

# **Connect:Direct<sup>®</sup> for OpenVMS**

## **Installation and Administration Guide**

**Version 3.5**

**Connect:Direct for OpenVMS Installation and Administration Guide  
Version 3.5**

**First Edition**

Copyright © 1989 - 2010.

Sterling Commerce, Inc.

ALL RIGHTS RESERVED

**STERLING COMMERCE SOFTWARE**

\*\*\*TRADE SECRET NOTICE\*\*\*

THE CONNECT:DIRECT® FOR OPENVMS SOFTWARE (“STERLING COMMERCE SOFTWARE”) IS THE CONFIDENTIAL AND TRADE SECRET PROPERTY OF STERLING COMMERCE, INC., ITS AFFILIATED COMPANIES OR ITS OR THEIR LICENSORS, AND IS PROVIDED UNDER THE TERMS OF A LICENSE AGREEMENT. NO DUPLICATION OR DISCLOSURE WITHOUT PRIOR WRITTEN PERMISSION. RESTRICTED RIGHTS.

This documentation was prepared to assist licensed users of the Sterling Commerce Connect:Direct for OpenVMS software. The Sterling Commerce Software, the related documentation and the information and know-how it contains, is proprietary and confidential and constitutes valuable trade secrets of Sterling Commerce, Inc., its affiliated companies or its or their licensors (collectively “Sterling Commerce”), and may not be used for any unauthorized purpose or disclosed to others without the prior written permission of Sterling Commerce. The Sterling Commerce Software and the information and know-how it contains have been provided pursuant to a license agreement which contains prohibitions against and/or restrictions on its copying, modification and use. Duplication, in whole or in part, if and when permitted, shall bear this notice and the Sterling Commerce, Inc. copyright legend.

Where any of the Sterling Commerce Software is used, duplicated or disclosed by or to the United States government or a government contractor or subcontractor, it is provided with RESTRICTED RIGHTS as defined in Title 48 CFR 52.227-19 and is subject to the following: Title 48 CFR 2.101, 12.212, 52.227-19, 227.7201 through 227.7202-4, FAR 52.227-14(g)(2)(6/87), and FAR 52.227-19(c)(2) and (6/87), and where applicable, the customary Sterling Commerce license, as described in Title 48 CFR 227-7202-3 with respect to commercial software and commercial software documentation including DFAR 252.227-7013(c) (1), 252.227-7015(b) and (2), DFAR 252.227-7015(b)(6/95), DFAR 227.7202-3(a), all as applicable.

References in this manual to Sterling Commerce products, programs, or services do not imply that Sterling Commerce intends to make these available in all countries in which Sterling Commerce operates.

Printed in the United States of America

**WARRANTY DISCLAIMER**

The Sterling Commerce Software and the related documentation are licensed either "AS IS" or with a limited warranty, as described in the Sterling Commerce license agreement. Other than any such limited warranties provided in the license agreement, NO OTHER WARRANTY IS EXPRESSED AND NONE SHALL BE IMPLIED, INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE OR FOR A PARTICULAR PURPOSE. The applicable Sterling Commerce entity reserves the right to revise this publication from time to time and to make changes in the content hereof without the obligation to notify any person or entity of such revisions or changes.

Connect:Direct is a registered trademark of Sterling Commerce. All Third Party Product names are trademarks or registered trademarks of their respective companies.

---

# Contents

<b>Chapter 1</b>	<b>About Connect:Direct for OpenVMS</b>	
	Connect:Direct for OpenVMS Documentation.....	7
	About This Guide.....	7
	Chapter Overview .....	7
	Before You Begin .....	8
	Conventions Used in This Guide .....	8
<b>Chapter 2</b>	<b>Reviewing the Preinstallation Procedure</b>	
	Reviewing Installation Configuration Options .....	11
	Reviewing the Preinstallation Steps.....	11
	Step 1: Setting Up the Connect:Direct for OpenVMS Manager Account .....	12
	Step 2: Setting Up Disk Quotas .....	12
<b>Chapter 3</b>	<b>Installing Connect:Direct for OpenVMS</b>	
	Understanding the Types of Configurations .....	15
	Reviewing the Installation Steps.....	16
	Step 1: Downloading and transferring the product executable .....	16
	Step 2: Logging in to the System Manager account .....	16
	Step 3: Invoking VMSINSTAL .....	17
	Loading the Connect:Direct for OpenVMS Software .....	17
	Responding to the Next Prompts .....	17
	VMSINSTAL Begins the Installation.....	17
	Step 4: Selecting Installation Options .....	17
	Option 1. Selecting the Connect:Direct for OpenVMS Target Directory .....	18
	Option 2. Purging Old Versions of the Connect:Direct for OpenVMS Product .....	18
	Option 3. Copying the Connect:Direct for OpenVMS Help File into the OpenVMS Help Library .....	18
	Option 4. Installing the NDMUI Command .....	18
	Upgrading to a New Release .....	19
	Reviewing Cluster Considerations .....	19
	Step 5: Ending the Installation .....	19

Step 6: Setting Default Directory to the Connect:Direct Directory .....	20
Step 7: Verifying Connect:Direct for OpenVMS File Installation.....	20
Reviewing Client Files .....	20
Reviewing Server Files .....	21
Step 8: Configuring the Connect:Direct for OpenVMS Environment .....	22

## Chapter 4      **Configuring Connect:Direct for OpenVMS**

Reviewing the Types of Configurations.....	23
Reviewing the Configuration Steps.....	23
Step 1: Logging In to the System Manager Account .....	24
Step 2: Invoking SETUPRIGHTS for the Server .....	24
Step 3: Granting the NDM_ADMIN Identifier to the Designated Connect:Direct for OpenVMS Manager Account on the Server Node .....	25
Step 4: Configuring the Connect:Direct for OpenVMS environment.....	25
Step 4A: Configuring a Server/Client Installation .....	26
Defining Connect:Direct for OpenVMS Initialization Parameters.....	28
Defining Remote Connect:Direct for OpenVMS Servers.....	31
Starting Connect:Direct for OpenVMS (Server).....	31
Stopping Connect:Direct for OpenVMS (Server).....	32
Reviewing Cluster Considerations for Starting Connect:Direct for OpenVMS (Server)	32
Reviewing Cluster Considerations for Stopping Connect:Direct for OpenVMS (Server)	32
Step 4B: Configuring Client-Only Installations.....	33
Starting Connect:Direct for OpenVMS (Client Only) .....	33
Stopping Connect:Direct for OpenVMS (Client Only) .....	33
Step 5: Editing the Network Map .....	34
Step 6: Testing the Connect:Direct for OpenVMS System.....	34

## Chapter 5      **Upgrading Connect:Direct for OpenVMS**

Planning for an Upgrade of Connect:Direct for OpenVMS.....	37
Directly Replacing the Existing Connect:Direct for OpenVMS Software.....	37
Installing Connect:Direct for OpenVMS Software in an Alternate Environment .....	38
Starting the Connect:Direct for OpenVMS System in an Alternate Environment .....	38
Cluster Considerations for Starting Connect:Direct for OpenVMS (Server) .....	39
Stopping the Connect:Direct for OpenVMS System in an Alternate Environment.....	39
Cluster Considerations .....	39
Connecting to the Connect:Direct for OpenVMS System in an Alternate Environment	39

## Chapter 6      **Maintaining Security**

Understanding Connect:Direct for OpenVMS Security.....	41
Signing On to the Connect:Direct for OpenVMS System .....	41

Understanding Security Considerations.....	41
Checking User Interface Security .....	42
Checking Security for Connect:Direct for OpenVMS Processes .....	42
Understanding Default Proxies.....	43
Creating a Proxy Entry .....	43
Using Connect:Direct for OpenVMS Proxy File.....	44
Reviewing the File Format .....	44
Using Proxy Wildcards.....	44
Establishing Permitted Neighbor Support.....	45
Creating a Permitted Neighbor Database.....	45
Activating a Permitted Neighbor Database .....	46
Modifying Permitted Neighbor Database .....	46
Sample Permitted Database Entry .....	47
Understanding Rights Identifiers .....	49
Assigning Rights Identifiers .....	50
Using Numeric Usernames .....	50
Managing Remote Access Control with a Password File.....	51
Generating a Password File .....	51
Input File Format .....	51
Password Source File Example.....	52
Modifying a Password File .....	52
Using a Password File.....	53
Specifying an snodeid Value .....	53
Performing Placeholder Substitution.....	53
Using a DCL Procedure .....	54

## **Chapter 7 Reviewing Additional Installation Considerations**

Using the Connect:Direct for OpenVMS Message Utility .....	55
Viewing Text Messages.....	56
Using the Connect:Direct for OpenVMS Translation Table Utility.....	56
Modifying the SYLOGIN.COM File.....	59
Activating Statistics Archival and Deletion.....	59
Deferring Statistics Files Records.....	59
Activating Statistics Archive-On-Demand .....	60
Sample Statistics Archival Procedure.....	60
Performing Daily Archival .....	61

## **Chapter 8 Modifying the Network Map**

Defining a Network Map .....	63
Reviewing a Sample NETDEF.COM Procedure .....	64
Adding Records to the Network Map .....	66
Using the INSERT NETMAP LOCAL_NODE Command .....	66
Using the INSERT NETMAP ADJACENT_NODE Command .....	66
Deleting Records from the Network Map.....	68
Using the DELETE NETMAP LOCAL_NODE Command .....	68

Using the DELETE NETMAP ADJACENT_NODE Command .....	69
Reviewing Examples of DELETE NETMAP Commands.....	69
Updating Records in the Network Map.....	69
Using the UPDATE NETMAP LOCAL_NODE Command .....	69
Using the UPDATE NETMAP ADJACENT_NODE Command.....	70
Updating a TCP/IP Type ADJACENT_NODE .....	70
Reviewing an Example of UPDATE NETMAP Command .....	71

## Glossary

## Index

---

# About Connect:Direct for OpenVMS

The Connect:Direct product links technologies and moves all types of information between networked systems/computers. It manages high-performance transfers by providing such features as: automation, reliability, efficient use of resources, application integration, and ease of use. Connect:Direct software offers choices in communications protocols, hardware platforms, and operating systems. It provides the flexibility to move information among mainframes, midrange systems, desktop systems, and LAN-based workstations.

The Connect:Direct for OpenVMS product supports connectivity between OpenVMS systems and the following systems:

- ❖ IBM systems (z/OS, VM, and VSE) using Transmission Control Protocol/Internet Protocol (TCP/IP)
- ❖ Alpha AXP systems using DECnet and TCP/IP
- ❖ HP NonStop, UNIX, NT, and OS/400 systems using TCP/IP

Refer to the *Connect:Direct for OpenVMS Release Notes* for enhancement and release-specific information.

---

## Connect:Direct for OpenVMS Documentation

See *Connect:Direct for OpenVMS Release Notes* for a complete list of the product documentation.

### About This Guide

The *Connect:Direct for OpenVMS Installation and Administration Guide* is for programmers and network operations staff who install and maintain the Connect:Direct for OpenVMS product.

This guide assumes knowledge of the Digital Equipment Corporation (DEC) OpenVMS operating system, including its applications, network, and environment. If you are not familiar with the OpenVMS operating system, refer to the DEC library of manuals.

Chapters 2 through 4 step you through the basic installation and configuration processes. Chapter 8 provides information to define your local and remote nodes in the network map. See *Before You Begin* on page 8 to help you gather the information you will need. To set up secure connections using Connect:Direct Secure+ Option for OpenVMS, see *Connect:Direct Secure+ Option for OpenVMS, Implementation Guide*.

---

## Chapter Overview

The organization of the *Connect:Direct for OpenVMS Installation and Administration Guide* follows:

- ❖ Chapter 1, *About Connect:Direct for OpenVMS*, provides general information about the Connect:Direct product and describes how Connect:Direct OpenVMS works.
- ❖ Chapter 2, *Reviewing the Preinstallation Procedure*, describes preinstallation steps to be performed on the OpenVMS system.
- ❖ Chapter 3, *Installing Connect:Direct for OpenVMS*, describes the product installation procedure.
- ❖ Chapter 4, *Configuring Connect:Direct for OpenVMS*, describes how to configure Connect:Direct software for your site.
- ❖ Chapter 5, *Upgrading Connect:Direct for OpenVMS*, describes procedures for applying a maintenance release of the product.
- ❖ Chapter 6, *Maintaining Security*, describes the product's security structure, how to use a proxy file, and how to manage remote access.
- ❖ Chapter 7, *Reviewing Additional Installation Considerations*, describes additional installation considerations, which include security, exits, message utility, and translation table utility.
- ❖ Chapter 8, *Modifying the Network Map*, describes the network map.
- ❖ *Glossary* defines Connect:Direct terms used in this manual.

---

## Before You Begin

This information is required during the installation and configuration processes. The information in parentheses is used in examples throughout this manual.

Connect:Direct for OpenVMS  
**Manager Account** \_\_\_\_\_ (manager)  
 User-defined

Connect:Direct for OpenVMS  
**Base Directory** \_\_\_\_\_ (ndmdir)  
 User-defined for the NDM\$\$DIRECTORY logical

Distribution product name: xxxxxxnnn \_\_\_\_\_

Fully qualified target directory: [SYS\$SYSDEVICE:[NDM]] \_\_\_\_\_

The information from the following command is required to request a license key from Sterling Commerce:

```
$ run ndm_license_display_profile.exe
```

Connect:Direct environment to configure: \_\_\_\_\_

Connect:Direct node name for this system: \_\_\_\_\_  
 [DECnet\_node.VMS.ndm\_environment]

---

## Conventions Used in This Guide

The *Connect:Direct for OpenVMS Installation and Administration Guide* uses certain notational conventions. This section describes the conventions used in this guide.



Convention	Description
UPPERCASE LETTERS	Uppercase letters in the command format indicate that you type in information as shown.
UPPERCASE and lowercase Letters	A statement, command, or parameter in uppercase letters followed by lowercase letters indicates an alternative to typing the entire command. For example, <code>SElect PROCess</code> means that you need only type <code>SEL PROC</code> for the command to be valid.
lowercase letters	Lowercase letters or words in commands or syntax boxes require substitution by the user. For example, <code>PNODE=primary-node-name</code> indicates that you must provide the name of the primary node.
Bold Letters	Bold print in syntax boxes indicates Connect:Direct commands and required parameters. For example, <code>DSN=filename</code> indicates that the parameter <code>DSN</code> is required.
Underlined Letters	Underlining indicates default values for parameters and subparameters. For example, <code>RETAIN=Yes No Initial</code> specifies that the default for <code>RETAIN</code> is <code>NO</code> .
Vertical Bars ( )	Vertical bars indicate that you can supply one of a series of values separated by the vertical bars. For example <code>HOLD=Yes No Call</code> specifies that <code>Yes</code> or <code>No</code> or <code>Call</code> is valid.
Brackets [ ]	Brackets indicate that information is optional. For example, <code>STARTT=([date day][,hh:mm:ssXM])</code> indicates that you can specify either a date or a day, a date or a day plus a time, or just a time.
Italics	Italic letters are placeholders for information you must provide. Italic font also indicates book, chapter, and section titles and is used for emphasis in the text.
Monospaced characters (characters of equal width)	Monospaced characters represent information for screens, commands, Processes, and reports.
Punctuation	Code all commas and parentheses as they appear.



---

# Reviewing the Preinstallation Procedure

This chapter describes the information you should gather before you install the Connect:Direct for OpenVMS product.

---

## Reviewing Installation Configuration Options

The Connect:Direct for OpenVMS product can be installed in two different configurations:

- ❖ Server
- ❖ Client

A *server* installation provides a full and functional Connect:Direct system that includes the server, session manager, and user interface. A *client* installation provides only the user interface and relies on server functions, installed on some other OpenVMS node, to execute Connect:Direct functions. You can install either configuration in both clustered and nonclustered OpenVMS environments.

Throughout the installation, you must provide information based on the type of installation and environment you chose.

The hardware and software requirements for server and client installations are described in the *Connect:Direct for OpenVMS Release Notes* document.

---

## Reviewing the Preinstallation Steps

The following table lists the preinstallation steps. You must have a license key to configure Connect:Direct for OpenVMS and register the product before you can actually use it. Refer to the *Connect:Direct for OpenVMS Release Notes* for more information about obtaining a temporary or permanent license key.

Number	Description
Step 1.	Setting up the Connect:Direct for OpenVMS manager account. <b>Note:</b> Skip this step if installing the client software only.
Step 2.	Setting up disk quotas.

If you are upgrading an existing Connect:Direct installation, refer to Chapter 5, *Upgrading Connect:Direct for OpenVMS*, for additional considerations.

---

## Step 1: Setting Up the Connect:Direct for OpenVMS Manager Account

---

**Note:** Skip this step if you are installing the client software only.

---

Installation is usually done from the OpenVMS system manager account assigned the following privileges:

- ❖ TMPMBX
- ❖ NETMBX
- ❖ SYSLCK
- ❖ SYSPRV
- ❖ ALTPRI
- ❖ SYSGBL
- ❖ SYSNAM
- ❖ CMKRNL
- ❖ DETACH
- ❖ OPER
- ❖ BYPASS
- ❖ WORLD
- ❖ EXQUOTA
- ❖ IMPERSONATE

---

**Note:** The account selected for installation is referred to as *system* in examples in this manual.

---

Issue the following command to assign these privileges:

```
$ SET DEF SYS$SYSTEM
$ RUN AUTHORIZE
UAF> MODIFY system/PRIV=(TMPMBX, NETMBX, SYSLCK, -
_UAF> SYSPRV, ALTPRI, SYSGBL, SYSNAM, CMKRNL, DETACH, -
_UAF> OPER, BYPASS, WORLD, EXQUOTA, IMPERSONATE)
UAF> EXIT
```

---

## Step 2: Setting Up Disk Quotas

---

**Note:** If you are working on a system that does not have disk quotas enabled, skip this step.

---

Approximately 10,000 blocks of disk space are needed to install the Connect:Direct product. In order to have enough room for the work files (network map file, two statistics files, and two Transmission Control Queue files), at least 30,000 blocks of disk space are needed.

Use the OpenVMS SYSMAN utility to issue the ADD command to allocate additional disk space to the manager account:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA ADD system/DEV=$disk1/PERMQUOTA=30000-
_SYSMAN> /OVERDRAFT=1000
SYSMAN> EXIT
```

If disk quota has already been set, use the SHOW command to verify that the quotas are sufficient.

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA SHOW system/DEV=$disk1
%SYSMAN-I-QUOTA, disk quota statistics on device $DISK1: -
Node
  UIC          USAGE   Permanent Quota      Overdraft Limit
[manager]      107         30000                1000
SYSMAN> EXIT
```

To adjust the Connect:Direct account disk quotas, use the SYSMAN utility to issue the MODIFY command:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA MODIFY system/PERMQUOTA=30000-
_SYSMAN> /OVERDRAFT=1000/DEV=$disk1
%SYSMAN-I-QUOTA, disk quota statistics on device $DISK1: -
Node
  UIC          USAGE   Permanent Quota      Overdraft Limit
[manager]      107         30000                1000
```



---

# Installing Connect:Direct for OpenVMS

This chapter explains how to install the Connect:Direct for OpenVMS product using the OpenVMS install utility. A table is also provided that lists the required installation steps.

---

## Understanding the Types of Configurations

You can install or upgrade the product in two different configurations:

- ❖ Server
- ❖ Client

A *server* installation provides a full and functional Connect:Direct system that includes the server, session manager, and user interface. A *client* installation provides only the user interface and relies on server functions, installed on some other OpenVMS node, to execute Connect:Direct functions. You can install either configuration in both clustered and nonclustered OpenVMS environments.

Throughout the installation, you must provide information based on the type of installation and environment chosen.

Installing or upgrading the Connect:Direct product in a clustered environment is the same as for a single server node in a nonclustered environment. If you are upgrading, review the information in Chapter 5, *Upgrading Connect:Direct for OpenVMS*, before proceeding.

---

## Reviewing the Installation Steps

The following table lists the required steps for installing Connect:Direct for OpenVMS.

Name	Description
Step 1.	Download the product zip file from either the CD-ROM acquired from Sterling Commerce or the Sterling Electronic Software Distribution (ESD) Portal to a PC running the Windows operating system. Unzip and extract the Connect:Direct for OpenVMS image on the PC, and then transfer the executable image for the hardware and operating system you are using to the system where you will be installing Connect:Direct for OpenVMS. For complete instructions, see the <i>Connect:Direct for OpenVMS Release Notes</i> .
Step 2.	Logging in to the System Manager account.
Step 3.	Invoking VMSINSTAL.
Step 4.	Selecting installation options. Option 1 Selecting the Connect:Direct for OpenVMS target directory. Option 2 Purging old versions of the Connect:Direct for OpenVMS product. Option 3 Copying the Connect:Direct for OpenVMS help file into the OpenVMS help library. Option 4 Installing the NDMUI command.
Step 5.	Ending the installation.
Step 6.	Setting default directory to the Connect:Direct directory.
Step 7.	Verifying Connect:Direct for OpenVMS file installation.
Step 8.	Configuring the Connect:Direct for OpenVMS environment.

---

### Step 1: Downloading and transferring the product executable

The *Connect:Direct for OpenVMS Release Notes* document contains detailed instructions on how to download the product zip file and transfer the executable image appropriate to your OpenVMS system. See *Connect:Direct for OpenVMS Release Notes* for detailed step-by-step procedures.

---

### Step 2: Logging in to the System Manager account

Log in to the system with the username of system and the appropriate password:

- ❖ username: system
- ❖ Password: xxxxxxxx



---

## Step 3: Invoking VMSINSTAL

VMSINSTAL is an OpenVMS utility used to install software on the OpenVMS operating system. Installation of Connect:Direct files on the OpenVMS operating system is automated by using the VMSINSTAL interactive question-and-answer format, as illustrated in the remainder of this chapter.

You must choose whether to install a server and a client or to only install a client.

### Loading the Connect:Direct for OpenVMS Software

Load the software, and type the following command at the DCL prompt (\$):

```
$ @SYS$UPDATE:VMSINSTAL xxxxxxxnnn
```

A description of the parameters follows:

#### xxxxxxnnn

is the name of the product to install. The product name will change as new versions of the software are released. If you are installing a server and a client, the format is *NDMSRVnnn*; the format for a client-only installation is *NDMUIinnn*. In both cases, nnn indicates the version number. The product names, with the current version, are listed in the *Connect:Direct for OpenVMS Release Notes*.

### Responding to the Next Prompts

After verifying that the active processes will not be affected by the installation, answer **YES** to the next two prompts:

```
*      Do you want to continue anyway [NO]?
*      Are you satisfied with the backup of your system disk [YES]?
```

For the next two prompts, enter the device name and directory (or logical name) where the Connect:Direct files are to be located, and then press **Enter** to accept the default answer of none to the installation options prompt.

```
*      Where will the distribution volumes be mounted:
*      Enter installation options you wish to use (none):
```

### VMSINSTAL Begins the Installation

The next set of messages indicate that VMSINSTAL is beginning the installation:

```
The following products will be processed:
NDMxxx Vx.x
      Beginning installation of NDMxxx Vx.x at HH:MM
%VMSINSTAL-I-RESTORE, restoring product save set A . . .
%NDMxxx-I-KITENTRD, NDM-VMS KITINSTAL.COM entered.
```

---

## Step 4: Selecting Installation Options

You are prompted to supply information needed to install the Connect:Direct product. Defaults have been provided; use these or supply information as it applies to your environment.

### Option 1. Selecting the Connect:Direct for OpenVMS Target Directory

Enter the device name and directory (or logical name) where the Connect:Direct files are to be located. The default is `SYSS$SYSDEVICE:[NDM]`.

```
* Enter target directory for NDM-VMS files [SYSS$SYSDEVICE:[NDM]]:
```

If this directory does not exist, it is created at this time. (In examples in this manual, the directory is referred to as *ndmdir*.)

### Option 2. Purging Old Versions of the Connect:Direct for OpenVMS Product

To purge the existing files, enter **YES**.

If you select the purge option, then Connect:Direct files that currently exist are replaced by those in the new version. Refer to Chapter 5, *Upgrading Connect:Direct for OpenVMS*.

```
* Do you want to purge files replaced by this installation [YES]?
```

### Option 3. Copying the Connect:Direct for OpenVMS Help File into the OpenVMS Help Library

Enter **YES** if you want users to have access to the online Connect:Direct help facility:

```
* Place NDM help in SYS$HELP:HELPLIB.HLB [YES]?
```

### Option 4. Installing the NDMUI Command

Enter **YES** if you want the NDMUI command permanently installed in the system command table:

```
* Place NDMUI command in system command table [YES]?
%NDMVMS-I-SYSTBLCMD, Placing NDMUI command in system DCL tables.
```

---

**Note:** To restrict access, enter **NO**. Unless the following command is set up in the user's LOGIN.COM file, users must enter the command each time they log in:

```
$ SET COMMAND NDM$$DIRECTORY:NDMUI.CLD
```

---

## Upgrading to a New Release

If you are upgrading, the installation continues and the system asks if you want to keep the following configuration files:

```
* If updating an existing node, keep the existing CONFIGSRV.COM file [YES]?
* If updating an existing node, keep the existing INITPARMS.DAT file [YES]?
* If updating an existing node, keep the existing NDM$RPX_SETPARMS.COM file [YES]?
* If updating an existing node, keep the existing SERVER.DAT file [YES]?
* If updating an existing node, keep the existing CONFIGUI.COM file [YES]?
```

## Reviewing Cluster Considerations

If you are installing the Connect:Direct product in a cluster, VMSINSTAL only adds the command to the node executing the installation procedure. To make the command available to all nodes in the cluster, execute the following steps on each node:

1. Log into the member node.
2. Set the default to the directory where Connect:Direct software is installed.
3. Issue the following command to update the system command table:

```
$ SET COMMAND /TABLE=SYS$LIBRARY:DCLTABLES -
_.$ /OUTPUT=SYS$LIBRARY:DCLTABLES NDMUI.CLD
```

4. Issue the following command so that the NDMUI command is available to all users on the node:

```
$ INSTALL /REPLACE /OPEN /HEADER SHARED
_.$ SYS$LIBRARY:DCLTABLES.EXE
```

5. Log out and then log in so that the correct version of DCLTABLES.EXE can be accessed by your process.

---

## Step 5: Ending the Installation

VMSINSTAL copies the Connect:Direct files to the target directory. The following messages indicate that installation is near completion:

```
%NDMxxx-I-SPECFILE, Specifying UI files to be copied to target
directory
%NDMxxx-I-SPECFILE, Specifying SRV files to be copied to target
directory
%NDMxxx-I-KITEXIT, NDM-VMS KITINSTAL.COM exited.
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories. . .
```

The final messages displayed are as follows:

```
Installation of NDMxxx Vx.x completed at HH:MM
VMINSTAL procedure done at HH:MM
```

---

## Step 6: Setting Default Directory to the Connect:Direct Directory

Issue the following command to change your default to the Connect:Direct directory:

```
$ SET DEFAULT ndmdir
```

This directory is the same directory as specified in Step 4, Option 1. If you used the default, the directory name is SYSSYSDEVICE:[NDM].

---

## Step 7: Verifying Connect:Direct for OpenVMS File Installation

Before proceeding, verify that the following files (separated into client and server files) have been installed from the installation kit. If any files are missing, contact Sterling Commerce.

Use the DIR command to list the files.

### Reviewing Client Files

The following are client files:

#### **CONFIGUI.COM**

is the command procedure that sets up initialization logicals for the UI.

#### **COPYRIGHT.TXT**

is the file that contains the banner with the copyright information.

#### **MSGFILE.DAT**

is the indexed file that contains messages for all Connect:Direct applications.

#### **MSGFILE.FDL**

is the File Definition Language (FDL) text file that defines the Connect:Direct message file.

Although MSGFILE.FDL is not used during installation, it is included for reference.

#### **NDM\$CONFIG.COM**

is the command procedure used to configure the Connect:Direct product.

#### **NDM\$CONNECT.COM**

is the command procedure that can be used to connect to Connect:Direct installations that execute in an alternate environment.

#### **NDM\$DISCONNECT.COM**

is the command procedure that can be used to disconnect from Connect:Direct installations that execute in an alternate environment.

#### **NDM\$SERVERDEF.COM**

is executed during Connect:Direct startup to define remote servers.

**NDM.CBJ**

is the file that contains the internal protocol format.

**NDMMSG.EXE**

is the executable file that adds, deletes, and displays Connect:Direct messages.

**NDMUI.CLD**

defines NDMUI as a DCL command.

**NDMUI.EXE**

is the executable NDMUI image.

**NDMUI.HLB**

contains the online help for the Connect:Direct product.

**NDM\_EXAMPLES.TLB**

is the text library that contains samples of Connect:Direct Processes and DCL command procedures.

**NDM\_SHUTDOWN.COM**

stops the server.

**NDM\_STARTUP.COM**

runs the server as a detached process.

**SERVER.DAT**

is the file that contains the user-defined list of remote servers. This file is read by NDM\$SERVERDEF.COM.

## Reviewing Server Files

The following are server files:

**CKPTFILE.FDL**

is the FDL text file that creates the checkpoint file, CKPTFILE.DAT.

**CONFIGSRV.COM**

is the procedure that sets up initialization logicals for the server.

**INITPARMS.DAT**

is the file that specifies the parameters to be set upon Connect:Direct for OpenVMS startup.

**NDM\_BLD\_XLATE.EXE**

is the executable image that customizes the ASCII/EBCDIC translation tables.

**NDM\_DEFAULT.XLT**

is the default ASCII/EBCDIC translation table.

**NDM\_SAMPSRC.TLB**

is the text library containing sample source code for dataexits.

**NDM\_SMGR.EXE**

is the executable session manager image.

**NDM\_SRV.EXE**

is the executable server (controller) image.

**NDM\_TIMESTAMP.EXE**

is the executable image that places time stamps in the logfile NDM\_STARTUP.LOG.

**NDM\_TYPE.TLB**

is the text library containing FDL definitions for use in the TYPE parameter of the Connect:Direct for OpenVMS COPY statement. the username does not exist in the UAF or the password does not compare correctly, the security check fails. For a complete explanation of the COPY statement, see the Connect:Direct for OpenVMS Processes Web site at <http://www.sterlingcommerce.com/Documentation/processes/processhome.html>.

**NDM\_XLATE.TLB**

contains the default and example ASCII/EBCDIC translation tables.

**NETFILE.FDL**

is the FDL text file that creates the network map, NETFILE.DAT.

**SETPRIV.COM**

sets up the privileges needed to invoke the server.

**SETUPRIGHTS.COM**

defines the rights identifiers for Connect:Direct functions in the rights data base.

**TCQE.FDL**

is the FDL text file that defines the Transmission Control Queue (TCQ) step file.

**TCQH.FDL**

is the FDL text file that defines the TCQ header file.

---

## Step 8: Configuring the Connect:Direct for OpenVMS Environment

Configure the Connect:Direct execution environment. This step is described in the next chapter, *Configuring Connect:Direct for OpenVMS*.

---

# Configuring Connect:Direct for OpenVMS

After installing the Connect:Direct for OpenVMS software, you must configure the product as appropriate to your environment. A table that lists the required configuration steps is also provided.

---

## Reviewing the Types of Configurations

As discussed previously, the Connect:Direct for OpenVMS product can be installed in two different configurations:

- ❖ Server
- ❖ Client

---

## Reviewing the Configuration Steps

The following table lists the required steps for configuring Connect:Direct for OpenVMS.

Number	Description
Step 1.	Logging in to the System account.
† Step 2.	Invoking SETUPRIGHTS for the server.
Step 3.	Granting the NDM_ADMIN identifier to the Connect:Direct for OpenVMS account on the server node.
† Step 4.	Beginning the configuration process by supplying license key information.
Step 4A.	Configuring server installations.
Step 4B.	Configuring client-only installations.
† Step 5.	Editing the Network Map.
Step 6.	Testing the Connect:Direct for OpenVMS system.

†Skip this step if installing the client software only.

---

## Step 1: Logging In to the System Manager Account

---

**Note:** Ensure that the OpenVMS system manager performs the Connect:Direct for OpenVMS installation.

---

You do not have to use the system manager account to perform this step. However, the account must have the same quotas as the OpenVMS system manager account and the following privileges:

- ❖ TMPMBX
- ❖ NETMBX
- ❖ SYSLCK
- ❖ SYSPRV
- ❖ ALTPRI
- ❖ SYSNAM
- ❖ CMKRNL
- ❖ DETACH
- ❖ OPER
- ❖ SYSGBL
- ❖ BYPASS
- ❖ WORLD
- ❖ EXQUOTA
- ❖ IMPERSONATE

---

## Step 2: Invoking SETUPRIGHTS for the Server

---

**Note:** Skip this step if installing the CLIENT software only, and proceed to Step 4.

---

Invoke the `SETUPRIGHTS` command procedure to define rights identifiers for the various Connect:Direct functions *for the server node only*. Use the following command to invoke `SETUPRIGHTS`:

```
$ @ndmdir:SETUPRIGHTS.COM
```

**Note:** `SETUPRIGHTS.COM` updates the UAF and should be run only once. The definition of rights identifiers is optional.

---



---

### Step 3: Granting the NDM\_ADMIN Identifier to the Designated Connect:Direct for OpenVMS Manager Account on the Server Node

---

**Note:** Skip this step if installing the client software only, and proceed to Step 4B.

---

Grant the rights identifier NDM\_ADMIN to the Connect:Direct manager account used by the administrator. The rights identifier allows the administrator to issue any Connect:Direct command. Issue the following command to grant NDM\_ADMIN:

```
$ SET DEF SYS$SYSTEM
$ RUN AUTHORIZE
UAF> GRANT/IDENTIFIER NDM_ADMIN ndm_manager
UAF> EXIT
```

**Note:** If the rights identifier has already been granted, you will receive an error message indicating that it is a duplicate.

---



---

### Step 4: Configuring the Connect:Direct for OpenVMS environment

---

**Note:** Skip this step if installing the client software only, and proceed to Step 4B.

---

If you have not received the permanent license key file from Sterling Commerce, refer to the *Connect:Direct for OpenVMS Release Notes* for instructions on how to obtain this file. If you have received the license key file, copy it to your default Connect:Direct directory before proceeding.

You can enter the license key information using one of the following methods:

- ❖ Copying and pasting the license key string when prompted during the configuration procedure
- ❖ Before the configuration procedure begins, copying and pasting the license key string into a file called NDM\$KEY.TXT located in your default NDM\$\$DIRECTORY. If you use this method, Connect:Direct looks for the license key string in the NDM\$KEY.TXT file, and if it finds it and the license key is valid, continues executing the configuration utility.

To start the configuration procedure, return to your default Connect:Direct directory and type the following configuration command:

```
$ SET DEF ndmdir
@NDM$CONFIG.COM
```

The following prompt is displayed:

```
-----
                          NDM Configuration Utility
-----

Enter a question mark (?) at any time for help.

* Register Product Initialization Key [NO]? YES
```

If you have copied the license key string into a file called NDM\$KEY.TXT in your default Connect:Direct directory, type **NO** and press **Enter**. Connect:Direct validates the license key and the configuration procedure concludes.

To enter the license key string, press **Enter** to accept the default YES value. The following prompt is displayed:

```

Enter the NDM Product Installation Key.

      The Product Installation Key Is A Case-Sensitive
      String Of 60 Alphanumeric Characters. You May
      Input The Key Directly From The Keyboard Or
      Indirectly Through A File Which Contains The
      Product Installation Key.

      If No Characters Are Input, The Product Installation
      Key Is Assumed To Be Given By The First Record In The
      NDM$SDIRECTORY:NDM$KEY.TXT File.

KEY>

```

Open the license key file, copy the text in the line containing the LicenseKey encrypted value, and paste it in the KEY> prompt.

---

**Note:** The license key looks like:

```
89qvyaV$a7Fgkpdj79TaMARfbE@2Ipm038GD@Uj0$Rw8a97mSTjzfhehgwrZ
```

Be very careful when you copy the key and make sure no carriage returns are inserted or lines truncated, which will invalidate the key file.

---

The following prompt is displayed:

```
* Is this information correct [NO]? YES
```

Press the **Enter** key to continue.

---

## Step 4A: Configuring a Server/Client Installation

There are two types of environments within which the Connect:Direct product can be configured. The first is a system-wide environment, configured so the product is known to all users. A system-wide environment type of configuration is detailed in this chapter.

The second type of environment, an alternate environment, is most commonly used during a product upgrade. When the Connect:Direct product is installed in an alternate environment, the alternate environment, as well as its logical name tables and user interface, are not automatically known to all users. Users wanting to use this Connect:Direct system must specifically request a connection to it. For more information on defining an alternate environment, refer to *Installing Connect:Direct for OpenVMS Software in an Alternate Environment* on page 38.

```
* Enter the NDM environment to configure [NDM]:
```

After specifying your system-wide environment, the procedure verifies your selection as follows:

```
* Is the system-wide environment NDM correct [NO]?
```

A server must *not* be executing in the environment you have selected. A server can be running if you are continuing from a previously aborted configuration attempt or reconfiguring the product with different operational parameters. If a server is running, you must enter *NO* in response to the following prompt:

```
* Should this procedure continue [NO]?
```

There might be messages indicating that the configuration process is creating various data files. This is customary with new installations. The following prompt requests the Connect:Direct node name for this system:

```
* Enter the NDM node name for this system  
[DECnet_node.VMS.ndm_environment]:
```

Unless you are upgrading or reconfiguring an existing Connect:Direct installation, the procedure executes to completion from this point unassisted. If you are upgrading or reconfiguring your system, refer to Chapter 5, *Upgrading Connect:Direct for OpenVMS*.

When NDM\$CONFIG.COM has completed execution, the Connect:Direct node is capable of performing local (PNODE=SNODE) data transfers. This allows you to functionally test the product without requiring access to remote nodes. In addition, NDM\$CONFIG.COM creates command procedures for both product initialization and shutdown.

The command file informs you of the file names and directories of these procedures, and the following message is displayed on the screen:

```
NDM Configuration is complete.
```

The Connect:Direct product is now installed on your system. You can set up additional remote servers at this time, if necessary. This procedure is described in *Defining Remote Connect:Direct for OpenVMS Servers* on page 31. Descriptions of starting and stopping the Connect:Direct product begin on page 31. To start and stop Connect:Direct for OpenVMS in a cluster environment, see *Reviewing Cluster Considerations for Starting Connect:Direct for OpenVMS (Server)* on page 32 and *Reviewing Cluster Considerations for Stopping Connect:Direct for OpenVMS (Server)* on page 32.

## Defining Connect:Direct for OpenVMS Initialization Parameters

Initialization parameters are processed during Connect:Direct startup and provide information for the server. The parameters have default values that you can override by including the parameter in NDM\$\$DIRECTORY:INITPARMS.DAT. The following is an example.

```

ndm$$detach_process_monitor_enable  true
ndm$$rpx_translate_runtask_enable   yes
!
ndm$$srv_timestamp                   60 !minutes
ndm$$inactivity_timeout              5
!NDM$$CKPT_INTERVAL                 500k
NDM$$WTRETRIES                       "00:02:00" !2minutes
NDM$$RETRY_LIMIT                     10 ! times

Ndm$$stat_shutdown                  yes
!Proxy: ! indicates comments!!!!!!!
ndm$$proxy_file_enable               enable
ndm$$proxy_file                     disk$axp:[mchar3.qa32]ndm_proxy.dat
ndm$$proxy_file_system               enable
ndm$$proxy_vms_system_file_enable    yes

!tcpip:
ndm$$tcp_listenport                  6677
NDM$$TCP_RU_SIZE                     16384
NDM$$TCP_LISTENER_COUNT              1
NDM$$TCP_ACTIVE_CONNECT_ADDRESS_DISPLAY_ENABLE  ndmuser

```

The following table provides a description of all initialization parameters in alphabetical order.

Initialization Parameter	Default	Description
NDM\$\$CKPT_DAYS	8	Number of days that checkpointing information is retained after a data transfer is interrupted.
NDM\$\$CKPT_INTERVAL	none	If defined, specifies the default checkpoint interval. If this parameter is not defined, then checkpointing is only active when an interval is specifically stated in a COPY statement. Valid values are nM (megabytes) and nK (kilobytes).
NDM\$\$COMPRESS_R2R	Y	Enables compression algorithm to be used during transmit and receive operations.
NDM\$\$COMPRESS_R2X	Y	Enables compression algorithm to be checked and validated.
NDM\$\$CPXI_ENABLE	Y	Enables a cluster setup. Add this initialization parameter to the initparms file of the primary server in the cluster.
NDM\$\$DETANDNDM\$\$DETACH_PROCESS_MONITOR_ENABLE	TRUE	Specifies to execute Run Tasks as detached Processes.
NDM\$\$DIROWN	1	Specifies file ownership. NDM\$\$DIROWN=1 will make all transfers default with directory ownership. NDM\$\$DIROWN=0 will not allow directory ownership.

Initialization Parameter	Default	Description
NDM\$\$DISSETPROT	1	Specifies protection/ownership. If NDM\$\$DISSETPROT is defined as 1, Connect:Direct for OpenVMS will not attempt to set ownership on the file after it has been copied. If it is not defined as 1, Connect:Direct for OpenVMS will attempt to set ownership as it normally does unless an override is specified in the SYSOPTS of the COPY statement.
NDM\$\$FILE_ACCESS_RETRY_REMOTE_NODE_DBX	NDM\$FILE_ACCESS_RETRY_REMOTE_NODE.DAT	File allocation retry is specified based on the destination dataset being inaccessible at time of copy operation.
NDM\$\$PASSWORD_FILE	NDM\$\$ DIRECTORY :NDM\$ PASSWORD. DAT	Enables username/password to be represented by an identifier.
NDM\$\$PERMITTED_NEIGHBOR_TCP_CALL_DISPLAY_ENABLE	Y	For Diagnostic use.
NDM\$\$PROXY_FILE	none	Defines Connect:Direct proxy filename by means of this logical name. If this name is not defined, Connect:Direct assumes the proxy filename to be: NDM\$\$DIRECTORY:NDM_PROXY.DAT
NDM\$\$PROXY_FILE_ENABLE	none	Enables the use of the Connect:Direct proxy file. If this name is not defined, Connect:Direct produces a proxy file in the original manner using the VMS authorize facility.
NDM\$\$PROXY_FILE_SYSTEM	none	Specifies the actions to be taken if no entry is found in the Connect:Direct proxy file. When the name is defined, Connect:Direct uses the authorize utility to produce a proxy file that is processed as in previous Connect:Direct versions. When the name is not defined, Connect:Direct returns a status of <i>nosuchuser</i> and the proxy login fails.
NDM\$\$PROXY_FILE_TYPE_SX	Y	Enables the use of an encrypted proxy file for security access.
NDM\$\$PROXY_VMS_SYSTEM_FILE_ENABLE	Y	Specifies that VMS NETUAF proxies are in use.
NDM\$\$RETRY_LIMIT	99999	Sets the number of times a process will retry before terminating. This parameter may be overridden on a per-submission basis by the /RETRY_LIMIT value optionally specified when a Connect:Direct for OpenVMS script is submitted for execution. The minimum is 0; the maximum is 99999.
NDM\$\$RPX_TRANSLATE_RUN_TASK_ENABLE	Y	Enables RUN TASKS to execute as detached Processes.
NDM\$\$SM_PRI	4	Specifies the OpenVMS priority to be assigned to each session manager (running as a subprocess to the server). The normal OpenVMS interactive priority is 4.
NDM\$\$SRV_PRI	4	Specifies the OpenVMS priority to be assigned to the server (running as a OpenVMS detached process). The normal OpenVMS interactive priority is 4.
NDM\$\$SRV_TIMESTAMP	0	Specifies the interval in minutes between timestamps in the server log file and is a diagnostic tool. If 0 is specified, timestamps will not be placed in the server log.

Initialization Parameter	Default	Description
NDM\$\$STAT_DIRECTORY	Server Process startup directory	This name defines an alternate directory in which the statistics files are to reside. The default is the Connect:Direct for OpenVMS server process startup directory given by NDM\$\$DIRECTORY.
NDM\$\$STAT_DISABLE	TRUE	Enables access to archived statistics data to be recovered.
NDM\$\$STAT_INITIALIZE	none	If defined, initializes the statistics file at startup of Connect:Direct for OpenVMS.
NDM\$\$STAT_MAX_VERSION	3	This name defines the maximum number of versions of the statistics files that are to be retained.
NDM\$\$STAT_MAX_BLKSIZE	45000 blocks	This name defines the maximum number of content data blocks. When the content of the statistics files reach this threshold, a new version is begun and the oldest version is then deleted. The default value is 45000 blocks and there is no maximum value. The minimum value is 32 blocks.
NDM\$\$STAT_PROCESS_PRIORITY	4	This name defines the Connect:Direct for OpenVMS statistics process priority. The process priority ranges from 0 to 31. The default priority is 4.
NDM\$\$STAT_MAX_MESSAGE_RX_QSIZE	24	This name defines the length of the Connect:Direct for OpenVMS statistics process receive message queue and specifies the maximum number of buffered statistics messages. The default value is 24 and the maximum value is 31. The minimum value is 12. Note that there is no limit to the number of outstanding messages. This parameter, together with the statistics process priority, can be used to increase Connect:Direct for OpenVMS throughput.
NDM\$\$STAT_SHUTDOWN	YES	Disables statistics recording for improved performance.
NDM\$\$TCP_LISTENER_PORT	1364	Specifies the port number on which the local Connect:Direct for OpenVMS will listen for TCP/IP connection requests. This name supersedes NDM\$\$TCPLISTENPORT, which is still valid.
NDM\$\$TCP_LISTENER_COUNT	none	When TCP/IP is enabled, this is the number of permanent TCP listeners to bring up when the server starts. The minimum is 1; the maximum is 128.
NDM\$\$TCP_RU_SIZE	4096	Defines RU size for TCP/IP transmissions. The minimum is 2048; the maximum is 16,384.
NDM\$\$TCP_SESSION_CTRL_SEND_CC	TRUE	Enables copy termination records to be exchanged.
NDM\$\$TCP_SESSION_CTRL_SET_XNODE_INDICATOR	TRUE	When set, ensures that copy termination record information is exchanged with the transfer node.
NDM\$\$TCP_SESSION_PRIORITY	6	Priority of all TCP session managers. The minimum is 1; the maximum is 31.
NDM\$\$TCP_SESSION_SOURCE_PORT_DBX	NDM\$TCP_SESSION_SOURCE_PORT_DBX.DAT	Enables firewall navigation and network map checking.

Initialization Parameter	Default	Description
NDM\$\$TCQ	WARM	Specifies whether Connect:Direct will continue scheduling work previously queued in case a system interruption, such as a power failure, occurs. If you do not want Connect:Direct to automatically continue after a system interruption, set the value to COLD.
NDM\$\$WTRETRIES	"00:03: 00"	Sets the wait time between successive retries. It is expressed in the hh:mm:ss format. The default value is three minutes ("00:03:00"). While it is possible to have a value of zero seconds, (00:00:00) this is not recommended. Typically the minimum wait time should not be less than one minute (00:01:00). The minimum is 00:00:00; the maximum is 23:59:59. The value should be specified with double quotes.

## Defining Remote Connect:Direct for OpenVMS Servers

You can access remote Connect:Direct servers through the user interface, NDMUI, assuming that those servers have been defined locally. The datafile, SERVER.DAT, contains remote server definitions. Each time the product is initialized, this data file is read and the remote servers are made known to Connect:Direct. To add a remote server definition, you must have the following information:

- ❖ Alias name by which the remote server is to be referenced. The alias name is assigned by the installer, to be used as a nickname for the node. The /SERVER parameter in various Connect:Direct commands refers to the value in the alias name.
- ❖ DECnet node name of the system where the remote server resides.
- ❖ Name of the environment within which the remote server executes.

The following figure shows a sample SERVER.DAT file, in which a server on the DECnet node, DALL01, is executing in the Connect:Direct environment, TEST, with an alias of DALLAS.

```

*****
-----
USER-DEFINED REMOTE SERVER DEFINITIONS ARE ADDED TO THE
FOLLOWING LIST:
-----
*****
Server Alias  Server Node      Server Environment
DALLAS        DALL01             TEST

```

One or more spaces must separate each field. Column headings are provided in SERVER.DAT for clarity.

## Starting Connect:Direct for OpenVMS (Server)

To start the Connect:Direct product in a standard system-wide environment, execute the following command procedure:

```
$ @SYS$STARTUP:NDM$STARTUP_SERVER.COM
```

The procedure was created by the configuration utility, NDM\$CONFIG.COM, and resides in the SYS\$STARTUP directory.

---

**Note:** You can configure the Connect:Direct product to run in an alternate environment. For instructions on starting the product in an alternate environment, refer to *Installing Connect:Direct for OpenVMS Software in an Alternate Environment* on page 38.

---

## Stopping Connect:Direct for OpenVMS (Server)

To stop the Connect:Direct product in a standard system-wide environment, execute the following command procedure:

```
$ @SYS$MANAGER:NDM$SHUTDOWN_SERVER.COM
```

The procedure was created by the configuration utility NDM\$CONFIG.COM, and resides in the SYS\$MANAGER directory.

## Reviewing Cluster Considerations for Starting Connect:Direct for OpenVMS (Server)

To start the Connect:Direct product in a cluster, designate a single node as the server node. The selected node must have access to communication devices and support software for the required interfaces.

Add the following line to your system manager startup command procedure executed by the server node:

```
$ @SYS$STARTUP:NDM$STARTUP_SERVER.COM
```

Add the following line to the system manager startup command procedures on all other nodes in the cluster that require access to the user interface (NDMUI):

```
$ @SYS$STARTUP:NDM$STARTUP_CLIENT.COM
```

---

**Note:** Before you execute the startup commands, make sure that the NDM\$CPXI\_ENABLE initialization parameter has been added to the initparms file of the primary server in the cluster. See *Defining Connect:Direct for OpenVMS Initialization Parameters* on page 28.

---

## Reviewing Cluster Considerations for Stopping Connect:Direct for OpenVMS (Server)

---

**Note:** The Connect:Direct product can also be configured to run in an alternate environment. For instructions on starting the product in an alternate environment, refer to *Chapter 5, Upgrading Connect:Direct for OpenVMS*.

---



To shut down the Connect:Direct system in a cluster, execute the following:

1. SY\$MANAGER:NDM\$SHUTDOWN\_SERVER.COM command procedure to shutdown the Connect:Direct system on the server node
2. SY\$MANAGER:NDM\$SHUTDOWN\_CLIENT.COM command procedure to shutdown the Connect:Direct system on cluster nodes only executing the UI

---

## Step 4B: Configuring Client-Only Installations

Complete the following steps for a client-only installation.

You are first prompted for the DECnet node name assigned to the system on which the remote server resides.

```
* Default NDM Server DECnet node name?
```

Once the DECnet node name has been entered, you will be prompted for the environment name assigned when the remote SERVER installation was configured.

```
* Default NDM Server environment name [NDM]?
```

Because most servers are installed system-wide, the default response is *NDM*.

## Starting Connect:Direct for OpenVMS (Client Only)

Start the Connect:Direct product in a standard system-wide environment, executing the following command:

```
$ @SYS$STARTUP:NDM$STARTUP_CLIENT.COM
```

The procedure was created by the configuration utility NDM\$CONFIG.COM, and resides in the SY\$STARTUP directory.

---

**Note:** You can configure the Connect:Direct product to run in an alternate environment. For instructions on starting the product in an alternate environment, refer to *Installing Connect:Direct for OpenVMS Software in an Alternate Environment* on page 38.

---

## Stopping Connect:Direct for OpenVMS (Client Only)

To stop the Connect:Direct product in a standard system-wide environment, execute the following command procedure:

```
$ @SYS$MANAGER:NDM$SHUTDOWN_CLIENT.COM
```

The procedure was created by the configuration utility, NDM\$CONFIG.COM, and resides in the SYS\$MANAGER directory.

---

**Note:** You can configure the Connect:Direct product to run in an alternate environment. For instructions on starting the product in an alternate environment, refer to *Installing Connect:Direct for OpenVMS Software in an Alternate Environment* on page 38.

---

## Step 5: Editing the Network Map

---

**Note:** Skip this step if installing the client software only.

---

The command procedure, NETDEF.COM, is used to define nodes on the network known to the Connect:Direct system. NDM\$CONFIG.COM creates this file with your local node definition already in place. The Connect:Direct system refers, by default, to your local node as DECnet\_node.VMS.ndm\_environment. For example, the local DECnet node, DALL01, executing in the standard environment, NDM, would have the local node name of DALL01.VMS.NDM.

To define remote nodes, you must add additional records to the file. For a description of network map commands that you can use, refer to Chapter 8, *Modifying the Network Map*.

## Step 6: Testing the Connect:Direct for OpenVMS System

Use an editor to create and save the Process shown in the following figure as a file named COPY.NDM. Create and save a file named VMSFILE.SND to use as your test file. You can copy the Process as your test file.

```

COPY PROCESS SNODE=remote.node
STEP01  COPY  FROM  (DSN=cddir:VMSFILE.SND PNODE)      -
          TO    (DSN=remote.file SNODE)
STEP02  COPY  FROM  (DSN=remote.file SNODE)          -
          TO    (DSN=cddir:VMSFILE.RCV PNODE)

```

STEP01 of the Process in the previous figure is coded to copy VMSFILE.SND to another Connect:Direct node. STEP02 is coded to copy the same file back to the primary node (PNODE). Substitute the appropriate value for *remote.node*, which is the name of the remote node as specified in the network map. Substitute the value for *cddir*, which is the directory. Substitute the value for *remote.file*, which is the name of the file to which you are copying.

The file received at the PNODE is VMSFILE.RCV. By comparing the two files (VMSFILE.SND and VMSFILE.RCV), you can confirm that the Connect:Direct system is working correctly.

Using NDMUI, issue the SUBMIT command to submit the Process. The Connect:Direct for OpenVMS product must be running.

```

$ NDMUI
Connect:Direct> SUBMIT COPY
VSRV101I: Feedback: 0 Reply: 0 Function: dtf_submit
Process submitted successfully. Process number : 1
Connect:Direct>

```

Issue the `SHOW STATISTICS` command to check the status of the Process. The qualifier `/PNUM=1` identifies the Process as Process number **1**. You can use the `/DETAIL` qualifier for an expanded display that describes the status of the Process.

```
Connect:Direct> SHOW STATISTICS /PNUM=1
```



---

# Upgrading Connect:Direct for OpenVMS

To upgrade and use a new release, you must do a complete installation. This chapter discusses the two methods you can use to upgrade:

- ❖ Directly replacing the existing Connect:Direct for OpenVMS software and replacing or keeping your existing configuration and data files
- ❖ Installing the Connect:Direct for OpenVMS software in an alternate environment and replacing or keeping your existing configuration and data files

---

## Planning for an Upgrade of Connect:Direct for OpenVMS

Performing an upgrade is similar to performing an initial installation. For an upgrade, you may choose to retain some or all of the following configuration and data files that were created by the initial installation or as a result of the last upgrade. Default file names are shown in parentheses.

- ❖ NDM proxy file (ndm\_proxy.dat)
- ❖ Network map file (netfile.dat)
- ❖ Initialization parameter file (initparms.dat)
- ❖ License key file (ndm\$key.txt)
- ❖ ndm\$rp\*\_setparams.com file
- ❖ ndm\_startup\$setparams.com
- ❖ ndm\$event\_proc\*\_setparms.com
- ❖ Any user-defined translation table files
- ❖ configsrv.com
- ❖ server.dat
- ❖ configui.com

---

## Directly Replacing the Existing Connect:Direct for OpenVMS Software

---

**Caution:** It is recommended that you install Connect:Direct for OpenVMS in an alternate environment so that if it becomes necessary to revert back to an earlier version, you can easily do so.

---

Direct replacement of existing Connect:Direct software requires that the Connect:Direct system be shut down during the upgrade process. After you have downloaded the product executable and shut down the Connect:Direct system, you can proceed with the installation process starting with *Step 3: Invoking VMSINSTAL* on page 17. It is in the last part of the installation phase after the NDMUI command has been installed in the system command table that an upgrade varies from an initial installation. For each existing configuration file, you are prompted one by one if you want to keep it, and then after that series of questions, the upgrade continues in the same manner as a new installation. For details, see *Upgrading to a New Release* on page 19.

If you choose to reconfigure and write over existing files, use the configuration utility to create new empty configuration and data files. For details, see *Step 4: Configuring the Connect:Direct for OpenVMS environment* on page 25.

---

## Installing Connect:Direct for OpenVMS Software in an Alternate Environment

Install the Connect:Direct software in an alternate environment for access to a new release without disturbing the current production installation or to test the new software prior to executing it system-wide.

To install the software in an alternate environment, do the following steps:

1. Run the VMSINSTAL command procedure as you would for an initial installation. When prompted for a target directory, specify the alternate directory to which you want to install.
2. When NDM\$CONFIG.COM prompts you for an environment name, specify an environment other than *NDM*. The name must be 3 characters or less.
3. The Connect:Direct product is then installed into the alternate environment as if it was an initial installation. Upon completion of installation and configuration, add the network map information from the data files of the production version to the network map data for the alternate environment.
4. At your convenience, you can replace the production version with the newer and fully configured software in your alternate environment.

---

**Note:** If you do not preserve the original NETDEF.COM file when you move your alternate environment Connect:Direct into production, you must edit the new version to correct the LOCAL.NODE name. The local node will be named with the alternate environment (DECnet\_node.VMS.ndm\_environment) and must be changed to reflect the production environment.

---

## Starting the Connect:Direct for OpenVMS System in an Alternate Environment

To start the Connect:Direct system in an alternate environment, execute the following procedure:

```
$ @ndmdir:NDM$STARTUP_SERVER_environment.COM
```

Parameter descriptions follow:

***ndmdir***

is the name of the directory where Connect:Direct resides.

***environment***

is the Connect:Direct environment name selected during configuration.

The procedure was created by the configuration utility NDM\$CONFIG.COM.

## Cluster Considerations for Starting Connect:Direct for OpenVMS (Server)

To start the Connect:Direct product in an alternate environment in a cluster, designate a single node in the cluster as the server node. The selected node must have access to communication devices and support software for the required interfaces.

Execute the following procedure:

```
$ @ndmdir:NDM$STARTUP_SERVER_environment.COM
```

All other nodes in the cluster that require access to the user interface must execute the following:

```
$ @ndmdir:NDM$STARTUP_CLIENT_environment.COM
```

## Stopping the Connect:Direct for OpenVMS System in an Alternate Environment

To stop the Connect:Direct system in an alternate environment, execute the following procedure:

```
$ @ndmdir:NDM$SHUTDOWN_SERVER_environment.COM
```

The procedure was created by the configuration utility NDM\$CONFIG.COM.

## Cluster Considerations

To shut down the Connect:Direct system in an alternate environment in a cluster, execute the following command procedure on the node performing the server functions:

```
$ @ndmdir:NDM$SHUTDOWN_SERVER_environment.COM
```

Execute the following command procedure on all cluster nodes that the NDM\$STARTUP\_CLIENT\_environment.COM procedure was run on.

```
$ @ndmdir:NDM$SHUTDOWN_CLIENT_environment.COM
```

## Connecting to the Connect:Direct for OpenVMS System in an Alternate Environment

When you configure the Connect:Direct software in an alternate environment, it is not automatically known to system users. To use a Connect:Direct system that is configured in an alternate environment, execute the command procedure *ndmdir:NDM\$CONNECT\_environment.COM*. The command procedure was created by NDM\$CONFIG.COM.

Disconnect by executing *ndmdir:NDM\$DISCONNECT.COM*. This feature allows you to switch between a production version and the version in an alternate environment.





---

# Maintaining Security

This chapter discusses the following topics:

- ❖ Connect:Direct security
- ❖ Establishing permitted neighbor support
- ❖ Managing remote access control with a password file

---

## Understanding Connect:Direct for OpenVMS Security

Security for the Connect:Direct system is maintained by using standard OpenVMS facilities. Protection based on a user identification code (UIC) or Access Control Lists (ACL) associated with a particular OpenVMS username is honored by Connect:Direct.

The ability to perform Connect:Direct functions also can be permitted or denied by granting the corresponding defined rights identifiers to individual users.

The following paragraphs describe how the Connect:Direct system handles security checking in various situations.

## Signing On to the Connect:Direct for OpenVMS System

Login processing is automatically performed once the first NDMUI command is issued for a particular NDMUI. It is also performed when `/SERVER=server_alias` is specified. Validation of the right of an OpenVMS user to sign on to a particular Connect:Direct system (including the local system) is performed by the server using the OpenVMS username that the user entered at LOGIN (supplied by the NDMUI). If the node is different from the one running the NDMUI, the username is treated as a proxy id on the server node and is processed in a way similar to a proxy login initiated by DECnet.

## Understanding Security Considerations

If the User Interface (UI) is running on the same node as the server, the User Authorization File (UAF) is checked to verify that the username exists locally. However, no password verification is performed.

To process a login request from a remote node, the server searches for a proxy entry in the Connect:Direct for OpenVMS proxy file or the system proxy file.

## Checking User Interface Security

The UI verifies that users are authorized to issue a Connect:Direct command by checking to see if that authority is returned in the information received from the server at login time.

## Checking Security for Connect:Direct for OpenVMS Processes

Security checking during Process execution is based on a username in the Process submitter field and information in either the PNODEID or SNODEID fields.

If Connect:Direct for OpenVMS is executing the Process, it uses the submitter username (OpenVMS account username) and any PNODEID information for security checking. If the remote node is executing the Process, Connect:Direct for OpenVMS uses the submitter username and any SNODEID information for security checking.

The Connect:Direct for OpenVMS system validates the security username for the Process in one of two ways:

- ❖ Verifying the username and password in the UAF
- ❖ Verifying that a node and username combination for the Process submitter exists in the Connect:Direct for OpenVMS proxy database or the system proxy database in a way similar to a DECnet proxy login

Whenever an OpenVMS userid and password are specified explicitly in the PNODEID or SNODEID parameters of a Process, the password is encrypted and compared with the password field in the UAF of the specified OpenVMS username. If the username does not exist in the UAF or the password does not compare correctly, the security check fails. For a complete explanation of the PROCESS statement, see the Connect:Direct for OpenVMS Processes Web site at <http://www.sterlingcommerce.com/Documentation/processes/processhome.html>.

The second validation method uses proxies and has the advantage that passwords are not required in the Process. The Process username is assumed to be an OpenVMS proxy id. Users can optionally specify a real OpenVMS username in the PNODEID or SNODEID parameter, which is used as the security username if the proxy id processing is successful.

Benefits of using a proxy are as follows:

- ❖ A proxy does not reveal a user password on the communications link.
- ❖ Connect:Direct protocol only allows 8-character usernames or passwords for PNODEID and SNODEID. OpenVMS usernames and passwords longer than 8 characters cannot be specified.

---

**Note:** OpenVMS usernames and passwords may be greater than 8 characters on TCP/IP connections *only*.

---

The following is an example of an AUTHORIZE command showing how the proxy database records can be viewed:

```
UAF> SHOW/PROXY "rmt_ndm_node::rmt_ndm_user"
```

Parameters are described as follows:

***rmt\_ndm\_node***

is the remote node as defined in the ADJACENT.NODE record of the Connect:Direct for OpenVMS network map.

***rmt\_ndm\_user***

is the remote user id.

## Understanding Default Proxies

The Connect:Direct system uses proxy definitions to extract the *real* OpenVMS username to be used for security purposes. An example follows:

```

UAF> SHOW/PROXY "SC.MVS.USER1::USER1"

Default proxies are flagged with (D)

"SC.MVS.USER1::USER1"
    USER1 (D)                USER2
UAF>

```

If you specify an OpenVMS username in the PNODEID or SNODEID parameters without specifying a password, that username is used as the security username if it is in the list of usernames for the proxy entry. If no username is specified in the PNODEID or SNODEID parameters, the default username in the proxy entry is selected as the security id. If no default is indicated in the entry, the first username in the list is selected as the security id.

## Creating a Proxy Entry

The following command shows how to create a proxy entry:

```

$ SET DEF SYS$SYSTEM
$ RUN AUTHORIZE
UAF>ADD/PROXY "rmt_ndm_node::rmt_ndm_user" local_user
UAF>EXIT

```

Parameters are described as follows:

### **rmt\_ndm\_node**

is the name of the remote node (the SNODE).

### **rmt\_ndm\_user**

is the security user username from the SNODE.

---

**Note:** The parameters *rmt\_ndm\_node* and *rmt\_ndm\_user* can be 1 to 31 alphanumeric characters long. The entire node name and username combination must be in double quotes if any of the following characters are in either the node name or username:

```
@^#()-+=[:!\"'<>,/?.
```

*A period is acceptable in a node name but not in a username.*

---

### **local\_user**

is the local OpenVMS username.

Issue the following commands to set up a proxy for a CLIENT software node to connect to a server node:

```

UAF>ADD/PROXY "CLIENT_NDM_NODE::CLIENT_USER" -
_UAF> SERVER_USER /DEFAULT
UAF>ADD/PROXY "SERVER_NDM_NODE::CLIENT_USER" -
_UAF> SERVER_USER /DEFAULT

```

## Using Connect:Direct for OpenVMS Proxy File

The NDM\$\$PROXY\_FILE\_ENABLE, NDM\$\$PROXY\_FILE, and NDM\$\$PROXY\_FILE\_SYSTEM initialization parameters allow you to use a special Connect:Direct proxy file instead of OpenVMS proxies. Refer to *Defining Connect:Direct for OpenVMS Initialization Parameters* on page 28 for the description of the initialization parameters.

## Reviewing the File Format

The Connect:Direct proxy file is a standard sequential text file with the following format:

```

!
! ----- The Standard Model
!
!   node::remote_user      local_user
!
! ----- Allow Access From MVS
!
!       ss.mvs.emccal::emccal      emccal
!
! ----- An Unusual Username
!
!       ss.mvs.emccal::@who@      what
!
! ----- Wildcard Proxy Forms
!
!

```

The exclamation (!) character is used as a tail comment introducer and can be used anywhere. Proxy entries are not case sensitive, and any characters can be used with the exception of embedded spaces. Quotation marks as found in the system network proxy file are not necessary. Do not specify quotation marks unless they are present in the proxy item fields.

## Using Proxy Wildcards

You can use wildcard entries in the Connect:Direct for OpenVMS proxy file. These forms follow the standard OpenVMS proxy format conventions. The following are examples of using proxy wildcards:

```
remote_node::remote_userlocal_user
```

```
remote_node::remote_user*
```

```
remote_node::*local_user
```

```
remote_node::*
```

```
*::remote_userlocal_user
```

```
*::remote_user*
```

```
*:* local_user
```

```
*:* *
```

---

**Note:** If `local_user` is a wildcard, then it defaults to the `remote_user` value.

---

The following is an example of a proxy file that uses wildcards.

```
CD41::MCHARITAR2@csg.stercomm.com    MCHARx
MC35::MCHAR1                          emccax
mc35::mchar1                          MCHARx
*::aanth1                              mcharx
*::Jrackler                            MCHARx
MCHAR2.NT33::mchar2                   mCHARx
aixpaul::cdpaul                       Mcharx
11AIX1.CD.V3304::*                    MCHARx
csd.mvs.level1::JROGE1                mcharx
CSD.OS390.LEVEL1::*                   mcharx
CSD.OS390.LEVEL1::MCHAR3              mcharx
*::durham                              mcharx
*::MBENS2                              mcharx
SL3100::*                              MCHARx
PML3300::*                             MCHARx
PML3100:MCHAR1                        MCHARx
```

---

## Establishing Permitted Neighbor Support

Permitted neighbor support allows a local node to verify an inbound session from an adjacent remote node. This facility is a superset of the Network Map Checking feature found on other Connect:Direct platforms. You can optionally use a permitted neighbor database to provide auditability and security to ensure the availability of resources for critical applications at specified times.

A permitted neighbor database is a table where each entry is an adjacent node name. The times that access is permitted to the local node are also established in the database.

With permitted neighbor support, you can also prevent the execution of a Connect:Direct Submit/RunJob/RunTask function from certain adjacent nodes. The specified function is disallowed to all users on the specified nodes without modifying specific accounts or the user authorization file.

## Creating a Permitted Neighbor Database

Create a permitted neighbor database by running the utility program `ndm_permitted_neighbor_dbx.exe`. This program reads an input source definition file. The resulting output is searched at the time of an inbound session.

The utility reads the source definition file, `ndm$permitted_neighbor_dbx.src`, and creates an output file, `ndm$permitted_neighbor_dbx.dat`, which you can rename. Execute the utility by entering the following command:

```
run ndm_permitted_neighbor_dbx
```

## Activating a Permitted Neighbor Database

Activate the permitted neighbor database by specifying the following logical names:

### **NDM\$\$PERMITTED\_NEIGHBOR\_DBX**

specifies the filename of the permitted neighbor database used to validate the node name of the requesting node.

### **NDM\$\$PERMITTED\_NEIGHBOR\_DISABLE**

disables all inbound sessions and can be defined or deassigned at any time. Although this function can be achieved by using an empty permitted neighbor database, using this logical name can be more convenient.

The value of this logical name is irrelevant. It can be defined externally or as an initial parameter and can be defined or deassigned at any time. When specified, it overrides NDM\$\$PERMITTED\_NEIGHBOR\_DBX. Both logical names can specify the same file. However, these logical names will generally specify different files and there are no restrictions regarding filenames.

### **NDM\$\$PERMITTED\_NEIGHBOR\_RUN\_DBX**

specifies the filename of the permitted neighbor database used to validate the node name of the requesting node when attempting to execute a Connect:Direct Submit/RunJob/RunTask function on the local node.

### **NDM\$\$PERMITTED\_NEIGHBOR\_RUN\_DISABLE**

specifies that the execution of Connect:Direct Submit/RunJob/RunTask functions from all requesting nodes is to be disabled.

The value of this logical name is irrelevant. It can be defined externally or as an initial parameter and can be defined or deassigned at any time. When specified, it overrides NDM\$\$PERMITTED\_NEIGHBOR\_RUN\_DBX. Both logical names can specify the same file. However, these logical names will generally specify different files and there are no restrictions regarding filenames.

## Modifying Permitted Neighbor Database

You can change the logical name specifications without terminating Connect:Direct for OpenVMS. By using version numbers (or by *rename*), the new permitted neighbor database file(s) will take effect at the next inbound session.

## Sample Permitted Database Entry

A template file describing the format and conventions of a permitted database entry is supplied in the ndm\_sampsrc.tlb example library. The following is a sample database entry.

```

!-----
!
! Name:          PERMITTED NEIGHBOR DEFINITION TEMPLATE
!
! Description:   This file contains the input definitions for the
!               permitted neighbor database. Each entry has the
!               following form:
!
!               node days p_times s_times * * * *
!
!               where ...
!
!               node   - The Connect:Direct node name of the remote system.
!
!               days   - A 7-character string which specifies permitted
!                       days.
!
!               p_times - A 24-character string which specifies primary
!                       permitted times.
!
!               s_times - A 24-character string which specifies secondary
!                       permitted times.
!
!               *      - A field reserved for future use.
!
!               The 'days' field specifies the permitted days in Monday,
!               Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday
!               (MTWTFSS) order.
!
!               A 'p' indicates a primary day.
!               An 's' indicates a secondary day.
!
!               MTWTFSS
!               Example: The specification pppppss indicates that Monday,
!                       Tuesday, Wednesday, Thursday, and Friday are primary
!                       days while Saturday and Sunday are secondary days.
!
!               The 'p_times' field indicates the permitted hours of a
!               primary day in 0-23 order.
!
!               A '#' indicates access is permitted and all others deny
!               access.
!
!               0000000001111111112222
!               012345678901234567890123
!
!               Examples: The specification -----#####----- allows
!                       access between 8:00 am and 4:59 pm.
!
!               0000000001111111112222
!               012345678901234567890123
!
!               The specification ##### allows
!               access at all times.

```

(continued)

```

!
!
!
!   The 's_times' field indicates the permitted hours of a
!   secondary day and is specified in the same manner as
!   'p_times'.
!
!
!           000000000011111111112222
!           012345678901234567890123
!
!   Examples: The specification -----##----- allows
!             access between 8:00 am and 9:59 am.
!
!
!   New entries may be created by duplicating the template
!   entry. The definition file is used by the utility program
!   (ndm_permitted_neighbor_dbx.exe) as input to construct a
!   resulting database file to be searched at the time of an
!   incoming session. A new database file may be generated at
!   any time when enabled with a complete file specification
!   and takes effect at the next incoming session.
!
!
!   The utility program reads 'ndm$permitted_neighbor_dbx.src'
!   source file and creates 'ndm$permitted_neighbor_dbx.dat'
!   database file (via 'run ndm_permitted_neighbor_dbx') which
!   may then be renamed as desired.
!
!
!   The use of this facility is entirely OPTIONAL and is
!   specified by the following logical names:
!
!
!           NDM$$PERMITTED_NEIGHBOR_DBX
!
!           This logical name specifies which nodes are permitted
!           access using the file specified by the logical name.
!
!           NDM$$PERMITTED_NEIGHBOR_RUN_DBX
!
!           This logical name specifies which nodes are permitted
!           to execute a C:D Submit/RunJob/RunTask function on
!           the local node using the file specified by the
!           logical name.
!
!           These logical names may be defined externally or as
!           initial parameters and may be changed or deassigned
!           at any time.
!
!           Note that if specified, these logical names may
!           specify different files.
!
!           The '!' character is used to specify a tail-comment.
!
!
!
!
!           Primary Times           Secondary Times
!-----
!  NODE NAME      MTWTFSS  000000000011111111112222  000000000011111111112222  RESERVED
!-----
!                -012345678901234567890123--012345678901234567890123-----
!
!remote_node_name  pppppps  ##### * * * *

```



## Understanding Rights Identifiers

The username is checked to see which Connect:Direct functions are allowed, such as, SUBMIT, COPY, and RUN TASK. Authority to perform Connect:Direct functions is assigned to an OpenVMS account by granting rights identifiers corresponding to the functions.

In addition to all operator rights identifiers, administrators receive the following:

<b>Rights Identifiers for Administrators (NDM_ADMIN)</b>	<b>Definition</b>
NDM_[NO]DELNET	Denies/permits issuing DELETE NETMAP commands.
NDM_[NO]UPDNET	Denies/permits issuing UPDATE NETMAP commands.
NDM_[NO]INSNET	Denies/permits issuing INSERT NETMAP commands.

In addition to all general user rights identifiers, operators receive the following:

<b>Rights Identifiers for Operators (NDM_OPER)</b>	<b>Definition</b>
NDM_[NO]STOPNDM	Denies/permits issuing STOPNDM commands.
NDM_[NO]SELPROC_ALL	Denies/permits deleting, suspending, flushing, changing, and selecting information for any Process submitted by any user.
NDM_[NO]SECOVRD	Denies/permits specifying PNODEID/SNODEID values in a Process.
NDM_[NO]MODIFY	Denies/permits issuing Modify commands.
NDM_[NO]SELSTAT_ALL	Denies/permits selecting statistics information for any Process submitted by any user.
NDM_[NO]SUBSTMT	Denies/permits using the SUBMIT statement in a Process.

The following rights identifiers and definitions apply to general users:

<b>Rights Identifiers for General Users (NDM_GENUSER)</b>	<b>Definition</b>
NDM_[NO]CHGPROC	Denies/permits issuing CHANGE PROCESS commands.
NDM_[NO]FLSPROC	Denies/permits issuing FLUSH PROCESS commands.
NDM_[NO]SUSPROC	Denies/permits issuing SUSPEND PROCESS commands.
NDM_[NO]SELNET	Denies/permits issuing SELECT/SHOW NETMAP commands.
NDM_[NO]SIGNON	Denies/permits issuing SIGNON commands.

<b>Rights Identifiers for General Users (NDM_GENUSER)</b>	<b>Definition</b>
NDM_[NO]COPY	Denies/permits using the COPY statement in a Process.
NDM_[NO]DELPROC	Denies/permits issuing DELETE PROCESS commands.
NDM_[NO]SELPROC	Denies/permits issuing SELECT/SHOW PROCESS commands. Only Processes submitted by the user are selected.
NDM_[NO]SUBCMD	Denies/permits issuing SUBMIT commands.
NDM_[NO]SELSTAT	Denies/permits selecting statistics information of your work.
NDM_[NO]RUNTASK	Denies/permits using the RUN TASK statement in a Process.
NDM_[NO]RUNJOB	Denies/permits using the RUN JOB statement in a Process.

## Assigning Rights Identifiers

One of the three groups of rights identifiers can be assigned to an individual user. Individual users also can be granted Connect:Direct authority by explicitly granting individual functions.

Another way to assign rights is to assign a group rights identifier with status and functions explicitly disabled. For example, a user could be granted operator status but have the NDM\_NOSTOPNDM username applied, denying that user a right (to stop Connect:Direct execution) that is otherwise granted by the group NDM\_OPER.

The following commands are used to assign a user, or users, NDM\_GENUSER rights:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF> GRANT /ID NDM_GENUSER username1
UAF> EXIT
```

**Note:** Having NDM\_OPER status implies having NDM\_GENUSER status. Having NDM\_ADMIN implies having NDM\_OPER and NDM\_GENUSER status.

## Using Numeric Usernames

OpenVMS does not allow rights identifiers to be assigned to numeric usernames. If you have numeric usernames, create a file named NDMRIGHTS.DAT in NDM\$\$DIRECTORY. Add the following information to the file:

- ❖ Numeric username, which is defined as a username beginning with a number. The remainder of the username can be alphabetic or numeric. For example, 123456 and 1ABC1D are valid numeric usernames.
- ❖ One or more rights identifiers, separated by spaces. Continuation characters are supported.

A sample of NDMRIGHTS.DAT follows:

```

! Rights Identifiers for Numeric VMS Usernames
! -----
!
! Username      Rights
! -----
123456          ndm_admin ndm_noselstat_all    ! First username
1ABCDE          ndm_admin ndm_nodelnet -      ! Second username
                ndm_noupdnet ndm_noinsnet
1ABC2D          ndm_genuser                    ! Third username

```

---

## Managing Remote Access Control with a Password File

You can control outbound remote site access more efficiently by maintaining a password file. A password file allows you to specify remote access control within a script thus eliminating the need to maintain passwords in scripts or command procedures.

### Generating a Password File

The following privileges are required to create a password file:

- ❖ SYSPRV
- ❖ OPER
- ❖ SYSNAM
- ❖ CMKRNL
- ❖ SECURITY
- ❖ READALL

Use the following command procedure to generate a password file:

```
@ ndm$generate_password_file pl
```

In the previous example, *pl* specifies the input source file.

### Input File Format

The password generation source file is a text file with blank-delimited case-sensitive username and password fields. The ! (tail comments) are ignored. The format of an input file entry is as follows:

```
identifier username password
```

The following are definitions for the input file entries:

#### **identifier**

is a 1–128 character name used to uniquely identify a username/password entry. All characters, except spaces, are valid. Identifiers must be unique; however, multiple identifiers can refer to the same username/password combination.

For example, if you have a secondary node name of 'uk\_node' and a user name of 'blair', you can use the following string as an account identifier:

```
uk_node::blair
```

**username**

is a 1–255 character account username. All characters, except spaces, are valid.

**password**

is a 1–255 character account password. All characters, except spaces, are valid.

The utility generates an output file (ndm\$password.dat) that contains the account identifier, username, and the encrypted password.

## Password Source File Example

The following is an example of a password source file:

```
!
!           VIP
!
us_node.president      bill      clinton    ! USA
uk_node.prime         tony      blair      ! UK
japan_node.prime      keizo     obuchi     ! JAPAN
cis_node.president    boris     yeltsin    ! RUSSIA/CIS
canada_node.prime     jean      chretien   ! CANADA
vatican_node.pope     john      paul       ! VATICAN
!
!           VIM
!
disney_node.world1    mickey    mouse      ! OTHER/NEVER
disney_node.world2    minnie    mouse      ! OTHER/NEVER
```

## Modifying a Password File

Complete the following steps to create or modify a password file.

1. Edit the password file by entering the following command:

```
$ edit your_password_file.txt
```

2. Enter the following command:

```
$ @ ndm$generate_password_file your_password_file.txt
```

3. Delete the file by entering the following command:

```
$ delete your_password_file.txt;* /erase
```

4. Rename the file by entering the following command:

```
$ rename ndm$password.dat some_meaningful_name.dat
```

## Using a Password File

You can use a password file to complete the following tasks:

- ❖ Specify an 'snodeid' value with the /XSID qualifier.
- ❖ Perform placeholder substitution into compiled scripts.

## Specifying an snodeid Value

The user interface program, NDMUI, references a password file when the Connect:Direct Submit verb contains the /XSID qualifier. The password filename is specified by the logical name 'NDM\$\$PASSWORD\_FILE' and can be defined at any level or mode. There are no restrictions on the name or number of password files, and access to such files is determined by standard OpenVMS conventions.

The following is an example of using the /XSID qualifier:

```
ndmui> submit script_file.ndm /xsid=remote_acct
```

The qualifier value performs a lookup in the password file. If the specified entry exists, the 'snodeid' field is established or is overridden with the entry username and encrypted password. If the entry does not exist, no error is generated and the remote access control specified in the script, if any, remains intact.

## Performing Placeholder Substitution

You can use the password file to perform placeholder substitution in compiled scripts. Normally you would place an 'snodeid' specification in a script consisting of the "?\_act1" placeholder as follows:

```
txname process snode=remote.node snodeid=(?_act1)
```

Site-written procedures can also access the password file to define remote username placeholder values and to perform additional processing. The file format is as follows:

0	128	132	136	392
Identifier	Username_length	Password_length	Username	Password

The following are definitions for the password file:

### Identifier

account identifier name in uppercase and blank-filled with a maximum of 128 characters.

### Username\_length

the 4-digit decimal ASCII length of the Username.

### Password\_length

the 4-digit decimal ASCII length of the Password.

### Username

the remote account username zero-filled with a maximum of 255 characters.

### Password

the encrypted remote account password zero-filled with a maximum of 255 characters.

## Using a DCL Procedure

An example procedure, **ndm\$csx\_pwd\_lookup.com**, which illustrates how to use a password file within a DCL procedure is provided in the **ndm\_sampsrc.tlb** library.

---

# Reviewing Additional Installation Considerations

This chapter discusses the following topics:

- ❖ Using the Connect:Direct message utility
- ❖ Using the Connect:Direct translation table utility
- ❖ Modifying the SYLOGIN.COM file
- ❖ Activating statistics archival and deletion

---

## Using the Connect:Direct for OpenVMS Message Utility

NDMMSG.EXE is a utility that allows you to display, create, delete, and replace messages in the Connect:Direct message file. The following figure shows how to invoke the utility by defining a foreign command, displaying a message, and exiting the utility.

```
$ ndmmsg:= $ ndm$$directory:ndmmsg.exe
$ ndmmsg ndm$$directory:msgfile.dat
Message file opened for READ/WRITE/DELETE/UPDATE access :
ndm$$directory:msgfile.dat
NDMMSG> msg scpa000i
COPY successfully completed.
NDMMSG> exit
$
```

The file specification of the message file is optional. If it is not specified, the logical NDM\$\$MSGFILE in the UI logicals table is translated and that file specification is used.

---

**Note:** It is recommended that you use the NDM\$MSGID\_DISPLAY.com procedure rather than NDMMSG.EXE.

---

## Viewing Text Messages

Messages consist of short text, a single line of information, followed by long text, additional lines with more detailed information. Only the short text is displayed by the Connect:Direct for OpenVMS user interface when an error occurs. To view the long text, issue the following UI command:

```
SHOW MESSAGE msgid
```

---

## Using the Connect:Direct for OpenVMS Translation Table Utility

The NDM\_BLD\_XLATE.EXE utility allows you to define character translation tables and place them in the Connect:Direct translation table library for use during file transfer operations.

To invoke the utility, define an OpenVMS foreign command as follows:

```
$ nxl := $ndm$$directory:ndm_bld_xlate.exe
```

Issue the NXL command with two arguments as follows:

- ❖ File specification of the file containing the utility commands and table data
- ❖ File specification of the OpenVMS text library to contain the translation table(s) being defined

For example:

```
$ nxl user_xlate_defs.xlt ndm$$directory:ndm_xlate.tlb
```

Each table consists of a lower and upper half. The upper portion of the table will translate ASCII to another character set, typically EBCDIC or some variation of it. The lower portion of the table translates from EBCDIC to ASCII.

You can create or replace translation tables and optionally use an existing translation table as a model for the new one. Specify an offset followed by the actual values to place in the table at that offset. More than one offset command can be specified per module.

A SET RADIX command allows you to specify the radix of the offset value and the table input values.

The following list describes the utility commands:

### **DEFINE /qualifiers**

defines a translation table.

Values for /qualifiers are as follows:

**/MODEL=table name** specifies an existing translation table to use as a base to modify and store as the *defined* table.

**/REPLACE** replaces the translation table if it already exists.

### **FILLER=n|LOC**

specifies a value to initialize the translation table with before filling it in. Locations within the table that have not been assigned a specific value will contain this value.

**n** is a value in the radix specified by the SET RADIX command.



**LOC**, the default, specifies that each location in the table will have as its value its displacement within the table (that is, displacement x'40' will have a value of x'40'). The lower half and the upper half of the table will each contain the values x'00' through x'FF'.

**NOLOG\_INPUT|LOG\_INPUT**

specifies whether input will be displayed. The default is NOLOG\_INPUT, which specifies that input will not be displayed.

**OFFSET=n|(L,n)|(U,n)**

gives the offset into the translation table being defined where the following table values should be placed.

**n** can range from 0 to 511 where OFFSET=n. For (L,n)|(U,n), **n** can range from 0 to 255.

**L** indicates the lower 256 bytes of the translation table.

**U** indicates the upper 256 bytes of the translation table. The default is zero.

**SET RADIX=DECIMAL|HEX|OCTAL**

set the radix for data input. HEX is the default.

**!|\***

indicate comments and can be included in the input. Code either of these characters in column one to create a comment.

The following figure shows a sample table definition. The actual data follows the DEFINE command and is a series of numbers in the specified radix separated by blanks.

```

!   This file contains the commands and data for
!   the NDM_BLD_XLATE.EXE utility to create the
!   NDM_DEFAULT translate table in NDM_XLATE.TLB.
!
define ndm_default /replace
set radix=hex
!
!   Define the ASCII-to-EBCDIC part of the table.
offset=0
00 01 02 03 37 2D 2E 2F 16 05 25 0B 0C 0D 0E 0F
10 11 12 13 3C 3D 32 26 18 19 3F 27 1C 1D 1E 1F
40 5A 7F 7B 5B 6C 50 7D 4D 5D 5C 4E 6B 60 4B 61
F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 7A 5E 4C 7E 6E 6F
7C C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6
D7 D8 D9 E2 E3 E4 E5 E6 E7 E8 E9 AD E0 BD 5F 6D
79 81 82 83 84 85 86 87 88 89 91 92 93 94 95 96
97 98 99 A2 A3 A4 A5 A6 A7 A8 A9 C0 4F D0 A1 07
20 21 22 23 24 15 06 17 28 29 2A 2B 2C 09 0A 1B
30 31 1A 33 34 35 36 08 38 39 3A 3B 04 14 3E E1
41 42 43 44 45 46 47 48 49 51 52 53 54 55 56 57
58 59 62 63 64 65 66 67 68 69 70 71 72 73 74 75
76 77 78 80 8A 8B 8C 8D 8E 8F 90 9A 9B 9C 9D 9E
9F A0 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7
B8 B9 BA BB BC BD BE BF CA CB CC CD CE CF DA DB
DC DD DE DF EA EB EC ED EE EF FA FB FC FD FE FF
!
!   Define the EBCDIC-to-ASCII part of the table.
offset=100
00 01 02 03 9C 09 86 7F 97 8D 8E 0B 0C 0D 0E 0F
10 11 12 13 9D 85 08 87 18 19 92 8F 1C 1D 1E 1F
80 81 82 83 84 0A 17 1B 88 89 8A 8B 8C 05 06 07
90 91 16 93 94 95 96 04 98 99 9A 9B 14 15 9E 1A
20 A0 A1 A2 A3 A4 A5 A6 A7 A8 5B 2E 3C 28 2B 7C
26 A9 AA AB AC AD AE AF B0 B1 21 24 2A 29 3B 5E
2D 2F B2 B3 B4 B5 B6 B7 B8 B9 7C 2C 25 5F 3E 3F
BA BB BC BD BE BF C0 C1 C2 60 3A 23 40 27 3D 22
C3 61 62 63 64 65 66 67 68 69 C4 C5 C6 C7 C8 C9
CA 6A 6B 6C 6D 6E 6F 70 71 72 CB CC CD CE CF D0
D1 7E 73 74 75 76 77 78 79 7A D2 D3 D4 5B D6 D7
D8 D9 DA DB DC DD DE DF E0 E1 E2 E3 E4 5D E6 E7
7B 41 42 43 44 45 46 47 48 49 E8 E9 EA EB EC ED
7D 4A 4B 4C 4D 4E 4F 50 51 52 EE EF F0 F1 F2 F3
5C 9F 53 54 55 56 57 58 59 5A F4 F5 F6 F7 F8 F9
30 31 32 33 34 35 36 37 38 39 FA FB FC FD FE FF

```

---

## Modifying the SYLOGIN.COM File

If you use SET TERMINAL/INQUIRE in your SYLOGIN.COM or LOGIN.COM files, the RUN TASK statement will output an error message for an invalid device type.

To prevent the error message, include the following commands in your LOGIN.COM files:

```
$ term_name=f$logical ("sys$command")
$ if (f$length(term_name) .eq. -
    f$locate("MBA", term_name)) then -
    if (f$mode() .eqs. "INTERACTIVE") then -
    set terminal/inquire
```

---

## Activating Statistics Archival and Deletion

The Connect:Direct for OpenVMS statistics file can be archived and deleted automatically. A site can determine the maximum size and the maximum number of versions that are to be retained. When the content of the statistics files reaches the site-specified limit, a new version is automatically created and the oldest versions are then deleted.

The logical name parameters that control this process can be defined as initialization parameters. They are as follows:

- ❖ NDM\$\$STAT\_DIRECTORY
- ❖ NDM\$\$STAT\_MAX\_VERSION
- ❖ NDM\$\$STAT\_MAX\_BLKSIZE
- ❖ NDM\$\$STAT\_PROCESS\_PRIORITY
- ❖ NDM\$\$STAT\_MAX\_MESSAGE\_RX\_QSIZE

Refer to the table on page 28 for a complete list of the parameter definitions.

## Deferring Statistics Files Records

Statistics files records are written in a deferred manner; that is, they are not written as they are generated. Because of this behavior, it is possible that the statistics files may not reflect the most recent activity at all times. Since there is such wide variation in Connect:Direct for OpenVMS usage patterns, you should configure the generation of the statistics files in a manner that reflects your site's specific usage pattern. The optional parameters that control the deferral of statistics files records are as follows:

**ndm\$\$stat\_process\_priority**

this logical name is used to establish the base priority of the process which records the statistics records. Its value is between 0 and 31. There is no default value.

**ndm\$\$stat\_message\_window\_size**

this logical name is used to establish a message window size, that is, it determines the maximum number of statistics records which may be outstanding at any moment. Its value is between 10 and 10000. There is no default value.

**ndm\$\$stat\_message\_window\_priority**

this logical name is used to establish a message window priority. When specified, the priority of the process which records statistics records is elevated to this value and after the window is closed, the process priority is lowered to its base value. Its value is between 16 and 31. There is no default value.

These parameters are defined in the optional *ndm\_stat\_writer\$setparams.com* procedure. This procedure, if present, is executed by the Connect:Direct for OpenVMS startup procedure. Unlike parameters specified in the *initparms.dat* files, these are specified with ordinary DCL as in the following example:

```
$!
$ define ndm$$stat_process_priority          10 ! Base Priority
$ define ndm$$stat_message_window_size      16 ! Message Window
$ define ndm$$stat_message_window_priority  16 ! Floating Priority
$!
$exit 1
```

A site should specify, or not specify, these values according to expected usage. A site with low usage need not specify these parameters. If your site has heavy usage, then employ a message window if you wish the statistics files to reflect current activity.

---

## Activating Statistics Archive-On-Demand

You can archive the statistics files on demand regardless of file size, version limitations, or time parameters. This procedure can be executed directly or from within a site-written procedure. You must have OPER privilege in order to invoke this procedure. Invoke the DCL procedure by issuing the following command:

```
@ ndm$stat_archive [pl]
```

In the previous example, *pl* is the environment name. If not specified, the default is NDM.

## Sample Statistics Archival Procedure

A sample procedure is provided which illustrates how you can invoke this procedure at regular intervals. The example procedure **ndm\$stat\_archive\_maintenance.com** is located in the **ndm\_sampsrctlb** text library.

Invoke this procedure by issuing the following command:

```
@ ndm$stat_archive_maintenance [initial] [repeat] [env]
```

The following are the parameter definitions:

**initial**

is the initial execution delta time

**repeat**

is the repeat interval delta time

**env**

is the environment name

## Performing Daily Archival

Use the following procedure to perform daily archival:

```
' @ ndm$stat_archive_maintenance tomorrow "+1-00:00:00.00" '
```

The following is an example of a weekly archival cycle beginning at midnight Saturday, August 15. In this example, assume that today is Wednesday, August 12.

```
' @ ndm$stat_archive_maintenance "15-AUG:" "+7-00:00:00.00" .'
```

The procedure is not all inclusive and presents only one possible way to invoke the **ndm\$stat\_archive.com** procedure.



---

# Modifying the Network Map

Before you can use the Connect:Direct for OpenVMS system, you must load the network map with the local node and adjacent nodes. After modifying the network map, you might decide to add or delete additional nodes and records or you might want to update the records already in the network map.

This chapter describes the following network map (NETMAP) commands by explaining their function, syntax, and qualifiers:

- ❖ INSERT NETMAP
- ❖ DELETE NETMAP
- ❖ UPDATE NETMAP

Examples of each command are at the end of each section.

---

**Note:** The SHOW NETMAP command is discussed in the *Connect:Direct for OpenVMS User's Guide*.

---

---

## Defining a Network Map

During installation, the Connect:Direct software creates the network map file. The network map file contains node records that describe the nodes in the Connect:Direct network. The local node record is the name of your node in the network map. An adjacent node is the remote-site node used when the local node requests a connection. The network map created during installation contains a local node record, as well as an adjacent node record for the local node (allowing PNODE=SNODE sessions for testing the installation). You can create additional adjacent node records manually.

The network map record for an adjacent node contains the following:

- ❖ Adjacent node name
- ❖ Default connection
- ❖ Maximum number of communication sessions
- ❖ Default class used to establish sessions
- ❖ Information on communications paths

The local node record contains only the local node name.

Examples of the network map record are shown in the *SHOW NETMAP* section of the *Connect:Direct for OpenVMS User's Guide*.

Network definitions in one network map must match the definitions in the network maps for the other nodes on the network. For example, suppose that an OpenVMS node communicates with an IBM z/OS node. The way

in which each node is defined to the other node must correspond. For instance, if you changed node definitions in the Connect:Direct for OpenVMS network map but not in the Connect:Direct for z/OS network map, the nodes could no longer communicate with each other.

If you are using Connect:Direct Secure+ Option for OpenVMS, you should use the Keep in Sync option available in Secure+ Option whenever you add a record to the network map. Also, whenever you delete a record in the network map, you should manually delete the record for the same node in the Secure+ Option parameters file. See the *Connect:Direct Secure+ Option for OpenVMS Implementation Guide* for complete instructions.

You should grant users authorization privileges for the NETMAP commands. All users should be able to use the SHOW NETMAP command. This command allows users to look at, but not alter, the nodes in the network that they are authorized to use. The extent to which users can use the other NETMAP commands depends on their authorization.

---

## Reviewing a Sample NETDEF.COM Procedure

The example on page 65 shows the command procedure, NETDEF.COM, created during installation; this file contains sample commands for creating network map records. The environment-dependent variables are in lower case and are defined following the sample NETDEF.COM file. The LOCAL.NODE record, as well as the PNODE=SNODE ADJACENT.NODE record, were created by the installation procedure; others must be added manually.

---

**Note:** If you make changes directly to the NETDEF.COM file, you must re-execute it in order for the changes to take effect.

---

The following are the six types of records that can be defined in a network map:

- ❖ The LOCAL\_NODE record defines the name by which the other nodes will know the Connect:Direct for OpenVMS node. The node record was created by the installation procedure. In the sample NETDEF.COM file, *vmsnode.ndm* is the name of the LOCAL\_NODE record.
- ❖ An ADJACENT\_NODE record for PNODE=SNODE Processes makes it possible to send data to your node. This node record was created by the installation procedure. The adjacent node name for PNODE=SNODE Processes must match the name assigned in the LOCAL\_NODE record, *vmsnode.ndm*.
- ❖ An ADJACENT\_NODE record for each remote OpenVMS node running a server defines the node name of a remote OpenVMS system in your network.
- ❖ An ADJACENT\_NODE record for each remote TCP/IP node running a server defines the node name of a remote TCP/IP system in your network.



In the following example, each INSERT command is preceded by a DELETE command, allowing the command procedure to be re-executed whenever changes are made to it. Errors are generated if you try to delete a node that has not been added. These errors should be ignored.

```

!-----!
! NETMAP configuration command procedure. Created by the !
! Connect:Direct setup command procedure when the default !
! local node access is defined. Additional node !
! definitions are added to this file. !
!-----!
! Example definitions of remote nodes
!-----!
! Remote DEC node
$NDMUI
!
!DELETE NETMAP ADJACENT_NODE=dallas.vms.ndm
!INSERT NETMAP ADJACENT_NODE=dallas.vms.ndm PARSESS=(16,1) -
!  COMM_PATH= -
!  ("NAME=dallas TYPE=DECNET DECNET.NODE=dallas DECNET.OBJ=ndm") -
!  /DEFCONN_MODE=FIRST
!
! Remote TCP/IP node
!
!DELETE NETMAP ADJACENT_NODE=newark.tcp.ndm
!INSERT NETMAP ADJACENT_NODE=newark.tcp.ndm PARSESS=(16,1) -
!  COMM_PATH=("NAME=newark TYPE=TCP TCP.HOSTNAME=255.255.255.255
!  TCP.PORTNUM=2001") -
!  /DEFCONN_MODE=first
!
! Remote IBM node
!
!-----!
! Add site specific definitions here.
!-----!
!
!DELETE NETMAP TCP_NODE=NTALL
!INSERT NETMAP TCP_NODE=NTALL/PATH=(NT40@10.20.4.210::1364,NT33@10.20.4.188::1364)
!
!DELETE NETMAP TCP_NODE=AIXALL
!INSERT NETMAP TCP_NODE=AIXALL/FILE=UNIX_NETMAP.DAT
!
! website
!DELETE NETMAP ADJACENT_NODE=THEWORLD
!INSERT NETMAP ADJACENT_NODE=THEWORLD PARSESS=(16,1) -
!  COMM_PATH=("NAME=DLNT TYPE=TCP TCP.HOSTNAME=204.214.3.37 TCP.PORTNUM=1364") -
!  /DEFCONN_MODE=FIRST
!
!-----!
! Exit User Interface
!-----!
EXIT
$!
$!-----!
$! Exit DCL command procedure
$!
$!-----!
$!
$EXIT

```

The following is a list of the environment-specific parameters in the network map. An explanation of each parameter, as well as where the correct value can be found, is included.

### **ADJACENT\_NODE**

is the name of a remote system to which you will be communicating. It matches the LOCAL.NODE defined in the network map at the remote node.

### **APPLID**

is the VTAM Application ID name of the remote system. It matches the APPLID defined in the LOCAL.NODE statement in the IBM network map.

### **DECNET.OBJ**

is the user-defined DECnet object name of the Connect:Direct system running on the remote node. It corresponds to the environment name.

### **LOCAL\_NODE**

is the name of the OpenVMS node as it will be known by other nodes. It must match the ADJACENT.NODE record for this node in the IBM network map.

### **NAME**

is the user-defined name of the path. It must be unique.

**PARSESS**

indicates that parallel sessions are allowed. The first number is the maximum number of PNODE communication sessions within any class; the second number is the default class.

**TYPE**

specifies the type connection the record is defining. Refer to the *INSERT NETMAP ADJACENT\_NODE* section of this chapter for a complete description of this parameter.

---

## Adding Records to the Network Map

INSERT NETMAP commands allow you to insert or add the local node and adjacent node records into the network map. You can use the INSERT command to add a new node to the NETMAP without shutting down the Connect:Direct system.

The following sections discuss the INSERT NETMAP commands:

- ❖ INSERT NETMAP LOCAL\_NODE
- ❖ INSERT NETMAP ADJACENT\_NODE

### Using the INSERT NETMAP LOCAL\_NODE Command

The INSERT NETMAP LOCAL\_NODE command allows you to add the local node to the network map. The local node is the name of the local Connect:Direct node.

The following figure shows the format of the INSERT NETMAP LOCAL\_NODE command.

```
$ NDMUI INSERT NETMAP LOCAL_NODE=local.ndm -
_> /SERVER=server_alias
```

The command and qualifier are described as follows:

**INSERT NETMAP LOCAL\_NODE=*local.ndm***

is the NETMAP command for inserting a local node, where *local.ndm* is the 1–16 character name of the local node. This is the name by which the other nodes will know the local Connect:Direct for OpenVMS node.

**/SERVER=*server\_alias***

specifies the particular server to receive the command. This qualifier is optional.

*server\_alias* specifies the 1–16 alphanumeric character name of the server as defined in NDM\$DIRECTORY:SERVER.DAT.

### Using the INSERT NETMAP ADJACENT\_NODE Command

The INSERT NETMAP ADJACENT\_NODE command allows you to add an adjacent node to the network map. The adjacent node record defines a remote system with which the local Connect:Direct for OpenVMS communicates.

The following figure shows an example of the INSERT NETMAP ADJACENT\_NODE command format for a DECnet connection.

```
$ NDMUI INSERT NETMAP ADJACENT_NODE=decnet.node-
_ $ PARSESS=(x,y)-
_ $ COMM_PATH=( "NAME=vms2 TYPE=DECNET DECNET.NODE=node
DECNET.OBJ=objname" )
```

The following figure shows the INSERT NETMAP ADJACENT\_NODE command format for a TCP/IP connection:

```
$ NDMUI INSERT NETMAP ADJACENT_NODE=tcp.node-
_ $ PARSESS=(x,y)-
_ $ COMM_PATH=( "NAME=vms3 TYPE=TCP TCP.HOSTNAME=nnn.nnn.nnn.nnn
TCP.PORTNUM=n" )-
_ $ /DEFCONN_MODE=first
```

The following figure shows an example of the INSERT NETMAP ADJACENT\_NODE command format for a PNODE=SNODE connection.

```
$ NDMUI INSERT NETMAP ADJACENT_NODE=sc.vms.p2-
_ $ PARSESS=(x,y)-
_ $ COMM_PATH=( "NAME=pnodesnode TYPE=PNODESNODE" )
```

The command, parameters, and qualifiers are described as follows:

#### **INSERT NETMAP ADJACENT\_NODE=*remote.ndm***

is the NETMAP command for inserting an adjacent node, where the *remote.ndm* is the 1–16 character name of the remote system with which you will communicate. You must create an adjacent node record for each remote system with which you will communicate.

You can define a second adjacent node record for the local node. Defining the local node as an adjacent node allows you to execute a PNODE=SNODE Process, making it possible to send data to your node. For PNODE=SNODE operations, the adjacent node name must match that assigned as *local.ndm*.

#### **PARSESS=(*x,y*)**

indicates that concurrent sessions are allowed, where *x* is the maximum number of active communication sessions (PNODE sessions) and *y* is the default class.

Because each session has a corresponding unique class value, the maximum number of sessions and maximum number of classes are equal. User-specified class values allow a Process to execute on the session having the matching class value, or on sessions with higher class values.

#### **COMM\_PATH=**

**("NAME=*path* TYPE=DECNET|TCP|PNODESNODE . . .")**

specifies the name of the communications path and the type of connection to make. You can specify any number of paths by adding a new record and renaming it. Each of the names in the communications path should be unique to that specific path. All valid parameters are listed as follows:

#### **NAME=*path***

is the user-specified name of the communications path. The name can be 1–16 alphanumeric characters long.

#### **TYPE=(DECNET|TCP|PNODESNODE)**

**DECNET** specifies that the communications path is for a DECnet connection. The subqualifiers DECNET.NODE and DECNET.OBJ are also required when TYPE=DECNET.

**TCP** specifies that you are using Transmission Control Protocol/Internet Protocol (TCP/IP) communications. The subqualifiers TCP.HOSTNAME and TCP.PORTNUM are also required when TYPE=TCP.

**PNODESNOE** indicates that this record is to be used for copies within the local node.

**DECNET.NODE=*node***

specifies the name of the DECnet node where Connect:Direct for OpenVMS software is running. (For TYPE=DECNET nodes only)

**DECNET.OBJ=DECNET object name**

is the DECnet object name of the Connect:Direct system running on the remote node. (For TYPE=DECNET nodes only)

**TCP.HOSTNAME=*nnn.nnn.nnn.nnn***

is the IP address of the node where the remote Connect:Direct is running. (For TYPE=TCP only)

**TCP.PORTNUM=*n***

specifies the TCP/IP port number of the remote Connect:Direct node. (For TYPE=TCP only)

**/DEFCONN\_MODE= (*first|scan|name*) (Optional)**

defines the manner in which default communications paths are used.

*first* (default) specifies that Connect:Direct for OpenVMS will use the first COMM\_PATH specification only in making a connection.

*scan* specifies that Connect:Direct for OpenVMS will use each COMM\_PATH specification in turn, until a connection is successfully made.

*name* represents a specific COMM\_PATH name that Connect:Direct for OpenVMS should use when making connections.

## Deleting Records from the Network Map

The DELETE NETMAP command allows you to delete the local node and adjacent node records from the network map. If you delete adjacent node records that you might need again, re-enter them with the INSERT NETMAP command.

The following sections discuss the DELETE NETMAP commands:

- ❖ DELETE NETMAP LOCAL\_NODE
- ❖ DELETE NETMAP ADJACENT\_NODE

### Using the DELETE NETMAP LOCAL\_NODE Command

The DELETE NETMAP LOCAL\_NODE command allows you to specify the local node to be deleted from the network map. The local node is the name of the Connect:Direct for OpenVMS node as defined in the network map.

The following example shows the format of the DELETE NETMAP LOCAL\_NODE command.

```
$ NDMUI DELETE NETMAP LOCAL_NODE
```

The command is described as follows:

**DELETE NETMAP LOCAL\_NODE**

is the NETMAP command for deleting a local node. The name of the local node is not used.

**Using the DELETE NETMAP ADJACENT\_NODE Command**

The DELETE NETMAP ADJACENT\_NODE command allows you to specify the name of the adjacent node to be deleted from the network map. The adjacent node record defines the remote system with which the local Connect:Direct for OpenVMS communicates.

The following example shows the format of the DELETE NETMAP ADJACENT\_NODE command.

```
$ NDMUI DELETE NETMAP ADJACENT_NODE=remote.ndm
```

The command is described as follows:

**DELETE NETMAP ADJACENT\_NODE=*remote.ndm***

is the NETMAP command for deleting an adjacent node, where node name is the 1-16 character name of an adjacent node as defined in the network map file.

**Reviewing Examples of DELETE NETMAP Commands**

The following command deletes the local node from the network map. The name of the local\_node is not used.

```
$ NDMUI DELETE NETMAP LOCAL_NODE
```

The following command deletes the adjacent node named SC.MVS.KS from the network map.

```
$ NDMUI DELETE NETMAP ADJACENT_NODE=SC.MVS.KS
```

---

**Updating Records in the Network Map**

The UPDATE NETMAP commands allow you to modify network map records. The following sections describe the two UPDATE NETMAP commands:

- ❖ UPDATE NETMAP LOCAL\_NODE
- ❖ UPDATE NETMAP ADJACENT\_NODE

**Using the UPDATE NETMAP LOCAL\_NODE Command**

The UPDATE NETMAP LOCAL\_NODE command allows you to rename the local node in the network map. The local node is the name of the Connect:Direct for OpenVMS node.

The following example shows the format of the UPDATE NETMAP LOCAL\_NODE command.

```
$ NDMUI UPDATE NETMAP LOCAL_NODE=local.ndm -  
_ $ /SERVER=server_alias
```

The command and qualifier are described as follows:

**UPDATE NETMAP LOCAL\_NODE=*local.ndm***

is the NETMAP command for changing the node name of a local node, where the node name is the 1-16 character name of a local node as defined in the network map file.

**/SERVER=*server\_alias***

specifies the particular server to receive the command. This qualifier is optional.

*server\_alias* specifies the 1–16 alphanumeric character name of the server as defined in NDM\$\$DIRECTORY:SERVER.DAT.

## Using the UPDATE NETMAP ADJACENT\_NODE Command

The UPDATE NETMAP ADJACENT\_NODE command allows you to modify an adjacent node in the network map. The adjacent node record defines the remote system with which the local Connect:Direct for OpenVMS communicates. For example, you can use the UPDATE NETMAP ADJACENT\_NODE command to insert, modify, or delete a communications path for the adjacent node. The UPDATE NETMAP ADJACENT\_NODE command generates a series of prompts that are based on the TCP/IP protocol.

### Updating a TCP/IP Type ADJACENT\_NODE

The following figure shows the format of the UPDATE NETMAP ADJACENT\_NODE command for a TCP/IP connection. Entering **Y** in the Delete CommPath field will delete the specified communications path from the Network Map and return to the Connect:Direct> prompt.

---

**Note:** In general, the preferred method of updating the netmap is by use of the DELETE and INSERT commands, as shown in the NETDEF.COM command procedure.

---

```

$ NDMUI
Connect:Direct> UPDATE NETMAP ADJACENT_NODE=remote.node
CommPath Name:  path_name
Delete CommPath (Y/N):  N
Insert After CommPath:  path_name
ParSess Default:  defclass
ParSess Maximum:  maxclass
TCP Hostname (or address):  nnn.nnn.nnn.nnn
TCP Port Number:  n
Default Connection Mode:  mode

```

The command, parameters, and qualifiers are described as follows:

---

**Note:** The parameters used in the UPDATE NETMAP commands, unless listed as follows, are the same used in the previously defined INSERT NETMAP commands. Refer to page 66 for information regarding those parameters.

---

**UPDATE NETMAP ADJACENT\_NODE =*nodename***

is the NETMAP command for updating an adjacent node, where the node name is the 1–16 alphanumeric character name of an adjacent node as defined in the network map file.

**Insert After CommPath**

specifies an existing communications path after which a new COMM\_PATH should be inserted.

## Reviewing an Example of UPDATE NETMAP Command

The following example updates the local node in the network map. The new name of the local node must be specified.

```
$ NDMUI UPDATE NETMAP LOCAL_NODE=NDM.VAX2
```





## A

### **Adjacent Node**

An adjacent node is an entry in the Network Map that defines a Connect:Direct node with which the local Connect:Direct node can communicate. The adjacent node is also referred to as a remote node.

### **Application Program Interface (API)**

The Application Program Interface (API) is a Connect:Direct component that accepts commands and places them in an executable format.

## C

### **Checkpoint Restart**

The Checkpoint Restart feature eliminates the need to re-transmit an entire file in the event of a transmission failure. If a copy procedure is interrupted, Connect:Direct restarts that copy at the last checkpoint.

### **Command Line Interface**

The Command Line Interface is a Connect:Direct interface that allows users to submit Connect:Direct Processes and commands from their native command line environment.

### **Commands**

Connect:Direct commands initiate and monitor activity within the Connect:Direct system.

## I

### **Interactive Mode**

The interactive mode of operation allows you to input multiple commands with one invocation of the user interface (NDMUI).

## L

### Local Node

The local node is the Connect:Direct server.

## N

### NDMUI

is the command that invokes the user interface. The user interface is one of the windows through which you communicate with Connect:Direct for OpenVMS.

### Network Map (Netmap)

The Network Map (netmap) is a file that identifies all valid Connect:Direct nodes in the network. One Network Map is associated with each Connect:Direct local node. The netmap has one entry for each of the other Connect:Direct nodes to which the local Connect:Direct node communicates. The netmap entries also contain the rules or protocol that the nodes adhere to when communicating.

### Node

A node is any site in a network from which information distribution can be initiated.

### Noninteractive Mode

The noninteractive mode of operation allows you to issue a single command in one invocation of the NDMUI.

## P

### Primary Node (PNODE)

The Primary Node (PNODE) is the Connect:Direct node on which the Process is submitted. The primary node can also be referred to as the controlling node or initiating node, but should not necessarily be interpreted as the sending node, since PNODE can be the receiver. In every Process, there is one PNODE and one SNODE specified. The submitter of a Process is always the PNODE.

### Process

A Process is a series of statements that initiate Connect:Direct activity, such as copying files, running jobs, and so on.

### Process Statements

Process Statements are instructions for transferring files, running operating system jobs, executing programs, or submitting other Connect:Direct Processes. They are used to build a Connect:Direct Process.

## R

**Remote Node**

A remote node is an entry in the Network Map that defines a Connect:Direct node with which the local Connect:Direct node can communicate. The remote node is also referred to as an adjacent node.

**Retry Interval**

The retry interval is the interval at which retries are performed if the Process encounters nonfatal errors.

**S****Secondary Node (SNODE)**

The secondary node (SNODE) is the Connect:Direct node that interacts with the primary node (PNODE) during Process execution. SNODE can also be referred to as the participating (non-controlling) or partner node. Every Process has one PNODE and one SNODE.

**Server**

The server accepts command requests, communicates with the session manager when work is placed in the Transmission Control Queue (TCQ), and accepts session establishment requests from remote nodes.

**Session Manager**

The session manager establishes communication sessions, performs standard session management functions, and executes Processes.

**Statistics File**

The statistics file holds Connect:Direct statistics records that document the history of a Process.

**Statistics Facility**

The Connect:Direct Statistics Facility records Connect:Direct activities.

**T****Transmission Control Queue (TCQ)**

The Transmission Control Queue (TCQ) holds information about Connect:Direct Processes that are currently executing or scheduled to execute in the future.

**U****User Interface (NDMUI or UI)**

The communications process used to interface with the server. Commands are issued through the user interface when you want to work with the Connect:Direct product.



## Symbols

/DEFCONN\_MODE qualifier 68

### A

adding the ndmui command 19  
  in a cluster environment 19  
  in a non-clustered environment 18

adjacent node record 63  
  contents 63  
  deleting 69  
  inserting 66  
  updating 70

ADJACENT\_NODE network map parameter 65

ADJACENT\_NODE record 64

alternate environment  
  connecting to 39  
  considerations 38  
  general 26  
  starting Connect:Direct for OpenVMS 38  
  stopping Connect:Direct for OpenVMS 39

ALTPRI privilege 12, 24

APPLID network map parameter 65

### B

bold letters, notational conventions 9

brackets, notational conventions 9

BYPASS privilege 12, 24

### C

CKPTFILE.FDL file, ckptfile.fdl file 21

client files

  CONFIGUI.COM 20  
  COPYRIGHT.TXT 20  
  MSGFILE.DAT 20

client files (continued)

  MSGFILE.FDL 20  
  NDM\$CONFIG.COM 20  
  NDM\$Connect.COM 20  
  NDM\$DISConnect.COM 20  
  NDM\$SERVERDEF.COM 20  
  NDM.CBJ 21  
  NDM\_EXAMPLES.TLB 21  
  NDM\_SHUTDOWN.COM 21  
  NDM\_STARTUP.COM 21  
  NDMMSG.EXE 21  
  NDMUI.CLD 21  
  NDMUI.EXE 21  
  NDMUI.HLB 21  
  SERVER.DAT 21

client installations 15, 23  
  overview 11

client-only installations, configuring 33

cluster (system-wide), shutting down Connect:Direct for OpenVMS 33

cluster environment 15  
  considerations 15

cluster environment (alternate)  
  shutting down Connect:Direct for OpenVMS 39  
  starting Connect:Direct for OpenVMS 39

cluster environment (system-wide)  
  shutting down Connect:Direct for OpenVMS 33  
  starting Connect:Direct for OpenVMS 32

CMKRNL privilege 12, 24

COMM\_PATH network map parameter 67

commands

  DELETE NETMAP ADJACENT\_NODE 69  
  DELETE NETMAP LOCAL\_NODE 68  
  INSERT NETMAP ADJACENT\_NODE 66  
  INSERT NETMAP LOCAL\_NODE 66  
  UPDATE NETMAP ADJACENT\_NODE 70  
  UPDATE NETMAP LOCAL\_NODE 69

CONFIGSRV.COM file 21

configui.com 20

CONFIGUL.COM file 20

configuration

- general 23
- granting the NDM\_ADMIN identifier to the manager account 25
- of a client-only installation 33
- of a server installation 26
- shutdown procedure
  - for a system-wide environment 32
  - client-only installations 33
- shutdown procedure, for an alternate environment 39
- startup procedure
  - for a system-wide environment, client-only installations 33
  - for a system-wide environment, server-only installations 31
- startup procedure, for an alternate environment 38

configuration options, overview 11

Connect:Direct, description 7

Connect:Direct directory 20

Connect:Direct environments

- alternate
  - connecting to 39
  - considerations 38
  - general 26
  - starting Connect:Direct for OpenVMS 38
  - stopping Connect:Direct for OpenVMS 39
- cluster (alternate)
  - shutting down Connect:Direct for OpenVMS 39
  - starting Connect:Direct for OpenVMS 39
- cluster (system-wide) 33
  - starting Connect:Direct for OpenVMS 32
- cluster, considerations 15
- system-wide
  - general 26
  - starting Connect:Direct for OpenVMS
    - client-only installations 33
    - server installations 31
  - stopping Connect:Direct for OpenVMS
    - client-only installations 33
    - server installations 32

Connect:Direct for OpenVMS node name 27

Connect:Direct for OpenVMS target directory 20

connectivity options 7

copying Connect:Direct for OpenVMS files, to the target directory 19

copying Connect:Direct for OpenVMS help files, in the OpenVMS help library 18

COPYRIGHT.TXT 20

customizing Connect:Direct for OpenVMS 23

## D

DECNET.NODE network map parameter 68

DECNET.OBJ network map parameter 65, 68

defining
 

- initialization parameters 28
- remote servers 31

DELETE NETMAP ADJACENT\_NODE command 69

DELETE NETMAP LOCAL\_NODE command 68

DETACH privilege 12, 24

disk quotas 12

displaying Connect:Direct messages 56

## E

editing the network map file 34

environments

- alternate
  - connecting to 39
  - considerations 38
  - general 26
  - starting Connect:Direct for OpenVMS 38
  - stopping Connect:Direct for OpenVMS 39
- cluster
  - considerations 15
  - general 11, 15
- cluster (alternate)
  - shutting down Connect:Direct for OpenVMS 39
  - starting Connect:Direct for OpenVMS 39
- cluster (system-wide)
  - shutting down Connect:Direct for OpenVMS 33
  - starting Connect:Direct for OpenVMS 32
- non-cluster, general 11, 15
- system-wide
  - general 26
  - starting Connect:Direct for OpenVMS
    - client-only installations 33
    - server installations 31
  - stopping Connect:Direct for OpenVMS
    - client-only installations 33
    - server installations 32

EXQUOTA privilege 12, 24

**G**

granting the NDM\_ADMIN identifier to the manager account 25

**H**

help files 18

**I**

IMPERSONATE privilege 12

initialization parameters

description 28

NDM\$\$STAT\_DIRECTORY 59

NDM\$\$STAT\_MAX\_BLKSIZE 59

NDM\$\$STAT\_MAX\_MESSAGE\_RX\_QSIZE 59

NDM\$\$STAT\_MAX\_VERSION 59

NDM\$\$STAT\_PROCESS\_PRIORITY 59

INITPARMS.DAT file 21

INSERT NETMAP ADJACENT\_NODE command 66

DECnet connection 67

PNODE/SNODE connection 67

TCP/IP connection 67

INSERT NETMAP LOCAL\_NODE command 66

inserting Connect:Direct messages 56

installation

configuration steps 23

configuration types 15, 23

disk quotas 12

general 15, 17

in a cluster environment

adding the NDMUI command 19

general 15

in a non-clustered environment

adding the NDMUI command, 18

options 17

privileges required 12

shutdown procedure

for a cluster environment (alternate) 39

for a cluster environment (system-wide) 33

startup procedure

for a cluster environment (alternate) 39

for a cluster environment (system-wide) 32

steps 16

testing 34

verifying 20

invoke SETUPRIGHTS for the server 24

**L**

local node record 63

contents 63

deleting 68

inserting 66

updating 69

LOCAL\_NODE record 64

LOGIN.COM 18

lowercase letters, notational conventions 9

**M**

maintenance updates 37

manager account 25

message utility

displaying messages 56

general 55

inserting messages 56

modifying the DCL command table

in a cluster environment 19

in a non-clustered environment 18

msgfile.dat 20

MSGFILE.DAT file 20

msgfile.fdl 20

MSGFILE.FDL file 20

**N**

NAME network map parameter 67

NDM\$\$STAT\_DIRECTORY, initialization parameter 59

NDM\$\$STAT\_MAX\_BLKSIZE, initialization parameter 59

NDM\$\$STAT\_MAX\_MESSAGE\_RX\_QSIZE, initialization parameter 59

NDM\$\$STAT\_MAX\_VERSION, initialization parameter 59

NDM\$\$STAT\_PROCESS\_PRIORITY, initialization parameter 59

ndm\$config.com 20

NDM\$CONFIG.COM file 20

ndm\$connect.com 20

NDM\$Connect.COM file 20

ndm\$disconnect.com 20  
 ndm\$serverdef.com 20  
 NDM\$SERVERDEF.COM file 20  
 ndm.cbj 21  
 NDM.CBJ file 21  
 NDM\_ADMIN rights identifier 25  
 NDM\_BLD\_XLATE.EXE file 21  
 NDM\_DEFAULT.XLT file 21  
 NDM\_EXAMPLES.TLB file 21  
 NDM\_SAMPSRC.TLB file 21  
 ndm\_shutdown.com 21  
 NDM\_SHUTDOWN.COM file, 21  
 NDM\_SMGR.EXE file 21  
 NDM\_SRV.EXE file 21  
 ndm\_startup.com 21  
 NDM\_STARTUP.COM file 21  
 NDM\_TIMESTAMP.EXE file 22  
 NDM\_TYPE.TLB file 22  
 NDM\_XLATE.TLB file 22  
 ndmmsg.exe 21  
 NDMMSG.EXE file 21  
 NDMUI command, adding to the system command  
     table 18  
 ndmui.cld 21  
 NDMUI.CLD file 21  
 ndmui.exe 21  
 NDMUI.EXE file 21  
 ndmui.hlb 21  
 NDMUI.HLB file 21  
 NETDEF.COM command procedure 64  
 NETFILE.FDL file 22  
 NETMAP commands 64  
 NETMBX privilege 12, 24  
 network definitions 63  
 network map  
     adding records 66  
     and adjacent node records 63  
     definition 63

network map (continued)  
     DELETE commands 68  
     deleting records 68  
     editing 34  
     examples  
         DELETE commands 69  
         UPDATE commands 71  
     INSERT commands 66  
     parameters  
         ADJACENT\_NODE 65  
         APPLID 65  
         COMM\_PATH 67  
         DECNET.NODE 68  
         DECNET.OBJ 65, 68  
         list of 65  
         LOCAL\_NODE 65  
         NAME 65, 67  
         PARSESS 66, 67  
         TCP.HOSTNAME 68  
         TCP.PORTNUM 68  
         TYPE 66, 67  
     qualifiers  
         /DEFCONN\_MODE 68  
         /SERVER 66, 70  
     sample commands for creating 64  
     types of records 64  
     UPDATE commands 69  
     updating records 69  
 notational conventions  
     bold letters 9  
     brackets 9  
     lowercase letters 9  
     miscellaneous 9  
     underlined letters 9  
     uppercase and lowercase letters 9  
     uppercase letters 9  
     vertical bars 9

## O

OPER privilege, oper privilege 12, 24

## P

PARSESS network map parameter 66, 67  
 privileges 12

## R

remote servers  
     defining 31



remote servers (continued)  
 example 31

replacing the existing Connect:Direct for OpenVMS  
 software 38

rights identifiers, granting 25

## S

Secure+ Option 64

server files  
 CKPTFILE.FDL 21  
 CONFIGSRV.COM 21  
 INITPARMS.DAT 21  
 NDM\_BLD\_XLATE.EXE 21  
 NDM\_DEFAULT.XLT 21  
 NDM\_SAMPSRC.TLB 21  
 NDM\_SMGR.EXE 21  
 NDM\_SRV.EXE 21  
 NDM\_TIMESTAMP.EXE 22  
 NDM\_TYPE.TLB 22  
 NDM\_XLATE.TLB 22  
 NETFILE.FDL 22  
 SETPRIV.COM 22  
 SETUPRIGHTS.COM 22  
 TCQE.FDL 22  
 TCQH.FDL 22

server installations  
 configuring 26  
 overview 11, 15, 23

/SERVER qualifier 66, 70

server.dat 21

SERVER.DAT file 21

servers, defining remote 31

SETPRIV.COM file 22

SETUPRIGHTS command procedure 24

SETUPRIGHTS.COM file 22, 24

shutdown procedure  
 for a cluster environment (alternate) 39  
 for a cluster environment (system-wide) 33  
 for a system-wide environment  
 client-only installations 33  
 server installations 32  
 for an alternate environment 39

shutting down Connect:Direct for OpenVMS, for a cluster  
 environment (system-wide) 33

shutting down Connect:Direct for OpenVMS, for a cluster  
 environment (alternate) 39

starting Connect:Direct for OpenVMS  
 for a cluster environment (system-wide) 32  
 for a system-wide environment, client-only  
 installations 33

starting Connect:Direct for OpenVMS, for a cluster  
 environment (alternate) 39

starting Connect:Direct for OpenVMS, for a system-wide  
 environment, server installations 31

starting Connect:Direct for OpenVMS, for an alternate  
 environment 38

startup procedure  
 for a cluster environment (alternate) 39  
 for a cluster environment (system-wide) 32  
 for a system-wide environment  
 client-only installations 33  
 server installations 31  
 for an alternate environment 38

statistics file  
 archiving 59  
 archiving daily, sample 61  
 archiving on demand 60  
 sample 60  
 deleting 59

statistics file records  
 deferring 59  
 message window 60  
 ndm\_stat\_writer\$setparams.com 60  
 parameter  
 ndm\$\$stat\_message\_window\_priority 60  
 ndm\$\$stat\_message\_window\_size 59  
 ndm\$\$stat\_process\_priority 59

stopping Connect:Direct for OpenVMS  
 for a system-wide environment, client-only  
 installations 33

stopping Connect:Direct for OpenVMS, for a system-  
 wide environment, server installations 32

stopping Connect:Direct for OpenVMS, for an alternate  
 environment 39

supported connectivity 7

SYLOGIN.COM file 59

SYSGBL privilege 12, 24

SYSLCK privilege 12, 24

SYSNAM privilege 12, 24

SYSPRV privilege 12, 24

system manager account 25

system-wide environment

- general 26
- server installations 32

starting Connect:Direct for OpenVMS

- client-only installations 33
- server installations 31

stopping Connect:Direct for OpenVMS

- client-only installations 33
- server installations 32

## T

target directory, for Connect:Direct for  
OpenVMS 18, 19, 20

TCP.HOSTNAME 68

TCP.PORTNUM 68

TCQE.FDL file 22

TCQH.FDL file 22

testing the installation 34

TMPMBX privilege 12, 24

translation table utility

- general 56
- invoking 56
- sample 58

TYPE network map parameter 66, 67

## U

underlined letters, notational conventions 9

update adjacent\_node command, TCP/IP protocol 70

UPDATE NETMAP ADJACENT\_NODE command 70

UPDATE NETMAP LOCAL\_NODE command 69

upgrade considerations

- for a cluster environment

- adding the NDMUI command, 19
  - general 15

- for a cluster environment (alternate)

- shutting down Connect:Direct for OpenVMS 39
  - starting Connect:Direct for OpenVMS 39

- for a cluster environment (system-wide)

- shutting down Connect:Direct for OpenVMS 33
  - starting Connect:Direct for OpenVMS 32

- planning 37

uppercase and lowercase letters, notational  
conventions 9

uppercase letters, notational conventions 9

utilities

- message

- displaying messages 56

- general 55

- inserting messages 56

- translation table

- general 56

- invoking 56

- sample 58

## V

verifying the installation 20

vertical bars, notational conventions 9

VMSINSTAL command procedure, invoking 17

## W

WORLD privilege 12, 24