support DECnet and TCP/IP:

1. Create a startup parameter file called:

```
SYS$STARTUP:ACMSDI$PARAMS.DAT
```

Include the following line in the file:

```
TRANSPORT=(DECNET,TCPIP)
```

2. Add the following command to your system startup command procedure:

```
@SYS$STARTUP:ACMSDI$STARTUP SYS$STARTUP:ACMSDI$PARAMS.DAT
```

3.2.2 Preparing the Client for DECnet

TP Desktop Connector supports the use of DECnet on OpenVMS platforms. On OpenVMS platforms, TP Desktop Connector software supports DECnet clients by default, no preparation is required.

3.2.3 Troubleshooting the DECnet Transport

If the client is unable to create a session with the gateway, take the following steps on the client to determine whether the client can reach the gateway node:

1. Use one of the following NCP or NCL commands to determine whether the gateway node name is defined in the client database:

```
NCP SHOW NODE server-node
```

or

```
NCL SHOW NODE server-node ALL IDENTIFIERS
```

where *server-node* is the name of the node on which the gateway resides.

2. Use various tools to determine whether the node is reachable at this time.

- a. Use NCP TELL or NCL SHOW NODE ALL STATUS:

```
NCP TELL server-node SHOW EXEC
```

or

```
NCL SHOW NODE server-node ALL STATUS
```

- b. Use SET HOST:

```
SET HOST server-node
```

If the node is reachable, but the application cannot sign in to TP Desktop Connector or the client cannot call a particular task, do the following:

1. Use SHOW SYSTEM to determine whether the gateway process is running:

```
$ SHOW SYSTEM
```

Look for the process name ACMSDI\$SERVER. If this process is present, the gateway is running.

2. Use NCP or NCL to determine whether the gateway is listening for DECnet connections. The gateway uses the DECnet object 87. For example:

```
$ MCR NCP SHOW OBJECT OBJ_87
```

```
Object Volatile Summary as of 11-MAR-2002 09:41:53
```

| Object | Number | File/PID | User Id | Password |
|--------|--------|----------|---------|----------|
| OBJ_87 | 87 | 202118DF | | |

or

```
$ MCR NCL SHOW SESSION CONTROL APPLICATION 87
```

```
Node 0 Session Control Application 87  
at 2002-05-08-11:28:33.872-04:00I586.479
```

Identifiers

```
Name = 87
```

If this object is present, the gateway is listening for connections from DECnet clients.

If a large number of users are connecting to the gateway using DECnet, you need to increase DECnet services quotas. The quotas include:

- a. Maximum number of links
 - b. Nonpaged pool parameters (SRP, IRP, or LRP quotas)
 - c. VIRPAGMAX
3. Use an ACMS command in the following format to determine whether the ACMS application is running:

```
ACMS/SHOW APPL my_appl_name
```

where:

my_appl_name is the name of your application.

If the target application is not running, you must start it.

4. Use REPLY to determine whether the gateway node is logging network errors. For example:

```
$ REPLY/ENABLE=(NETWORK)
```

Refer to the DECnet documentation for more information.

3.3 Using the TCP/IP Transport

Transmission Control Protocol Internet Protocol (TCP/IP) is an Internet protocol suite. TCP/IP offers two means of task-to-task communications, UDP and TCP. TP Desktop Connector uses TCP for communications from client to server. TCP has the following features:

- Connection-based
- Reliable
- Sequenced
- Byte-stream service

- Connection set-up overhead

3.3.1 Preparing the Gateway for the TCP/IP Transport

The gateway uses TCP/IP Services for OpenVMS. See the *Software Product Description* for information about the supported versions of the TCP/IP Services for OpenVMS product. The QIO services, provided by this product, perform TCP/IP I/O.

When TCP/IP is activated in the gateway, it listens for TCP/IP connections.

3.3.1.1 Installation Requirements for the Gateway Using TCP/IP

When using TCP/IP as a transport, you must install and start TCP/IP Services for OpenVMS software before starting the gateway.

3.3.1.2 Activating the Gateway on TCP/IP

By default, TCP/IP is not activated by the gateway. The gateway must be started with a parameter file. You must place the string TCPIP in the transport list. For example, to activate TCP/IP and DECnet in the gateway, you can place the following line in the gateway parameter file:

```
Transport=(TCPIP,DECnet)
```

This line causes the gateway to listen for connections on both DECnet and TCP/IP transports.

3.3.1.3 Setting an Alternate Gateway TCP/IP Port Number

By default, the gateway listens for connections to TCP/IP port number 1023. This is the highest privileged port number provided by TCP/IP Services, and is the recommended port number to use. However, if an alternate port number must be used, include a line in the gateway parameter file with the desired port number. For example:

```
TCPIP_PORT=1022
```

This line causes the gateway to listen for connect requests to TCP/IP port number 1022 on the gateway system. Of course, the client must also be aware of the alternate port number to use. See Section 3.3.2 for information on the TCP/IP client implementation.

3.3.2 Preparing the Client for the TCP/IP Transport

The following clients currently support the TCP/IP transport:

- Microsoft Windows (using the WINSOCK TCP/IP implementation)
- OpenVMS
- Tru64 UNIX

3.3.2.1 Configuring the Libraries for TCP/IP

The following sections describe how to configure your dynamic-link libraries for TCP/IP (see Section 3.3.2.2).

3.3.2.2 Using the TCP/IP Transport on Microsoft Windows

Several TCP/IP vendors have agreed on a standard interface to TCP/IP on Microsoft Windows. This standard is known as WINSOCK. Using the WINSOCK standard, you can write applications without regard to the specific TCP/IP Microsoft Windows implementation, if the implementation supports the WINSOCK standard.

To use this standard with Windows, TP Desktop Connector provides a TP Desktop Connector DLL that supports the WINSOCK TCP/IP implementation. To use the TCP/IP WINSOCK transport, select the TP Desktop Connector ACMSDIWS.DLL. Create a copy of the ACMSDIWS.DLL file and rename this to ACMSDI.DLL. Place the ACMSDI.DLL file in the executable path so that it can be located by the operating system when running the application.

These files are located on the OpenVMS server in the SYS\$COMMON:[ACMSDI] directory tree in the appropriate self-extracting archive. Although the files are unique to the operating system, they have the same file name. Refer to ***HP TP Desktop Connector for ACMS Installation Guide*** to locate the appropriate file for your Windows operating system.

3.3.3 Building a Tru64 UNIX TP Desktop Connector Client that Uses TCP/IP

The Tru64 UNIX client supports the TCP/IP that is built into the library LIBACMSDI.A. The TP Desktop Connector kit includes this code. To build a TCP/IP-based client, the default DECnet object is removed from the TP Desktop Connector library and replaced with the TCP/IP transport module. Refer to ***HP TP Desktop Connector for ACMS Installation Guide*** for more information on installing client software and building the samples.

3.3.4 Setting Up the Environment for TCP/IP

The following sections discuss setting up the environment to use the TCP/IP transport with Windows clients.

3.3.4.1 Specifying the Gateway TCP/IP Port Number on the Client

The default TCP/IP port number used on the gateway is 1023. If the default is used, no action is necessary.

If the port TCP/IP port number 1023 is unacceptable, you can override the default by setting an environment variable on the clients, at the system or program level. However, you must specify the same port number on the gateway as you do on the client.

3.3.4.2 Setting an Alternate TCP/IP Port Number on Windows

You can override the default TCP/IP port number in the following manner:

```
SET ACMSDI_TCPIP_PORT_host=1022
```

where:

host is the TCP/IP name of the gateway system.

3.3.5 Troubleshooting the TCP/IP Transport

To insure that the gateway is listening for TCP/IP connections, perform the following steps:

1. Get information on the port that TP Desktop Connector is monitoring for connections:

```
$ TCPIP SHOW DEVICE_SOCKET/PORT=1023/FULL
```

```
Device_socket: bg2139          LOCAL          REMOTE
                                Port:          1023          0
                                Type: STREAM   Host:          0.0.0.0      0.0.0.0
                                Service:
                                RECEIVED        SEND
                                Queued I/O       0            0
                                QLEN            0            0
                                QLEN            0            4096
                                QLIMIT          5            0
                                TIMEO           0           16384
                                ERROR            0            0
                                OOBMARK         0            0
                                I/O completed    0            0
                                Bytes transferred 0            0

Options:  ACCEPTCONN KEEPALIVE
State:    PRIV
RCV Buff: None
SND Buff: None
```

2. Verify that the port is owned by the gateway, using the device_socket number:

```
$ SHOW DEVICE BG2139:/FULL
```

Device BG2139: is online, mounted, network device, mailbox device.

| | | | |
|------------------|------------------|----------------------|-------------------|
| Error count | 0 | Operations completed | 1 |
| Owner process | "ACMSDI\$SERVER" | Owner UIC | [SOFTSUPP,SYSTEM] |
| Owner process ID | 5C000234 | Dev Prot | |
| Reference count | 1 | Default buffer size | 256 |

If a large number of users are connecting to the gateway using TCP/IP, you need to increase TCP/IP Services quotas. The quotas include:

- Number of device sockets
- Number of IRPs
- Internet memory buffers

Refer to *TCP/IP Services for OpenVMS System Manager's Guide* for more information.

Index

A

ACMSDI\$CLIENT
 authorization check, 2-2
 use, 2-3
ACMSDI\$SHUTDOWN.COM
 procedure use, 2-13
ACMSDI\$STARTUP.COM
 procedure use, 2-7, 2-15
ACMSDI_LOG
 environment parameter, 2-22
ACMSDI_MAXBUF
 setting, 2-20
ACMSPARAM.COM
 variables required for, 2-14
ACMS system
 displaying information, 2-6
 parameter setting, 2-14
 starting, 2-6
AGENT_CNT
 desktop effect on variable, 2-14
AGENT_SUB_CNT
 desktop effect on variable, 2-14
Application
 running simultaneously, 2-21
AST limit
 initial value, 2-18
 TP Desktop Connector gateway for ACMS
 default value, 2-16
Authorization, 2-2
 ACMS user, 2-4
 desktop system terminal, 2-4
 OpenVMS submitter node, 2-2
 task access, 2-4

B

Buffered I/O
 limit
 default value, 2-16
 initial value, 2-18
 operations
 default value, 2-16
 initial value, 2-19
Buffer size
 setting desktop client program, 2-20
Building
 TCP/IP-based client, 3-7

C

Communication
 description, 1-3
Component
 network, 1-3
Configuration
 desktop system, 1-1
Cyclic redundancy checking, 2-12

D

DECnet transport, 3-3
Default network transports, 3-2
Desktop client program
 running simultaneously, 2-21
Direct I/O
 TP Desktop Connector gateway for ACMS
 default value, 2-16

Dynamic-link libraries, 3–2

E

Enqueue limit

TP Desktop Connector gateway for ACMS
default value, 2–16

Environment variables, 2–13

Error

desktop client program logging, 2–22

Extent

TP Desktop Connector gateway for ACMS
default value, 2–16

F

File limit

TP Desktop Connector gateway for ACMS
adjusting, 2–18
default value, 2–16

G

Gateway

account as agent
authorizing, 2–4
TP Desktop Connector
OpenVMS authorization, 2–2
overriding defaults, 2–15
process quota defaults, 2–16
starting, 2–7
stopping, 2–13
tuning process quotas, 2–15

I

Integrity checking, 2–12

L

Libraries

dynamic-link, 3–2
static-link, 3–1

Limit

AST, 2–18
buffered I/O

Limit

buffered I/O (cont'd)
initial, 2–18
operations, 2–19
open file, 2–18

Links

use, 2–21

Locks outstanding limit

TP Desktop Connector gateway for ACMS
default value, 2–16

Logging

desktop client program, 2–22

M

Management

description, 1–2
procedures, 2–1

Microsoft DOS

message buffer
setting, 2–20

N

Network

component description, 1–3

O

OpenVMS

ACMSDI_LOG, 2–22
authorization, 2–2
message buffer
setting, 2–20
transports, 1–4

P

Page file

TP Desktop Connector gateway for ACMS
adjusting, 2–19
default value, 2–16

Password

system
in TP Desktop Connector
authorization, 2–3

Priority
 TP Desktop Connector gateway for ACMS
 default value, 2–17

Process
 quotas
 TP Desktop Connector gateway for
 ACMS, 2–15

R

REMOTE_NODE_CNT
 Desktop effect on variable, 2–14
REMOTE_SUB_CNT
 Desktop effect on variable, 2–15
Requirements for applications using DLLs,
 3–2
Rights identifier
 ACMSDI\$CLIENT, 2–3

S

Shutdown
 TP Desktop Connector gateway for ACMS,
 2–13
Sign in
 log file access, 2–22
Software
 synchronizing versions, 2–1
Specifying port number, 3–8
Startup
 TP Desktop Connector gateway
 user-name assumptions, 2–7
 TP Desktop Connector gateway for ACMS,
 2–7
Static-link libraries, 3–1
System
 user-name assumptions, 2–7

T

Task
 canceling, 2–6
TCP/IP environment, 3–7

TCP/IP transport, 3–5
 using Multinet, 3–6

Time
 synchronizing gateway and desktop
 system, 2–22

Timer queue entries
 TP Desktop Connector gateway for ACMS
 default value, 2–17

TP Desktop Connector gateway for ACMS
 starting, 2–7
 stopping, 2–13
 tuning process quotas, 2–15

Transport
 DECnet, 3–3
 TCP/IP, 3–5

Transport methodology, 3–1

Troubleshooting
 procedures for, 2–21

Tuning
 TP Desktop Connector gateway for ACMS,
 2–15

U

User
 authorization, 2–2
 canceling, 2–6

V

Version
 synchronizing software, 2–1

W

WINSOCK standard, 3–7

Working set
 TP Desktop Connector gateway for ACMS
 adjusting, 2–19
 default value, 2–17
 maximum size
 default value, 2–16

