VSI OpenVMS

VSI TCP/IP Version 10.6 Installation, Upgrade, and Quick Start Guide

Document Number: DO–DVTQSG–01A

Publication Date: August 2019

This document describes how to install, upgrade, and initially configure VSI TCP/IP for OpenVMS.

Revision Update Information: This guide supercedes the VSI TCP/IP Installation and Quick Start Guide, Version 10.5

Operating System and Version: VSI OpenVMS Integrity Version 8.4-2L1 or higher

Software Version: VSI TCP/IP for OpenVMS Version 10.6

VMS Software, Inc. (VSI)
Bolton, Massachusetts, USA

VSI TCP/IP Version 10.6 Installation, Upgrade, and Quick Start Guide:

Legal Notice

The VSI OpenVMS documentation set is available on DVD.

Preface

1. Introducing VSI TCP/IP Version 10.6

VSI TCP/IP for OpenVMS Version 10.6 (hereafter referred to as VSI TCP/IP) is an OpenVMS TCP/IP stack product produced and supported by VMS Software, Inc. It is based on the MultiNet TCP/IP stack produced by Process Software.

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

2. Prerequisites

Please note the following prerequisites for VSI TCP/IP:

- For VSI TCP/IP V10.6 you can perform a fresh installation or you can upgrade from VSI TCP/IP V10.5.
- With VSI TCP/IP V10.6 you can import and convert your existing MultiNet configuration files so that they can be used by VSI TCP/IP.
- You can run VSI TCP/IP on VSI OpenVMS Integrity Version 8.4-2L1 or higher only.
- VSI TCP/IP requires the use of ODS-5 system disks. If you attempt an installation on a system disk that is not ODS-5, you will see a message similar to the following:

```
Configuring VSI I64VMS VSI_TCPIP V10.6: VSI TCP/IP for OpenVMS IA64

    ©Copyright 2019 VMS Software, Inc.

VSI TCP/IP requires installation on an ODS-5 system disk.

The disk on which you are installing, $1$DGA150:, is not an ODS-5 disk.

        Current IP stack:  TCP/IP Services
                 Running:  yes
```

- VSI TCP/IP requires the installation of an OpenVMS patch kit, VMS842L1I_CLUCONFIG-V0100.PCSI$COMPRESSED, which is distributed in the same directory as the VSI TCP/IP kit. If this patch kit is not installed first, or at the same time as VSI TCP/IP, the installation of VSI TCP/IP will fail.

3. Intended Audience

The audience for this manual is network administrators and system managers, who are familiar with OpenVMS systems and TCP/IP networking basics. For a description of VSI TCP/IP and network concepts, see Appendix A.

Users of this manual are expected to obtain and reference any additional documentation specific to their hardware. Users are expected to know how to identify the various devices on their system and be familiar with the console and console commands.

4. VSI TCP/IP Support

VSI supports VSI TCP/IP running on VSI OpenVMS Integrity Version 8.4-2L1 or higher only. Please contact your support channel for help with this product.

5. VSI Encourages Your Comments

You may send comments or suggestions regarding this manual or any VSI document by sending electronic mail to the following Internet address: `<docinfo@vmssoftware.com>`.

6. Typographical Conventions

The following conventions are used in this manual:

Convention	**Meaning**
Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key (*x*) or a pointing device button.
`...`	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one. In installation or upgrade examples, parentheses indicate the possible answers to a prompt, such as: `Is this correct? (Y/N) [Y]`
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for directory specifications and for a substring specification in an assignment statement. In installation or upgrade examples, brackets indicate the default answer to a prompt if you press Enter without entering a value, as in: `Is this correct? (Y/N) [Y]`
|	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.

Convention	**Meaning**
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold type	Bold type represents the name of an argument, an attribute, or a reason. Bold type also represents the introduction of a new term.
italic type	Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Example`	This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies website addresses, UNIX command and pathnames, PC-based commands and folders, and certain elements of the C programming language.
`-`	A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.
numbers	All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

Chapter 1. Read this Guide First

This guide provides an overview of the product and the VSI TCP/IP documentation set. VSI encourages users to review this manual before installing and configuring VSI TCP/IP. There are significant differences between previous OpenVMS network stacks and the new VSI TCP/IP stack. These differences are most apparent in the installation and configuration procedures. However, all users will find this guide helpful to get VSI TCP/IP up and running quickly.

Chapter 2 provides instructions for installing, setting up your initial configuration, and upgrading VSI TCP/IP. It also includes the instructions to switch between VSI TCP/IP, TCP/IP Services, and MultiNet.

Chapter 3 provides important information to start basic network services.

1.1. Supported OpenVMS Versions

You can run VSI TCP/IP on VSI OpenVMS Integrity Version 8.4-2L1 or higher only.

1.2. VSI TCP/IP Documentation

The following table lists and describes the VSI TCP/IP documentation set.

Table 1.1. VSI for OpenVMS Documentation

Title	Description
VSI TCP/IP Version 10.6 Installation, Upgrade, and Quick Start Guide	A new manual that helps you quickly install, configure, and upgrade VSI TCP/IP.
VSI TCP/IP for OpenVMS User's Guide	This manual is intended for anyone who will be using VSI TCP/IP. The appendices in this document contain user command reference sections for network services such as FTP, TELNET, and TFTP.
VSI TCP/IP for OpenVMS Administrator's Guide: Volume I	*Volume I* provides information about configuring basic VSI TCP/IP services.
VSI TCP/IP for OpenVMS Administrator's Guide: Volume II	*Volume II* provides information to configure FTP, DHCP, SNMP, Kerberos, and SSH. It also explains how to set up printer services and font servers.
VSI TCP/IP for OpenVMS Administrator's Reference	This document describes how to use the VSI TCP/IP user commands. Included are instructions for beginning users and command pages for advanced users.
VSI TCP/IP for OpenVMS Programmer's Reference	This manual is designed to get you started as an application programmer using VSI TCP/IP. Once you have installed VSI TCP/IP, you will find a number of example programs in the directory pointed to by the logical IP$EXAMPLES:.
VSI TCP/IP for OpenVMS Messages, Logicals, and DECnet Applications	This document describes system messages, logicals, error and status codes, and DECnet application services.

Title	Description
VSI TCP/IP for OpenVMS Software Product Description (SPD)	This document describes VSI TCP/IP for OpenVMS features and capabilities, as well as licensing and ordering information.
VSI TCP/IP Version 10.6 Release Notes	This document provides late-breaking product and documentation release notes.

Chapter 2. Installing, Setting up Initial Configuration, and Upgrading

This chapter describes the steps to install, set up an initial configuration, and upgrade VSI TCP/IP.

2.1. Installing VSI TCP/IP

Follow the steps in this section to perform a fresh installation of VSI TCP/IP.

If you install VSI TCP/IP on a system running MultiNet, you can import and convert the existing MultiNet configuration files to be used by VSI TCP/IP.

Note

VSI WBEM Service (WBEMCIM) V3.0-C180108 or later for VSI OpenVMS Integrity Servers is required for compatibility with VSI TCP/IP. VSI highly recommends that you install the WBEMCIM kit prior to installing VSI TCP/IP.

If WBEM Services V3.0-C180108 or later is not installed, the following messages are displayed by the WBEM services configuration, startup, and shutdown procedures.

```
LCKHVN>> @sys$startup:wbem_services$startup
%SYSTEM-F-ABORT, abort
%RUN-S-PROC_ID, identification of created process is 23800460
%SYSTEM-F-ABORT, abort
%WBEMCIM-I-SERVERWAIT, Waiting for CIMServer to start. 180 seconds
 remaining...
%SYSTEM-F-ABORT, abort
%WBEMCIM-I-SERVERWAIT, Waiting for CIMServer to start. 170 seconds
 remaining...
```

1. **Log in to your SYSTEM account**.

   The installation procedure copies files onto the system disk. You must be logged in to the SYSTEM account (or another fully privileged account) to perform the installation.

2. **Gather information for the installation**.

   During installation, you *must* configure at least one network device during the initial configuration phase. Gather the required information before installing the PCSI kit. The following table provides a checklist for your information.

Table 2.1. IP Transport Parameter Checklist

Parameter Name	Description	Your Value
Internet host name	The name by which your system will be known. If you plan to use DNS to resolve host names, you must use the fully qualified host name supplied by your network administrator or Internet access provider (for example, ZKO3.EXAMPLES.COM).	

Parameter Name	Description	Your Value
IP address	The dotted-decimal representation of your system's IP address. Obtain your IP address from your network administrator or Internet access provider.	
Subnet mask	The dotted-decimal representation of a 32-bit mask that determines the network portion of your IP address to allow your network to be subdivided into multiple network segments.	
Default route	The dotted-decimal representation of the IP address of the router to which IP packets are sent when there is no route to the destination host or network in your system's routing table.	
Use DNS	The answer to the question, "Do you want to use DNS to resolve host names?" If you answer *Yes* to this question, you must know the address of at least one, and up to three, name servers. It is possible that you do not have an internet connection but you want to set up DNS that is local to your site.	Yes | No
Timezone	The standard abbreviation for your local timezone (For example: EST, CST, MST, or PST). Enter your local timezone, or, if your system clock is not in the local timezone, enter the timezone your system clock uses. When prompted for your timezone, type a question mark (?) to see a list of valid timezone abbreviations. In the US, switching between Standard Time and Daylight Saving Time occurs automatically.	

3. **Read the Release Notes**.

   The *VSI TCP/IP Release Notes* contains important information about this release. You can access the *Release Notes* by entering:

   ```
   $ PRODUCT EXTRACT RELEASE_NOTES
   ```

   After the installation you can access them in SYS$HELP:VSI_TCPIP106.RELEASE_NOTES.

4. **Check OpenVMS version**.

   Ensure your system is running VSI OpenVMS Integrity Version 8.4-2L1 or higher. If you are not running this operating system version, you *must* upgrade before installing VSI TCP/IP. Enter the following command:

   ```
   $ SHOW SYSTEM/NOPROCESS
   ```

5. **Back up your system disk**.

   Make a backup copy of your system disk using Option #8 on the installation kit menu, or make a boot drive using @SYS$SYSTEM:I64VMS$PCSI_INSTALL_MIN.COM on your system.

6. **Check available disk space**.

   To determine how much free space is available on your system disk, use the following command:

   ```
   $ SHOW DEVICE SYS$SYSDEVICE
   ```

   The information displayed includes the free space on the disk. You should have at least 500K free blocks, or 245 MB, for this installation. This and other important information is included in the *VSI TCP/IP for OpenVMS Software Product Description (SPD)*.

7. **Ask other users to log off**.

   Make sure other users log off the system before you start the installation or before you modify any system parameters, as you will need to reboot the system. You can use the following command to notify users to save their work and log off.

   ```
   $ REPLY/ALL "shutdown_announcement"
   ```

8. **Update system parameters**.

   If you are installing VSI TCP/IP on your system for the first time, you need to modify the values in SYS$SYSTEM:MODPARAMS.DAT. If the following parameters are not in SYS$SYSTEM:MODPARAMS.DAT, you will need to add them with the recommended values shown below:

   ```
   MIN_GH_EXEC_DATA = 1500
   MIN_GH_EXEC_CODE = 4200
   ```

   After adding these values, enter the following AUTOGEN command. You must run this command to ensure that the new system parameters are set. These changes will take effect when you reboot the system at the end of the installation procedure.

   ```
   $ @SYS$UPDATE:AUTOGEN SAVPARAMS SETPARAMS FEEDBACK
   ```

Note

If you are using host tables instead of DNS, check the size of the NETWORK_DATABASE and HOSTTBLUK.DAT files. Increase the ADD_GBLPAGES statement value by 1 for each disk block used by these files. Increase the ADD_GBLSECTIONS value by 2, one for each file. *However, these files are not present until AFTER you install VSI TCP/IP.*

9. **Verify the location of the DCLTABLES.EXE file**.

   The installation procedure expects the DCLTABLES.EXE file to be in the SYS$COMMON:[SYSLIB] directory.

Note

If you *do* have a local copy of DCLTABLES.EXE, you need to check with your system manager before proceeding with the installation.

10. **Locate your PCSI kit file**.

    If the file type is ZIPEXE, run the file to extract the PCSI files.

11. **Make sure that the VSI I64VMS VMS842L1I_CLUCONFIG V1.0 patch kit is in the same directory as the VSI TCP/IP kit**.

    You *must* install the patch kit at the same time or prior to installing VSI TCP/IP. If you haven’t already installed the patch kit, you must choose both kits as shown in Step 13.

Important

If you choose not to install the CLUCONFIG V1.0 patch kit, the VSI TCP/IP 10.6 installation will fail with the following system messages:

```
%IP-F-SFTREF, Required patch not installed

 ------------------------------------------------------------------------
      VSI TCP/IP V10.6 will not be installed as it requires the OpenVMS
      patch kit VMS842L1I_CLUCONFIG.

      Please install this patch kit before you install VSI TCP/IP V10.6.

      The patch kit is distributed with VSI TCP/IP V10.6 in the same
      directory as the VSI TCP/IP V10.6 kit.

 ------------------------------------------------------------------------

Portion done: 0%
Recovery pass starting ...
Recovery pass concluded
%PCSI-E-S_OPFAIL, operation failed
%PCSIUI-E-ABORT, operation terminated due to an unrecoverable error
 condition
```

12. **Execute the PCSI kit with the following command**:

    ```
    $ PRODUCT INSTALL * /SOURCE=[directory_that_contains_PCSI_kit]
    ```

    An example output of executing the command is shown below.

    ```
    1 - VSI I64VMS VMS842L1I_CLUCONFIG V1.0   Patch (remedial update)
    2 - VSI I64VMS VSI_TCPIP V10.6            Layered Product
    ? - Help
    E - Exit
    Choose one or more items from the menu:
    ```

13. **Specify the item number that corresponds to the CLUCONFIG V1.0 patch kit and the VSI TCP/IP kit**.

    In this case, enter 1, 2. The PCSI procedure will now execute and will display system messages and prompt you for answers to specific questions.

```
Performing product kit validation of signed kits ...

The following products have been selected:
    VSI I64VMS VMS842L1I_CLUCONFIG V1.0    Patch (remedial update)
    VSI I64VMS VSI_TCPIP V10.6             Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

Configuring VSI I64VMS VMS842L1I_CLUCONFIG V1.0: VSI OpenVMS V8.4-2L1
 CLUCONFIG V1.0
   .
   .
   .
Configuring VSI I64VMS VSI_TCPIP V10.6: VSI TCP/IP for OpenVMS IA64

    © Copyright 2019 VMS Software, Inc.

        Current IP stack:  TCP/IP Services
                 Running:  yes
         Cluster over IP:  no

Do you want VSI TCP/IP to be your default IP stack? [YES]
```

Important

VSI TCP/IP requires that it be installed on an ODS-5 system disk only. If you try to install the kit on an ODS-2 disk, you will receive the following messages. The installation then terminates.

```
Configuring VSI I64VMS VSI_TCPIP V10.6: VSI TCP/IP for OpenVMS IA64

    © Copyright 2019 VMS Software, Inc.

VSI TCP/IP requires installation on an ODS-5 system disk.
The disk on which you are installing, DISK$DKA400:, is not an ODS-5 disk.
```

Neither kit has any configuration options. The following message will be displayed for each product:

```
* This product does not have any configuration options.

Execution phase starting ...

Do you want to continue? [YES] Yes

The following product will be installed to destination:
    VSI I64VMS VMS842L1I_CLUCONFIG V1.0     DISK$HUDSON:[VMS$COMMON.]
    VSI I64VMS VSI_TCPIP V10.6              DISK$HUDSON:[VMS$COMMON.]
Portion done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%
```

14. **If you have MultiNet installed on your system, you can import and convert your existing MultiNet configuration files to be used by VSI TCP/IP**.

    If you do not have a MultiNet configuration on your system, skip this step and proceed with Step 15.

    If you have MultiNet installed, the following system messages will be displayed:

```
    MultiNet is installed on your system.

Your MultiNet configuration data can be imported for use by VSI TCP/IP.
You can import this data now, or after the installation completes
by running

        $ @SYS$MANAGER:IP$IMPORT_MULTINET_CONFIGURATION


Do you want to import your MultiNet configuration now? [YES]

    This procedure will convert MultiNet configuration data to
    VSI TCP/IP format on these nodes:

        System Root    Host Name
        -----------    ---------
        SYS0           MYNODE

Do you want to continue? [YES]

  Copying cluster-common configuration files from
  [MULTINET.I64_COMMON.MULTINET] ...
   .
   .
   .

  Copying node-specific configuration files from
  [MULTINET.MYNODE.MULTINET] ...
   .
   .
   .

  Converting files ...
   .
   .
   .

  Updating services file ...
    SYS$COMMON:[IP.CONFIG]SERVICES.MASTER_SERVER

  Creating site-specific startup file ...
    SYS$STARTUP:IP$SYSTARTUP.COM

%IP-I-SUCCESS, MultiNet configuration data converted

Installation continuing ...
```

The MultiNet configuration files are copied and converted to be used by VSI TCP/IP everywhere on the system where you had a MultiNet configuration. Once you start VSI TCP/IP, your system will be running VSI TCP/IP with the same environment you were running with MultiNet.

After the MultiNet configuration data is converted, follow the installation instructions and proceed with Step 16.

Note

If you answer *No* to either of the questions in the previous example, the installation will proceed without converting your MultiNet configuration data.

You can perform the conversion of MultiNet configuration files after the VSI TCP/IP installation is complete by running the following command:

```
$ @SYS$MANAGER:IP$IMPORT_MULTINET_CONFIGURATION
```

15. **Configure a network interface**.

    The following user inputs are for example purposes only.

```
    Before starting VSI TCP/IP, you must define at least one interface.

Do you want to configure a network interface now? [YES]

Enter your fully-qualified Internet host name : ZKO3.EXAMPLES.COM

    VMS                  TCP/IP
    Device  Link  Speed  Interface  Address       Subnet        IPCI
    ------  ----  -----  ---------  ------------  ------------  ----
    EIA0    Up    1000                                          no
    EWA0    Up    1000                                          no

Select interface to configure [EIA0] : ewa0
Enter the Internet (IP) Address for interface EWA0: 10.10.108.13
Enter the Subnet mask for interface EWA0: 255.255.254.0
Enter the Internet (IP) Address of your default route: 10.10.108.1
Do you want to use DNS to resolve host names [YES]?

Enter one nameserver address at a time, and press <RETURN> when you are
done
 Enter the IP Address for your 1st nameserver : 10.10.109.253
 Enter the IP Address for your 2nd nameserver : 10.10.101.239
 Enter the IP Address for your 3rd nameserver :

Enter your local timezone [EST] :

            Interface:  EWA0
            Host Name:  ZKO3.EXAMPLES.COM
           IP Address:  10.10.108.13
          Subnet Mask:  255.255.254.0
        Default Route:  10.10.108.1
              Use DNS:  yes
       Run DNS Server:  no
          Nameservers:  10.10.109.253
                        10.10.101.239
            Time Zone:  EST

Are you satisfied with these values? [YES]

Additional interfaces can be added at any time with this command:

    $ IP CONFIGURE
```

Answering *No* to the question `Do you want to use DNS to resolve host names [YES]?` will skip to the timezone prompt.

You can find a complete listing of timezones in *VSI TCP/IP for OpenVMS Administrator's Guide: Volume I*.

Note

If you choose not to configure an interface, or you do not complete the configuration, you will receive the following message:

```
Before starting VSI TCP/IP, you must configure a network interface
with this command:

    $ @IP$:CONFIGURE
```

```
When the installation completes, you must replace the startup of your
existing IP stack in your system startup procedure with this command:

       $ @SYS$STARTUP:IP$STARTUP

    NOTE:  This procedure will start VSI TCP/IP or any other IP stack
    (TCP/IP Services, MultiNet, or TCPware) that is installed on your
    system.

    If you would like to be able to switch back and forth between your
    current IP stack and VSI TCP/IP, run this command each time you want
    to switch:

        $ @SYS$MANAGER:IP$SET_STACK

    A reboot is required to switch between network stacks.

Installation continuing ...
...100%

The following products have been installed:
    VSI I64VMS VMS842L1I_CLUCONFIG V1.0    Patch (maintenance update)
    VSI I64VMS VSI_TCPIP V10.6             Layered Product

VSI I64VMS VMS842L1I_CLUCONFIG V1.0: VSI OpenVMS V8.4-2L1 CLUCONFIG V1.0

    Release notes for the VMS842L1I_CLUCONFIG-V0100
    ECO kit are available in SYS$COMMON:[SYSHLP]

    VSI TCP/IP Release Notes have been provided in SYS$HELP
```

VSI TCP/IP is now installed.

16. **Disable other IP stack startup commands and add the VSI TCP/IP startup command to your system startup procedure**.

    Prior to rebooting, ensure that you comment out all commands in your system startup procedure, SYS$MANAGER:SYSTARTUP_VMS.COM, that start IP network stacks, such as:

    ```
    $ @TCPIP$STARTUP
    $ @START_MULTINET
    $ @TCPWARE:STARTNET
    ```

    You must also comment out any corresponding START/NETWORK command for your IP network stack.

    Add the following VSI TCP/IP startup command:

    ```
    $ @SYS$STARTUP:IP$STARTUP
    ```

    The IP$STARTUP command is used to start the default IP stack — whether it is VSI TCP/IP or a different installed IP stack — when the system is rebooted. Once rebooted, you can switch between IP stacks using the IP$SET_STACK procedure, which will require a reboot to complete the switch. See Section 2.4 for more information about switching between installed IP stacks.

17. **Reboot the system by entering**:

    ```
    $ @SYS$SYSTEM:SHUTDOWN
    ```

2.2. Establishing an Initial Configuration

This section describes the steps to set up an initial network configuration for VSI TCP/IP if it was not done or completed during the installation. Follow these steps only if you have not previously configured a network device, or if you want to add additional devices. If you wish to modify your network device(s), use the IP CONFIGURE command.

1. **Use CONFIGURE.COM to add your *first* device**.

   Configure the VSI TCP/IP IP transport for one standard network interface by running the CONFIGURE.COM command procedure. Enter:

   ```
   $ @IP$:CONFIGURE

   VSI TCP/IP Initial Configuration

   Enter your fully-qualified Internet host name: ZKO3.EXAMPLES.COM

       VMS                 TCP/IP
       Device  Link  Speed  Interface  Address       Subnet        IPCI
       ------  ----  -----  ---------  ------------  ------------  ----
       EIA0    Up    1000                                          no
       EWA0    Up    1000                                          no

   Select interface to configure [EIA0] : ewa0
   Enter the Internet (IP) Address for interface EWA0: 10.10.108.13
   Enter the Subnet mask for interface EWA0: 255.255.254.0
   Enter the Internet (IP) Address of your default route: 10.10.108.1
   Do you want to use DNS to resolve host names [YES]?
   ```

```
Enter one nameserver address at a time, and press <RETURN> when you are
done
 Enter the IP Address for your 1st nameserver : 10.10.109.253
 Enter the IP Address for your 2nd nameserver : 10.10.101.239
 Enter the IP Address for your 3rd nameserver :

Enter your local timezone [EST] :

            Interface:  EWA0
            Host Name:  ZKO3.EXAMPLES.COM
           IP Address:  10.10.108.13
          Subnet Mask:  255.255.254.0
        Default Route:  10.10.108.1
              Use DNS:  yes
       Run DNS Server:  no
          Nameservers:  10.10.109.253
                        10.10.101.239
            Time Zone:  EST

Are you satisfied with these values? [YES]
```

Answering *No* to the question `Do you want to use DNS to resolve host names [YES]?` will skip to the timezone prompt.

2. **If you want to, you can add additional interfaces at this time**.

   To define additional network interfaces, enter:

   ```
   $ IP CONFIGURE
   ```

   Note that if you add a new interface, you must reboot the system. Please consider adding any additional interfaces now to avoid unnecessary system reboots.

   The system starts the Network Configuration Utility and will display messages similar to the following example:

   ```
   VSI TCP/IP Network Configuration Utility
   [No checking is done against the MAXIMUM configuration]
   [Reading in configuration from IP$:NETWORK_DEVICES.CONFIGURATION]

   NET-CONFIG>
   ```

3. **Use NET-CONFIG to show a list of configured devices**.

   At the NET-CONFIG prompt, use the SHOW command to display a list of the configured interfaces.

   ```
   NET-CONFIG>SHOW
   Interface                            Adapter  CSR Address  Flags/Vector
   ---------                            -------  -----------  ------------
   se0      (Shared VMS Ethernet/FDDI)   -NONE-    -NONE-        -NONE-
      [TCP/IP: 10.10.110.2, IP-SubNet: 255.255.254.0 (/-1),
       IP-Broadcast: 10.10.111.255]
      [VMS Device: EIC0, Link Level: Ethernet]

   Official Host Name:             ia64.eng.vmssoftware.com
   Default IP Route:               10.10.108.1
   Domain Nameservers:             10.10.109.253
   ```

```
                                 10.10.101.239
Timezone:                        EST
Timezone Rules:                  US/EASTERN
Load UCX $QIO driver:            TRUE
Load PWIP (Pathworks) driver:    TRUE
```

The following example shows the command and system messages when device `se1` is added.

```
NET-CONFIG> ADD se1
[Adding new configuration entry for device "se1"]
VMS Device [XEA0]: eid0
Link Level Encapsulation Mode [ETHERNET]:
BSD Trailer Encapsulation: [DISABLED]
IP Address [NONE]: 10.10.110.2
IP SubNet Mask [NONE]: 255.255.254.0
Non-Standard IP Broadcast Address [10.10.111.255]: none
DHCP CLIENT [DISABLED]:
Jumbo Frames [DISABLED]: enabled
IPv6 on this interface [DISABLED]:
[se1 (Shared VMS Ethernet/FDDI): Csr=NONE, Flags=%X0]
NET-CONFIG> EXIT
```

See the *VSI TCP/IP for OpenVMS Administrator's Guide: Volume I* for more information on the IP CONFIGURE command.

4. Comment out other IP stack startup commands and add the following VSI TCP/IP startup command to your system startup procedure SYS$MANAGER:SYSTARTUP_VMS.COM:

   ```
   $ @SYS$STARTUP:IP$STARTUP
   ```

   The IP$STARTUP command is used to start the default IP stack — whether it is VSI TCP/IP or a different installed IP stack — when the system is rebooted. Once rebooted, you can switch between IP stacks using the IP$SET_STACK procedure, which will require a reboot to complete the switch. See Section 2.4 for more information about switching between installed IP stacks.

5. **Reboot the system by entering**:

   ```
   $ @SYS$SYSTEM:SHUTDOWN
   ```

2.3. Upgrading to VSI TCP/IP Version 10.6

Follow the steps in this section to upgrade VSI TCP/IP V10.5 to V10.6.

Note

The upgrade procedure requires you to shut down VSI TCP/IP on your system. You can choose to do this before starting the upgrade by entering the following command:

```
$ @SYS$MANAGER:IP$SHUTDOWN
```

or during the upgrade when you will be prompted to shut down VSI TCP/IP.

1. **Make sure you are logged in to your SYSTEM account**.

   The upgrade procedure will copy and remove files from the system disk. You must be logged in to the SYSTEM account (or another fully privileged account) to perform the upgrade to VSI TCP/IP V10.6.

2. **Install VSI TCP/IP V10.6 by entering**:

```
$ PRODUCT INSTALL VSI_TCPIP/SOURCE=[directory_location]
   .
   .
   .
The following product has been selected:
    VSI I64VMS VSI_TCPIP V10.6           Layered Product

Do you want to continue? [YES]

Configuration phase starting ...

You will be asked to choose options, if any, for each selected product
 and for any products that may be installed to satisfy software
 dependency requirements.

Configuring VSI I64VMS VSI_TCPIP V10.6: VSI TCP/IP for OpenVMS IA64

    © Copyright 2019 VMS Software, Inc.

        Current IP stack:  VSI TCP/IP
                 Running:  yes
         Cluster over IP:  no

VSI TCP/IP is running on node MYNODE and must be shut down before
 installing.

Do you want to shut down VSI TCP/IP and continue the installation? [YES]
```

Important

The installation procedure determines where on the system VSI TCP/IP needs to be shut down and prompts you to do it. Choose the default *Yes* for the following question to proceed with the installation. Otherwise, the installation terminates.

```
Do you want to shut down VSI TCP/IP and continue the installation? [YES]
```

The following message will be displayed:

```
Waiting for VSI TCP/IP to shut down -
%IP-S-SHUTDONE, VSI TCP/IP shutdown completed at 15-MAY-2019 07:45:41.55

* This product does not have any configuration options.

Execution phase starting ...

%-PCSI-W-VERINS, the installation of product VSI I64VMS VSI_TCPIP V10.6
%-PCSI-W-VERREM, will remove current product VSI I64VMS VSI_TCPIP V10.5

Do you want to continue? [YES]

The following product will be installed to destination:
    VSI I64VMS VSI_TCPIP V10.6           DISK$HUDSON:[VMS$COMMON.]
The following product will be removed from destination:
    VSI I64VMS VSI_TCPIP V10.5           DISK$HUDSON:[VMS$COMMON.]
```

```
Portion done:
 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%...100%

The following product has been installed:
    VSI I64VMS VSI_TCPIP V10.6           Layered Product
The following product has been removed:
    VSI I64VMS VSI_TCPIP V10.5           Layered Product

VSI I64VMS VSI_TCPIP V10.6: VSI TCP/IP for OpenVMS IA64

    VSI TCP/IP Release Notes have been provided in SYS$HELP.
```

3. **Reboot the system after installing VSI TCP/IP V10.6**.

   The following message will be displayed:

   ```
         A system reboot is required to use the new version of VSI TCP/IP.
   ```

   If you want to configure additional network interfaces, follow the instructions in Section 2.2 before entering the following command:

   ```
   $ @SYS$SYSTEM:SHUTDOWN
   ```

2.4. Switching TCP/IP Stacks

With the introduction of VSI TCP/IP, you may want to evaluate this new TCP/IP stack and compare it to TCP/IP Services, or another installed stack. To that end, VSI has provided a command procedure called SYS$MANAGER:IP$SET_STACK which allows you to switch between TCP/IP Services, VSI TCP/IP, MultiNet, or any others without having to modify your system startup procedure each time. IP$SET_STACK does not install or remove any products on your system. This procedure requires a system reboot for the changes to take effect.

Note

Multiple stacks can be installed on the system, although you can only run one at a time.

1. Prior to running IP$SET_STACK, ensure that you comment out all commands in your system startup procedure that start IP network stacks, such as:

   ```
   $ @TCPIP$STARTUP
   $ @START_MULTINET
   $ @TCPWARE:STARTNET
   ```

   You must also comment out the corresponding START/NETWORK command for your IP network stack, leaving only the following startup command:

   ```
   $ @SYS$STARTUP:IP$STARTUP
   ```

   The IP$STARTUP command is used to start the chosen IP stack — whether it is VSI TCP/IP or a different installed IP stack — when the system is rebooted.

2. Run the procedure. Enter:

   ```
   $ @SYS$MANAGER:IP$SET_STACK
   ```

This procedure displays a list of installed IP stacks, shows the IP stack currently running, and allows you to choose a different stack to run upon reboot.

```
Available network stacks:
        Name              Running   Address
        ----              -------   -------
    1   VSI TCP/IP        No
    2   TCP/IP Services   Yes       10.10.110.2

You can switch to this stack: VSI TCP/IP
Switch to stack number (E to exit) [1] :

    This system is currently running TCP/IP Services.
    You are setting this system to run VSI TCP/IP.

Do you want to save your changes? [YES]:

 ------------------------------------------------------------------------
     Before rebooting your system, you must remove all commands in your
     system startup procedure that start IP network stacks, such as
     @TCPIP$STARTUP, @START_MULTINET, @TCPWARE:STARTNET, or the
     corresponding START /NETWORK command for your IP network stack.

     Replace these commands with:

         $ @SYS$STARTUP:IP$STARTUP

     which will complete the switch of IP network stacks, and start the
     IP network stack of your choice, after rebooting.

 ------------------------------------------------------------------------

Do you want to reboot now? [YES]:
Changes take effect when the system is rebooted.
```

2.4.1. Customer Configuration Scenarios

This section provides instructions for switching between VSI TCP/IP, HPE TCP/IP Services, or MultiNet. For complete installation information, see Section 2.1.

Table 2.2. Switching Between Network Stacks

I am running…	Follow these directions…

TCP/IP Services or MultiNet within a cluster, and I want to install and switch to VSI TCP/IP.	1. Install VSI TCP/IP by entering:
	`$ PRODUCT INSTALL VSI_TCPIP`
	2. During the installation say you want this to be your default stack:
	`Do you want VSI TCP/IP to be your default IP stack? [YES]`
	3. Configure the first network interface during installation or later by entering:

I am running…	Follow these directions…

	`$ @IP$:CONFIGURE`
	4. Remove the old stack startup command from your system startup procedure and replace it with:
	`$ @SYS$STARTUP:IP$STARTUP`
	5. To run VSI TCP/IP, you must now reboot the system.

TCP/IP Services or MultiNet, have VSI TCP/IP installed, and I want to switch to VSI TCP/IP.	1. Prepare VSI TCP/IP to be run by entering: `$ @SYS$MANAGER:IP$SET_STACK` 2. To run VSI TCP/IP, reboot now or when you are ready.
VSI TCP/IP, and I want to switch to TCP/IP Services or MultiNet.	1. Prepare the other stack to be run by entering: `$ @SYS$MANAGER:IP$SET_STACK` 2. To run the other stack, reboot now or when you are ready. **Important** Do *not* remove IP$STARTUP from your system startup procedure or the VSI TCP/IP product from your system until your system is rebooted and IP$STARTUP has been run.

2.5. Installing VSI TCP/IP on a Common System Disk

Follow the steps in this section to install VSI TCP/IP on a common system disk. The benefit is that any node in the cluster that shares a common disk on which VSI TCP/IP has been installed can use VSI TCP/IP and any enabled service on the common disk.

1. Identify the name of the common system disk. In this case, the disk is $1$DGA5020:

```
STENIS>> SHOW DEVICE SYS$SYSDEVICE:

Device        Device   Error   Volume       Free      Trans  Mnt
Name          Status   Count   Label        Space     Count  Cnt
$1$DGA5020:   Mounted  0       TSTNEWSTACK  97.42GB   338    2
(STENIS)
```

2. Refer to Section 2.1 and follow the instructions to install VSI TCP/IP.

3. Once you are satisfied that you have installed and enabled your network devices, reboot the system by entering:

```
$ @SYS$SYSTEM:SHUTDOWN
```

Once the system has rebooted, VSI TCP/IP will be the default IP stack running on that system.

4. Log into each node in the cluster that is sharing the common disk and perform the following steps. In this example, BAYSTL is another node in the cluster that shares the common system disk with STENIS.

    a. Ensure that you are sharing the common system disk by entering the following command and compare the device names:

    ```
    $ SHOW DEVICE SYS$SYSDEVICE:
    ```

    b. Configure a network device on the current node (in this case, BAYSTL). Enter:

    ```
    BAYSTL> @SYS$STARTUP:IP$LOGICAL_NAMES
    BAYSTL> @IP$:CONFIGURE
    ```

    c. Follow the instructions in Section 2.2 and answer the prompts.

    d. Reboot the system by entering:

    ```
    $ @SYS$SYSTEM:SHUTDOWN
    ```

    Ensure that you boot the system using the common system disk.

Once it is rebooted, VSI TCP/IP will be running on the system. In addition, any network services, such as SSH or SNMP, already enabled and started on the first system, will be available on this system. You can enable additional services by following directions in Chapter 3.

2.6. Removing VSI TCP/IP and Saving the Existing Configuration

You can remove VSI TCP/IP by using the PCSI PRODUCT REMOVE command. Enter:

```
$ PRODUCT REMOVE *
```

A menu displays a list of products that you can remove from the system. For example:

```
1 - VSI I64VMS AVAIL_MAN_BASE V8.4-2L1 Layered Product
2 - VSI I64VMS CDSA V2.4-322A Layered Product
3 - VSI I64VMS SSL V1.4-503 Layered Product
4 - VSI I64VMS VMS V8.4-2L1 Operating System
5 - VSI I64VMS VSI_TCPIP V10.6 Layered Product
6 - All products listed above
? - Help
E - Exit

Choose one or more items from the menu: 5
The following product has been selected:
VSI I64VMS VSI_TCPIP V10.6 Layered Product

Do you want to continue? [YES]

The following product will be removed from destination:
VSI I64VMS VSI_TCPIP V10.6 DISK$STENISNEW:[VMS$COMMON.]
```

```
Portion done: 0%

Shutting down VSI TCP/IP ...

Connected to NETCONTROL server on "LOCALHOST"

server_name Network Control V10.6 at Thu 15-May-2019 10:14AM-GMT

VSI TCP/IP Master Server shutdown
```

- If you want to completely remove VSI TCP/IP from your system, answer the default *No* to the following question:

  ```
  Do you want to save your VSI TCP/IP configuration data? [NO]

  Deleting files..10%..20%..30%..40%..50%..60%..70%..80%..90%..100%

  The following product has been removed:
  VSI I64VMS VSI_TCPIP V10.6 Layered Product
  ```

- If you want to remove VSI TCP/IP, but save your existing VSI TCP/IP configuration data, answer *Yes* to that question.

  ```
  Do you want to save your VSI TCP/IP configuration data? [NO] yes

  Saving VSI TCP/IP configuration files ...
   Saving node-specific files to
   SYS$COMMON:[SYSMGR]IP$CONFIG_SPECIFIC.BCK ...
   Saving cluster-common files to
   SYS$COMMON:[SYSMGR]IP$CONFIG_COMMON.BCK ...
  ```

  You can use the saved configuration files to restore your VSI TCP/IP configuration when reinstalling VSI TCP/IP. For details on how to restore the saved VSI TCP/IP configuration data, see Section 2.7.

2.7. Restoring Your Previously Saved VSI TCP/IP Configuration

Follow the steps in this section to install VSI TCP/IP and restore the previously saved configuration.

Important

It is assumed that you have saved your VSI TCP/IP configuration data when you were removing the product. For details on how to save VSI TCP/IP configuration data, see Section 2.6.

1. Verify that the VSI TCP/IP configuration files have been saved on your system. If you have not saved the configuration files, you may have to restore them from a backup or you will have to reconfigure your VSI TCP/IP settings from scratch.

   The following example shows that multiple versions of the VSI TCP/IP configuration files have been saved. Note that the latest file versions will be used to restore the configuration.

   ```
   $ dir SYS$COMMON:[SYSMGR]IP$CONFIG_SPECIFIC.BCK

   Directory SYS$COMMON:[SYSMGR]
   ```

```
IP$CONFIG_SPECIFIC.BCK;2                IP$CONFIG_SPECIFIC.BCK;1

Total of 2 files.

$ dir SYS$COMMON:[SYSMGR]IP$CONFIG_COMMON.BCK

Directory SYS$COMMON:[SYSMGR]

IP$CONFIG_COMMON.BCK;2                  IP$CONFIG_COMMON.BCK;1

Total of 2 files.
```

2. Install VSI TCP/IP according to the instructions in Section 2.1.

Note

Do *not* configure any network interfaces during the installation. Answer *No* to the following question:

```
Do you want to configure a network interface now? [YES] no
```

3. After the VSI TCP/IP installation is complete, restore your previously saved VSI TCP/IP configuration data. Run the following command procedure:

```
$ @SYS$MANAGER:IP$RESTORE_CONFIGURATION

    This procedure will restore your previously-saved VSI TCP/IP
    configuration data that was performed during a PRODUCT REMOVE
    command.

    This data will overwrite your current configurations,
    if any, for all systems using this system disk.

Do you want to restore your saved VSI TCP/IP configuration data? [YES]

 Creating directories ...
 Restoring node-specific files ...
 Restoring cluster-common files ...

%IP-I-RESTORE, Configuration restored
```

4. Ensure that you comment out other IP stack startup commands and added the following VSI TCP/IP startup command to your system startup procedure SYS$MANAGER:SYSTARTUP_VMS.COM:

```
$ @SYS$STARTUP:IP$STARTUP
```

5. Reboot the system by entering:

```
$ @SYS$SYSTEM:SHUTDOWN
```

   System will boot with the VSI TCP/IP stack started and configured.

6. You can check your VSI TCP/IP configuration by entering:

```
$ IP CONFIG
VSI TCP/IP Network Configuration Utility
```

```
[No checking is done against the MAXIMUM configuration]
[Reading in configuration from IP$CONFIG:NETWORK_DEVICES.CONFIGURATION]
NET-CONFIG>SHOW

Interface                            Adapter  CSR Address  Flags/Vector
---------                            -------  -----------  ------------
se0      (Shared VMS Ethernet/FDDI)   -NONE-    -NONE-        -NONE-
   [TCP/IP: 10.10.10.10, IP-SubNet: 255.255.255.0 (/24),
    IP-Broadcast: 10.10.111.255]
   [VMS Device: EWA0, Link Level: Ethernet ]
   [IPv6 is enabled for this device]
   [IPv6 global address: 2001::4]
   [IPv6 mask length: 8]

Official Host Name:              ia64.eng.vmssoftware.com
Default IP Route:                10.10.111.1
Domain Nameserver:               10.10.101.239
Timezone:                        EST
Timezone Rules:                  US/EASTERN
Load UCX $QIO driver:            TRUE
Load PWIP (Pathworks) driver:    TRUE
NET-CONFIG>exit
[Configuration not modified, so no update needed]
$
```

Chapter 3. Enabling and Starting Network Services

Although VSI TCP/IP is installed, you must now configure the product to run any typical network services such as SSH, Telnet, or FTP. This chapter provides basic instructions to start and run services. For complete information about VSI TCP/IP network services and how to manage them, see *VSI TCP/IP for OpenVMS Administrator's Guide: Volumes I & II* .

3.1. Starting Typical Network Services

This section provides instructions for enabling and starting services on VSI TCP/IP.

3.1.1. Configuring DNS and Starting VSI TCP/IP

1. You must have a copy of the file `NAMED.CONF` in SYS$SPECIFIC:[IP.CONFIG]. VSI supplies the following template file:

   ```
   SYS$COMMON:[IP.CONFIG]NAMED_CONF.DEFAULT
   ```

   You should copy this file to the correct file name. Copying ensures that the template file is available for later use. Enter the following command:

   ```
   $ COPY/LOG IP$CONFIG:NAMED_CONF.DEFAULT IP$CONFIG:NAMED.CONF
   ```

   It is possible that the file IP$CONFIG:NAMED.CONF currently exists because you have previously configured VSI TCP/IP. If so, you may want to check the file to ensure that the settings meet your requirements.

2. Enable the DOMAINNAME service by entering:

   ```
   $ IP CONFIGURE/SERVER
   VSI TCP/IP Server Configuration Utility
   [Reading in configuration from IP$CONFIG:SERVICES.MASTER_SERVER]
   SERVER-CONFIG>ENABLE DOMAINNAME
   SERVER-CONFIG>Exit
   [Writing configuration to SYS$COMMON:[IP.CONFIG]SERVICES.MASTER_SERVER]
   ```

3. Start VSI TCP/IP. Enter the following command:

   ```
   $ @SYS$STARTUP:IP$STARTUP
   ```

4. Verify that DNS is working properly where "MYNODE" is the target server. Enter:

   ```
   $ IP
   _Operation: NSLOOKUP
   _HOSTNAME: MYNODE
   _SERVERNAME: DNSSVR
   Server:  DNSSVR.LAB.MYJOB.COM
   Address:  10.10.23.5

   Name:    MYNODE.LAB.MYJOB.COM
   Address:  10.10.25.9
   $
   ```

3.1.2. Configuring SSH

VSI recommends that you configure SSH from the SYSTEM account only. Non-system accounts do not provide sufficient privileges.

1. You must have a copy of the following SSH configuration files: SYS$SPECIFIC:[IP.CONFIG.SSH2]SSHD2_CONFIG. and SYS$SPECIFIC:[IP.CONFIG.SSH2]SSH2_CONFIG.

   Enter the following commands:

   ```
   $ COPY IP$CONFIG:SSHD2_CONFIG.TEMPLATE -
   _$ SYS$SPECIFIC:[IP.CONFIG.SSH2]SSHD2_CONFIG./LOG
   %COPY-S-COPIED, SYS$COMMON:[IP.CONFIG]SSHD2_CONFIG.TEMPLATE;1 copied to
   SYS$SPECIFIC:[IP.CONFIG.SSH2]SSHD2_CONFIG.;1 (10 blocks)

   $ COPY IP$CONFIG:SSH2_CONFIG.TEMPLATE -
   _$ SYS$SPECIFIC:[IP.CONFIG.SSH2]SSH2_CONFIG./LOG
   %COPY-S-COPIED, SYS$COMMON:[IP.CONFIG]SSH2_CONFIG.TEMPLATE;1 copied to
   SYS$SPECIFIC:[IP.CONFIG.SSH2]SSH2_CONFIG.;1 (5 blocks)
   ```

Note

The file names above do not have a file extension. Make sure that you copy the files exactly as documented in the previous example.

2. Enable SSH by entering:

   ```
   $ IP CONFIGURE/SERVER
   VSI TCP/IP Server Configuration Utility
   [Reading in configuration from IP$:SERVICES.MASTER_SERVER]
   SERVER-CONFIG>SELECT SSH
   [The Selected SERVER entry is now SSH]
   SERVER-CONFIG>ENABLE SSH
   SERVER-CONFIG>SET PARAMETERS
   You can now add new parameters for SSH.  An empty line terminates.
   Add Parameter: ENABLE-SSH2
   Add Parameter:
   [Service specific parameters for SSH changed]
   SERVER-CONFIG>SHOW/FULL SSH
   Service "SSH":
       INIT() = Merge_Image
       Program = "IP$SYSTEM:LOADABLE_SSH_CONTROL"
       Priority = 5
       Parameters = "ENABLE-SSH2"

   SERVER-CONFIG>RESTART
   Configuration modified, do you want to save it first ? [YES] yes
    [Writing configuration to SYS$COMMON:[IP.CONFIG]SERVICES.MASTER_SERVER]
   %RUN-S-PROC_ID, identification of created process is 21A0060F
   SERVER-CONFIG>Exit
   $
   ```

3. Restart the master server by entering:

   ```
   $ @IP$:START_SERVER RESTART
   ```

The system issues the following message:

```
%RUN-S-PROC_ID, identification of created process is 00000427
```

4. SSH is now configured, but because of a default client setting, you cannot make outbound connection without a *known key*. To remedy this condition, change the setting for `StrictHostKeyChecking` from `yes` to `ask`. Edit the file SYS$SPECIFIC:[IP.CONFIG.SSH2]SSH2_CONFIG. as follows. This results in a prompt when the system encounters a host key absent from the configuration.

```
## Crypto
#       Ciphers                         AnyStdCipher
#       MACs                            AnyStdMAC
        StrictHostKeyChecking           ask
#       StrictHostKeyChecking           yes
#       RekeyIntervalSeconds            3600
```

3.1.3. Starting Telnet

The following example shows how to start Telnet. Make sure you restart the Master Server after enabling this service.

```
$ IP CONFIGURE/SERVER
VSI TCPIP Server Configuration Utility
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>ENABLE TELNET
SERVER-CONFIG>EXIT
[Writing configuration to SYS$COMMON:[IP.CONFIG]SERVICES.MASTER_SERVER]

$ @IP$:START_SERVER RESTART
%RUN-S-PROC_ID, identification of created process is 0000041D
```

3.1.4. Starting FTP

The following example shows how to start FTP. Make sure you restart the Master Server after enabling this service.

```
$ IP CONFIGURE/SERVER
VSI TCPIP Server Configuration Utility
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>ENABLE FTP
SERVER-CONFIG>EXIT
[Writing configuration to SYS$COMMON:[IP.CONFIG]SERVICES.MASTER_SERVER]

$ @IP$:START_SERVER RESTART
%RUN-S-PROC_ID, identification of created process is 0000041D
```

3.1.5. Starting SNMP

```
$ IP CONFIGURE /SERVER
VSI TCP/IP Server Configuration Utility
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>SELECT SNMP
[The Selected SERVER entry is now SNMP]
SERVER-CONFIG>ENABLE SNMP
```

3.1.6. Enabling NFS and RPC Services

Enabling NFS requires the following services:

- NFS server
- RPCMOUNT mount server
- RPCQUOTAD quota server
- RPCLOCKMGR lock manager
- RPCSTATUS status monitor
- RPCPORTMAP RPC-protocol port mapper

See the following example:

Note

Before configuring NFS, you must create the required data files with the following commands:

```
$ IP CONFIGURE/NFS
    NFS-CONFIG> CREATE EXPORT
    NFS-CONFIG> CREATE GROUP
    NFS-CONFIG> CREATE PROXY
    NFS-CONFIG> EXIT
```

Without these files, OPCOM issues a fatal MOUNT error.

```
$ IP CONFIGURE/SERVER
VSI TCP/IP for OpenVMS Server Configuration Utility
[Reading in configuration from IP$:SERVICES.MASTER_SERVER]
SERVER-CONFIG>ENABLE NFS
SERVER-CONFIG>ENABLE RPCMOUNT
SERVER-CONFIG>ENABLE RPCQUOTAD
SERVER-CONFIG>ENABLE RPCPORTMAP
SERVER-CONFIG>ENABLE RPCLOCKMGR
SERVER-CONFIG>ENABLE RPCSTATUS
SERVER-CONFIG>RESTART
Configuration modified, do you want to save it first ? [YES] YES
[Writing configuration to
SYS$COMMON:[IP]SERVICES.MASTER_SERVER]
%RUN-S-PROC_ID, identification of created process is 0000017A
SERVER-CONFIG>EXIT
[Configuration not modified, so no update needed]
```

The NFS service requires additional configuration using the command:

```
$ IP CONFIGURE/NFS
```

To enable the NFS service after you make the configuration changes, you need to restart the master server by entering:

```
$ @IP$:START_SERVER RESTART
```

For more information about NFS, see *VSI TCP/IP for OpenVMS Administrator's Guide: Volume I* and *VSI TCP/IP for OpenVMS Administrator's Guide: Volume II*, *Section 12.2*.

Appendix A. VSI TCP/IP and Networking Overview

This chapter presents descriptions of VSI TCP/IP and general networking concepts.

A.1. Introduction

VSI TCP/IP provides applications, configuration tools, and programming libraries that make access to TCP/IP understandable and straight-forward.

VSI TCP/IP works on HPE Integrity servers running VSI OpenVMS Integrity 8.4-2L1 or later.

A.1.1. VSI TCP/IP for Users

With VSI TCP/IP, users can:

- Send electronic mail to and receive electronic mail from other computer systems using SMTP extensions to OpenVMS Mail.
- Access the Internet and other information services.
- Log into remote systems using TELNET, RLOGIN, or SSH.
- Execute commands on remote systems using RSHELL or SSH.
- Transfer files between local and remote systems with FTP, RCP, TFTP, SCP, and SFTP.
- Print files and manage print jobs on remote systems with the LPD and LPRM utilities.
- Talk to other users interactively with the TALK utility.
- Display information about other sites and users with the FINGER, RUSERS, and WHOIS utilities.

A.1.2. VSI TCP/IP for System Managers

With VSI TCP/IP, system managers can:

- Configure devices and services easily with command line-based configuration utilities.
- Provide IP connectivity for a variety of networking environments including DECnet-over-IP and Ethernet.
- Provide other networking connectivity over IP, including DECnet-over-IP.
- Provide access to NFS-mounted file systems with the VSI TCP/IP NFS software.
- Change the current configuration dynamically by modifying logical name definitions or by using the NETCONTROL utility.
- Provide security for logging into systems across the network with Kerberos and SSH software.
- Create and access name servers with DNS (Domain Name System) software.

- Configure dynamic routing with the GATED service, which supports routing protocols such as RIP, BGP, and others.
- Manage remote printing to print servers or to printers connected to the network with the LPD and stream client software.
- Provide remote access to local OpenVMS printers with the LPD server software.
- Provide electronic mail services with the SMTP and POP protocols; VSI TCP/IP provides SMTP enhancements for Message Router (MR), OpenVMS Mail.
- Access local and remote CD-ROMs, DATs, and conventional magnetic tape devices with the RMTALLOC utility.
- Synchronize system clocks from a central time server with NTP software and provide time updates to other hosts on the network.
- Provide binary compatibility with HPE TCP/IP Services for OpenVMS (formerly called UCX) to support Hewlett Packard Enterprise and third-party applications such as TeamLinks, DECmcc, and applications written to use DCE for OpenVMS.
- Diagnose system problems and messages with the CHECK, PING, TCPDUMP, TCPVIEW, TRACEROUTE, and X11DEBUG utilities.

A.1.3. VSI TCP/IP for Programmers

With VSI TCP/IP, programmers can:

- Program with socket library routines.
- Work with a $QIO interface.
- Program with RPC library routines.
- Access sample programs and user exits that can be used to provide additional security and to customize other services (such as printing).

A.2. TCP/IP Concepts

A.2.1. Physical Networks

Physical networks are the cables and associated wiring components that link computers to one another for network communications.

A.2.2. LAN (Local Area Network) Hardware Addresses

Network interface board manufacturers assign a unique hardware (physical) address to each interface board they produce.

A hardware address is usually composed of six numbers, one for each octet or eight-bit byte in the address value, separated by colons, such as 00:DD:A8:13:48:C5. The first three octets identify the manufacturer, while the remaining three octets are unique to the board.

A.2.3. IP Addresses

IP addresses identify hosts or interfaces on an IP network. An IP address consists of four numbers, one for each octet or eight-bit byte in the address value. IP addresses are written in dotted-decimal format, such as 191.87.34.22.

An IP address has two basic parts: a network number and a host number

Traditionally, the portions of the address that identify the network and host were determined by the class of the network:

Class A networks	Class A addresses are identified by a value from 1 to 127 in the first octet, such as in 26.1.1.1. In class A addresses, the first octet identifies the network, while the three remaining octets identify the host. For example, IP address 26.1.1.1 identifies host 1.1.1 on network 26.
Class B networks	Class B addresses are identified by a value from 128 to 191 in the first octet, such as in 161.1.1.1. In class B addresses, the first and second octets identify the network, while the remaining two octets identify the host. For example, IP address 161.1.1.1 identifies host 1.1 on network 161.1.
Class C networks	Class C addresses are identified by a value from 192 to 223 in the first octet, such as in 197.1.1.1. In class C addresses, the first three octets identify the network, while the remaining octet identifies the host. For example, IP address 197.1.1.1 identifies host 1 on network 197.1.1.

With the introduction of subnet masks, the division between the network and host portions of an IP address has become much more flexible. See *Subnet Masks* section for more information.

The network class determines the size of the network. A class A network can have 16,777,214 hosts, while a class B network can have 65,534 hosts, and a class C network can have only 254 hosts.

A.2.4. Subnet Masks

The original Internet addressing scheme made it possible for every host on a network to talk directly with every other host on the same network; other hosts were directly accessible if they used the same network number. In class A and class B networks, where very large numbers of hosts with the same network number are available, this scheme is no longer realistic because the underlying physical networks are constrained by bandwidth considerations. Ethernet networks cannot accommodate thousands or hundreds of thousands of hosts in a single, flat network space.

Subnet masks allow you to create multiple smaller networks from host addresses. For example, a class A network can be partitioned into class C subnetworks. These smaller, internal networks are called subnets. Subnet addresses are not exposed outside of the network; all changes to accommodate the additional addresses are handled internally. This simplifies routing information for the network and minimizes the amount of information the network must advertise externally.

Inside the network, you determine how to reallocate addresses by choosing how many bits of the host portion of each address are used as the subnet address and how many bits are used as the host address. You use subnet masks to divide the existing addresses into network and host portions. The subnet mask identifies how much of the existing address can be used as the network portion. The underlying physical network must also be divided into smaller, physical subnets when using a subnet mask to create subnets.

The following example illustrates how to create class C subnets from a class B network address:

The class B network address 161.44.0.0 can be divided by reallocating the first 24 bits of the 32-bit IP address to subnet addressing using the netmask 255.255.255.0. This reallocation allows you to use 161.44.1.0, 161.44.2.0, and so forth, up to 161.44.254.0 as network addresses. All traffic bound for any IP address beginning with the 16-bit network portion 161.44 will be routed to your site where internal routers handle subnetwork addresses. Valid addresses on the internal network, such as 161.44.4.42 and 161.44.224.12, can be reached from anywhere on the Internet; final delivery is handled by the routers on the individual physical subnets that contain the hosts associated with those addresses.

A.2.5. Broadcast Addresses

A system uses broadcast addresses to send information to all hosts on the network. Packets addressed to the network broadcast address are transmitted to every host with the same network number as the broadcast address. Broadcast packets are routinely used by the network to share routing information, field ARP requests, and send status and informational messages.

There are two common conventions used for broadcast addresses. The old convention, which older versions of SunOS and Berkeley UNIX BSD4.3 use, implements a broadcast address as the network portion of the address followed by all zeros. Using this convention, the broadcast address for the network 161.44 is 161.44.0.0. The new convention, which VSI TCP/IP and most other TCP/IP implementations use, implements a broadcast address as the network portion of the address followed by binary ones in all host portions of the address. In this scheme, the broadcast address for network 161.44 is 161.44.255.255.

If the network contains subnets, the broadcast address is relative to the local subnet. For example, host 128.44.12.1 with a subnet mask of 255.255.255.0 has an IP broadcast address of 128.44.12.255.

A.2.6. Host Names

Most sites assign host names to each system on the network because names are easier to remember than IP addresses. On a small, locally contained network, a host name may be only one word, such as WILLOW. However, on larger networks or on networks connected to the Internet, names are longer and denote a place in the organization and ultimately on the Internet. These longer, more detailed names are called fully qualified host names or fully qualified domain names (FQDNs). An example is LKG1.EXAMPLES.COM, where LKG1 is the individual host (or system) name, EXAMPLES identifies the organization to which it belongs, and COM indicates this organization is involved in commerce on the Internet.

A.2.7. TCP/IP Operation

The following steps present a highly simplified view of the events that occur during successful network communication.

1. Using the appropriate application, such as electronic mail, a user initiates communication to another system, identifying the remote system by name, such as LKG1.EXAMPLES.COM.

2. The application asks for the IP address of the system identified as LKG1.EXAMPLES.COM.

3. Using either DNS or host tables, the IP address of LKG1.EXAMPLES.COM is determined.

4. A connection is established using a three-way handshake.

5. Application information is organized into packets for transmission across the network.

6. The MTU (Maximum Transmission Unit) of the physical network is determined; if necessary, the packets are fragmented before being sent to the network interface card for delivery.

7. The hardware address of the next host (or hop) in the route to the target host is determined.

8. Each host along the route receives the packets and forwards them to the next hop in the route.

9. Once the packets arrive at the destination, they are reassembled in the appropriate order and delivered to the appropriate application. Some protocols acknowledge receipt of the packets to the sending host.

A.3. Basic TCP/IP Protocols

Networking protocols ensure reliable delivery of information from one host to another.

This section describes several of the more important TCP/IP protocols.

- IP (Internet Protocol)
- IPv6 (Internet Protocol V6)
- TCP (Transmission Control Protocol)
- UDP (User Datagram Protocol)
- SLIP (Serial Line Internet Protocol)
- PPP (Point-to-Point Protocol)

A.3.1. IP (Internet Protocol)

IP (Internet Protocol) is the networking protocol used to deliver data packets from one computer to another. The computers may reside on different networks as long as information can travel between them.

The IP layer in a TCP/IP stack is responsible for:

Routing data packets from one system to the next until they reach their destination	When a packet is received, the IP layer examines its routing and interface tables to see if the IP address of the destination host is one of its own IP addresses or a broadcast address. If the destination IP address is the same as the local IP address, IP passes the packet to the TCP or UDP layer. If the IP address does not belong to this host and is not a broadcast address, the IP layer determines the next hop in the route. If this host is configured as a router, it forwards the packet to the next hop. If this host is not configured as a router, it discards the packet.
Discovering the MTU	The MTU (Maximum Transmission Unit) is the size of the largest packet that can be sent along the physical network. The MTU depends on the type of physical network being used. For example, a typical MTU for Ethernet networks is 1500 bytes, while a typical MTU for FDDI is 4352 bytes.

	When the IP layer receives a packet to send, it determines which route it will use to forward the packet and obtains that route's MTU.
Fragmenting and reassembling packets	If a packet is larger than the MTU, the IP layer is responsible for breaking the packet into smaller pieces or fragments that travel along the network. A fragment can be further fragmented as required by the next segment of the network. All reassembly occurs at the destination, where the IP layer is responsible for putting all the fragments together in the right order before passing the packets on to the TCP or UDP layer.

A.3.2. IPv6 (Internet Protocol Version 6)

IPv6 is an advancement on IP (v4) which supports a larger address space and has various efficiency and security improvements. It has the same responsibilities that IPv4 does in the stack and can be used by the TCP or UDP layer.

A.3.3. TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol) provides a reliable mechanism for delivery of information to remote hosts.

On the sending host, the TCP layer of the TCP/IP stack is responsible for:

- Organizing the information sent by the application into segments (the amount of data that will fit into an IP datagram)
- Specifying the endpoints (ports) of the connection with the remote host
- Establishing a connection with the remote host
- Ensuring the remote host acknowledges packets that have been sent within a specified time

On the receiving host, the TCP layer of the TCP/IP stack is responsible for:

- Acknowledging received packets
- Organizing the packets into the correct sequence upon receipt from the sending host
- Forwarding the packets to the application using the specified port

TCP requires more overhead than UDP but provides reliable delivery of packets to the remote host.

A.3.4. UDP (User Datagram Protocol)

Applications can also use UDP (User Datagram Protocol) to deliver information to a remote host.

The UDP layer of the TCP/IP stack is responsible for:

- Organizing the information to be sent into a packet called a datagram
- Using a port to identify the program on the remote host to which the datagram is to be sent

- Verifying that the datagram contains the correct IP source and target addresses

UDP does not verify the successful delivery of packets to the target host. As a result, UDP requires less overhead than TCP. To accommodate this lack of verification, applications that use UDP often provide their own mechanisms for ensuring messages reach the target host in the correct sequence when required.

A.3.5. SLIP (Serial Line Internet Protocol)

SLIP (Serial Line Internet Protocol) allows the transmission of IP packets over serial lines. SLIP can be used over a direct connection between the serial ports of two systems, or over telephone lines with modems.

A.3.6. PPP (Point-to-Point Protocol)

Like SLIP, PPP (Point-to-Point Protocol) allows the transmission of IP packets over serial lines. PPP is a more versatile protocol than SLIP, and contains functionality that SLIP does not, such as:

- The ability to share the serial line with other protocols
- Error detection
- Support for both synchronous and asynchronous communication
- Dynamic configuration
- Negotiation of parameter values
- Support for different user-authentication protocols

While PPP is a more versatile serial-line protocol than SLIP, it is not available with all TCP/IP implementations.

A.4. Dynamic Configuration Protocols

To communicate with the rest of the network, a host must have an IP address. However, some systems do not have the hardware to permanently store an IP address. In addition, computers frequently share IP addresses when there are more computers than IP addresses, or when IP addresses are used only temporarily. For these situations, there are three dynamic configuration protocols: RARP, BOOTP, and DHCP.

A.4.1. RARP (Reverse Address Resolution Protocol)

RARP (Reverse Address Resolution Protocol) sends IP addresses to workstations that broadcast RARP requests containing their hardware addresses. RARP supplies IP addresses only and is commonly used by disk-less workstations to determine their Internet addresses.

A.4.2. BOOTP (Bootstrap Protocol)

BOOTP (Bootstrap Protocol) lets a host receive an IPv4 address and other configuration information from a BOOTP server on the network. BOOTP often specifies a bootstrap file for a client system to download, normally via TFTP (Trivial File Transfer Protocol). BOOTP lets systems that have no hard disk retrieve the information necessary to access their bootstrap file.

A.4.3. DHCP (Dynamic Host Configuration Protocol)

DHCP (Dynamic Host Configuration Protocol) builds upon the BOOTP protocol by letting a system receive all of the information necessary to function as a host on the network directly from a DHCP server. Unlike BOOTP, which only provides for permanent IPv4 addresses, DHCP supports three different mechanisms for allocating IPv4 addresses:

- Automatic: Hosts requesting an IPv4 address receive a permanent IPv4 address.
- Dynamic: Hosts requesting an IPv4 address receive a temporary IPv4 address.
- Manual: IPv4 addresses are manually configured and DHCP delivers the assigned addresses to requesting hosts.

A.5. Routing

Routing is the process of selecting the path that data packets take to reach their destination. Routing can be as simple as delivering packets to another host on the same network (*direct routing*), or it may involve forwarding packets to routers on the way to the destination network. This section explains the basics of IP routing.

IP routing determines how to forward packets to a destination host. When a packet is forwarded to a local host (that is, a host on the same network), routing is *direct*; if the packet must be forwarded through one or more routers to reach its destination, the route is *indirect*.

Routing information for indirect routes is stored in a table of IP and router address pairs. Information in the routing table can be specified in three ways:

Static routes	Static routes are used to specify routing information explicitly. They are usually easy to maintain, but they provide no mechanism to respond automatically to changing environments.
Default routes	Default routes are used when a host has no specific route for the destination host or network in its routing table. If data cannot be delivered directly (because the routing table has no entry for the destination host or network), the data is forwarded to the default router.
Dynamic routing	Dynamic routing can use a service such as GATED to exchange routing information between cooperating systems. The protocols used to exchange information are RIP (Routing Information Protocol), EGP (Exterior Gateway Protocol), HELLO (DCN Local Network Protocol), and BGP (Border Gateway Protocol).

The following sections describe routing tables and GATED in more detail.

A.5.1. The Routing Table

The routing table stores information about the routes that hosts can use to reach other hosts on the network or Internet. The routing table entries can be configured statically by the system manager, or dynamically by a program such as GATED.

- Static entries are established by manually entering information. Once a static routing table is established, you must update the table as changes occur.

- Dynamic entries are generated from information provided by a routing protocol (such as RIP) which collects information from other routers to populate the table. Dynamic routing solutions automatically share information and update the table as routing information changes.

  The routing table is designed to supply the next hop address (which is always local) for data bound for other networks. The routing table never contains information about routers beyond the local network segment, nor does it contain information about how to reach individual host addresses (although it can contain host-specific entries). Routers always forward data to networks until the destination network is the local network. When the data arrives at the destination network, it is forwarded directly to the appropriate host.

  Host-specific routes are special routing table entries that specify which router to use when data is bound for a specific remote host. Host-specific routes are frequently used to test new routers or to implement network security procedures.

A.5.2. Router Discovery

Router discovery is a method of finding a router when no default route entry exists in the routing table. When booting, a host using router discovery broadcasts a message asking for available routers. The available routers reply with a message indicating their address. The host adds the information to its routing table and sets the default route based on advertisements from routers on the local network automatically. Local routers must also support RDISC (Router Discovery protocol).

Under IPv6 Router Discovery is performed by the RTSOLD program after line initialization.

A.5.3. GATED

GATED can both learn and advertise known routes, allowing for automatic handling of network configuration changes and automatic selection of the best available route. Other routers on the local network must also support at least one of the protocols used by GATED (EGP, BGP, RIP, and HELLO). GATED only supports IPv4 routing.

A.6. DNS (Domain Name System) and Host Tables

DNS (Domain Name System) and host tables are two methods of mapping between host (computer) names and their IP addresses. When you specify a host by name, DNS or host tables are used to map the host name to its IP address. The host name can be local to your organization or anywhere in the world, if your site is connected to the Internet. DNS and host tables can also be used to map IP addresses to host names.

A.6.1. DNS (Domain Name System)

TCP/IP applications use DNS to convert host names to IP addresses, and vice versa. This conversion is called resolving.

A DNS resolver sends requests to another computer, called a DNS server, to resolve names into IP addresses. The DNS resolver can also send requests to the DNS server to resolve IP addresses to names.

DNS servers store host name and IP address information. If your computer needs information that is not on one DNS server, the server automatically requests the information from other servers.

Domains

In DNS terminology, a domain is a group of computers. The domain administrator determines which computers are in the domain. A domain name identifies a domain and consists of words separated by dots. An example of a domain name is FLOWERS.COM.

The parts of a domain name are created by the domain administrator or may be special words used on the Internet. Domain names can pertain to a site, an organization, or to types of organizations.

When read right to left, the first word in the domain name is the top-level domain which identifies the function of an organization or specifies a country name code. In the name FLOWERS.COM, .COM indicates an organization engaged in commerce. The top-level domain can also indicate a country, such as .FR for France, or .IT for Italy. The name of the organization is to the left of the top-level domain, such as FLOWERS. Any words to the left of the top-level domain are called subdomains. The left-most word in the domain name is the host name. For example, in OAK.FLOWERS.COM, OAK is a host in the FLOWERS.COM organization.

Domains and subdomains are organized in a hierarchical tree structure. Just as the root directory in VMS is expressed as an implicit 000000., the root directory in DNS is expressed as a dot (.). Domains are analogous to directories; subdomains are analogous to subdirectories within directories.

Top-level domains such as .ORG, .COM, and .EDU exist in the United States. Other countries group their domain names below their two-letter country code. Domains grouped under country codes include domains such as .CO for commercial and .AC for academic. In the United States, .US is occasionally used instead of another top-level domain name. Subdomains may provide additional geographic information, such as .PALO-ALTO.CA.US.

DNS Server

A DNS server is any computer running DNS software that lets it communicate with other DNS servers and store address information for later retrieval. DNS servers are also called name servers. Name servers cache (store) domain name information in memory for faster retrieval. Your network administrator provides the IP address of the name server on your network. Hosts implementing DNS come in five varieties:

Server	Description
Root name server	A root name server provides information about the start or base of the domain name tree. A root name server delegates authority to other primary name servers for the top-level domains such as .COM, .EDU, .US, .IT, etc. A root name server usually also handles those domains just below the root.
Primary name server	A primary name server has authority over one or more domains or subdomains. A primary name server reads information about the domain over which it has authority from the zone file, a special file that describes information about the domain and the hosts in that domain.
Secondary name server	A secondary name server for a domain receives information updates from the primary name server for that domain at regular intervals, and stores this information on disk. A secondary server is also authoritative for the domain.
Caching-only name server	A caching-only name server is not authoritative for any domain. If a caching-only name server cannot resolve a request, it forwards the request to an authoritative name server for that domain and caches the results for future use.

Resolver	A resolver sends requests for resolution to a DNS server. Any name server that can handle the request returns the response.

A.6.2. Host Tables

If DNS is not configured on your network, you can configure VSI TCP/IP host tables to resolve names and addresses. Like DNS, host tables also map between IP addresses and host names; unlike DNS, however, the information is stored locally on your computer and must be updated manually. Using host tables, you must ensure that every host name you specify while running TCP/IP applications is listed with its IP address. Whenever a change occurs on the network, such as when a new computer is added that you need to access, you must add the information to the host table. With the growth of the Internet, maintaining host tables for it has become practically impossible.

When you add or modify a host table entry, you specify the host name, the IP address, an optional description, and one or more optional, alternative names (aliases) for the host.

A.6.3. Using DNS and Host Tables Together

If you are using DNS, you may also want to use host tables. This is useful for temporary situations, such as when a new computer is added to the network, but has not yet been added to DNS.

The advantage of using DNS and host tables together for name resolution is that your system can access other systems even if the DNS server is not running or if the network is down. If you maintain entries in the host table for your local network, you can continue communicating with local systems until the DNS server or network is restored.

Warning

It is crucial to keep your host table entries synchronized with the DNS information.

A.7. ARP (Address Resolution Protocol)

Before hosts can communicate with each other, the sending host must discover the hardware address of the receiving host.

Hardware addresses are unique numbers (for example, 00:DD:A8:13:48:C5) assigned to network interface boards by their manufacturers or by network administrators.

ARP (Address Resolution Protocol) discovers the hardware address corresponding to a specific IP address and dynamically binds the hardware address to the IP address.

ARP is a low-level protocol that lets network administrators assign IP addresses to hosts on a network as they see fit. There is no need to match the addresses to those on the physical network because ARP handles this process dynamically.

An ARP mapping (also called a *translation*) provides the correct delivery address (that is, the hardware address) on the network for data destined for an IP address. ARP mappings are stored in a table in memory known as the *ARP cache*.

When data is to be delivered to a local IP address (an IP address on the same physical network), the TCP/IP stack broadcasts an ARP request to all hosts on the local network segment. The request message asks all hosts if the IP address belongs to them. If the IP address belongs to a host on the local network segment, that host adds its hardware address to the packet and returns it to the sender. All other hosts on the network discard the request. The ARP cache stores the address resolution information returned and makes it available each time network data is bound for that IP address.

Old mappings are deleted from the ARP cache automatically after a short period of time. Old mappings are also deleted automatically when they no longer work (that is, when new, correct mappings become available).

A.8. Neighbor Discovery

Neighbor Discovery is the IPv6 mechanism for mapping an IPv6 address to a hardware interface address. It works in similar ways to ARP does for IPv4, though it has improvements to reduce the impact on nodes other than the one that the address is being resolved for. Neighbor Discovery is also used as part of the Duplicate Address Detection portion of the autoconfiguration of interfaces.

A.9. SNMP (Simple Network Management Protocol)

SNMP (Simple Network Management Protocol) allows you to manage remote hosts on a network (for example, routers, hubs, and workstations). Both the network management host and the managed hosts (called agents) must follow the SNMP rules. Because SNMP is an open standard, you can mix and match network management hosts and agents from different vendors.

SNMP maintains information about your workstation in a management information base (MIB).

A.9.1. SNMP Traps

One of the main uses of SNMP is to make it easy to keep track of important events that occur on the managed network. To help automate network management, SNMP agents automatically send messages called traps to the network management host when certain events occur. For example, your workstation sends a trap when you reboot it.

One important type of SNMP trap is the *authentication failure trap*. Because SNMP network management hosts have access to sensitive configuration settings for the hosts on a managed network, it is important for network administrators to guard against breaches in network security that involve illegitimate use of SNMP messages.

For this reason, every SNMP message must be authenticated by network management hosts and SNMP agents using passwords called *communities*. If your agent receives an SNMP message that contains an incorrect community name for the type of operation requested, your agent sends a message to a network management host. This message contains information about the request your agent received:

- What the message requested
- Why your agent would not fulfill the request

A.9.2. SNMP Communities

An SNMP community is a type of password used by the SNMP network management host and SNMP agents to ensure that only known and trusted hosts can send SNMP messages to and receive SNMP messages from each other. Every SNMP message includes a community name, so every message can be validated. There are three types of community names:

Community name	Description
Read	The network management host must use the correct read community name when asking your SNMP agent to send it information about your host.
Write	The network management host must use the correct write community name when asking your SNMP agent to change some characteristic about your configuration.
Trap	If certain events happen in your workstation (for example, when you reboot your host, or when a network management host sends an SNMP message that contains the wrong read or write community name), your SNMP agent sends a trap message to a network management host. If your trap message is to be handled, the trap community name you send must match the name known to the target network management host.

A.10. Devices Supported by VSI TCP/IP

VSI TCP/IP supports a variety of network topologies. The supported network interfaces are:

- Ethernet controller
- DECnet-over-IP link
- Asynchronous PPP using any OpenVMS-supported terminal multiplexer
- SLIP (Serial Line IP) using any OpenVMS-supported terminal multiplexer
- Six-To-Four IPv6 over IPv4 tunneling interface

A.11. Protocols Supported by VSI TCP/IP

VSI TCP/IP is compatible with the current versions of the standard Internet networking protocol specifications as follows:

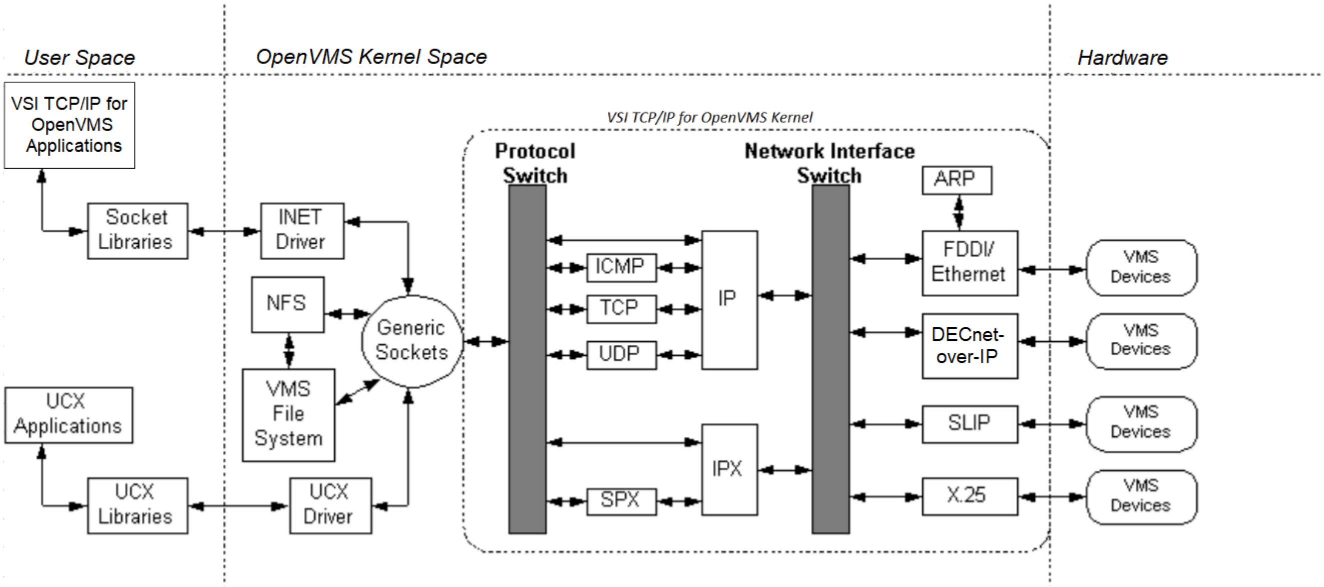
- BGP (Border Gateway Protocol): RFC-1105, RFC-1323
- BOOTP (Network Bootstrap Protocol): RFC-951, RFC-1534, RFC-1542, RFC-2132
- DHCP (Dynamic Host Configuration Protocol):RFC-2131, RFC-2132
- DNS (Domain Name Service): RFC-1034, RFC-1035, RFC-1101, RFC-1348
- EGP (Exterior Gateway Protocol): RFC-904
- Ethernet ARP (Address Resolution Protocol): RFC-826
- Ethernet RARP (Reverse Address Resolution Protocol): RFC-903
- FDDI (Fiber Distributed Data Interface): RFC-1188
- FINGER (User Status Protocol): RFC-1288

- FTP (File Transfer Protocol): RFC-959, RFC-1579, RFC-2389, RFC-2428
- ICMP (Internet Control Message Protocol): RFC-792, RFC-1256
- IP (Internet Protocol): RFC-791
- IP-over-X.25: RFC-87
- IPv6:RFC-2460, RFC-4294
- ICMP6. RFC 2463
- IPP (Internet Printing Protocol): RFC-2910, RFC 2911
- Kerberos: RFC-1411, RFC-1510
- NFS (Network File System): RFC-1094
- NTP (Network Time Protocol): RFC-1059
- OSPF (Open Shortest Path First): RFC-1583, RFC-1584
- Path MTU (Maximum Transmission Unit) Discovery: RFC-1191
- POP3 (Post Office Protocol Version 3): RFC-1725
- PPP (Point-to-Point Protocol): RFC-1332, RFC-1552, RFC-1661
- RIP (Routing Information Protocol): RFC-1058
- RPC (Remote Procedure Call Protocol): RFC-1057
- SLIP (Serial Line Internet Protocol): RFC-1055, RFC-1144
- SMTP (Simple Mail Transfer Protocol): RFC-821, RFC-822, RFC-974
- SNMP (Simple Network Management Protocol): RFC-1157, RFC-1213
- SNMP Agent eXtensibility (RFC-2741, RFC-2742)
- SYSTAT (Active Users Service Protocol): RFC-866
- TCP (Transmission Control Protocol): RFC-793
- TELNET (network virtual terminal protocol): RFC-854, RFC-855, RFC-856, RFC-857,
- RFC-2941
- RFC-858, RFC-859, RFC-860, RFC-885, RFC-1041, RFC-1073, RFC-1079, RFC-1091,
- RFC-1096, RFC-1205, RFC-1372, RFC-1411, RFC-1416
- TFTP (Trivial File Transfer Protocol): RFC-1350
- TN3270: RFC-1576
- UDP (User Datagram Protocol): RFC-768

- WHOIS (Directory Service Protocol): RFC-954
- XDR (eXternal Data Representation): RFC-1014

A.12. Understanding VSI TCP/IP Internals

This section describes how the VSI TCP/IP kernel interacts with the OpenVMS Operating System. To understand the information in this section, some background in OpenVMS internals is helpful. The following figure illustrates the VSI TCP/IP protocols and the overall organization of the VSI TCP/IP kernel.



VSI TCP/IP interacts directly with the OpenVMS Operating System. Generic sockets pass requests to the protocol switch, that differentiates between requests for IP use and for other support facilities. Requests are then sent through the network interface switch, which identifies the device for which the request is bound. When a request is received from a device, the steps occur in reverse.

A.12.1. The $QIO Interface

The programs implementing the lower layers of VSI TCP/IP—data link, network, and transport layers, with the exception of shared OpenVMS device drivers—reside in the VSI TCP/IP kernel. The VSI TCP/IP kernel is loaded into the OpenVMS S0 space (where the OpenVMS kernel is also loaded). Pages for S0 space are allocated when the OpenVMS system boots.

VSI TCP/IP kernel accommodates multiple $QIO interfaces. Each $QIO interface is implemented by a separate OpenVMS pseudo-device driver, allowing VSI TCP/IP to simultaneously support the $QIO interfaces of several popular networking implementations. This lets you use, without modification, applications designed for these other networking implementations. The default VSI TCP/IP $QIO interface, implemented in INETDRIVER.EXE, is used by the VSI TCP/IP shareable socket library and all VSI TCP/IP applications.

All VSI TCP/IP $QIO drivers communicate with the VSI TCP/IP kernel through a set of kernel transfer vectors that interface with the generic socket layer. The generic socket layer of the VSI TCP/IP kernel provides most of the facilities common to all network protocols (including reading and writing user-level data and synchronizing OpenVMS I/O request packets with network protocol events).

The generic socket layer uses the protocol switch for protocol-specific operations. The protocol-specific code (which may consist of several interconnected protocol modules) calls through the network interface switch to the appropriate network device driver to encapsulate and transmit packets.

The protocol-specific code can also receive timer interrupts through the protocol switch. Incoming packets are decapsulated by the network device drivers and passed to the protocol-specific code through the network interface switch.

A.12.2. Network Interface Device Drivers

The VSI TCP/IP kernel includes code allowing it to handle I/O for most network interface devices itself, rather than using OpenVMS device drivers. However, for devices that VSI TCP/IP shares with other software (for example, a Hewlett Packard Ethernet interface), the kernel uses standard OpenVMS device drivers and either VCI (VMS Communication Interface), FFI (Fast Function Interface), or ALTSTART driver interfaces.

A.12.3. Custom Applications

The include and library files (optionally installed with VSI TCP/IP) provide access to several programming interfaces for writing custom client and server applications. These interfaces include:

- A 4.3BSD-compatible shareable socket library
- A 4.4BSD-compatible shareable socket library
- An RPC (Remote Procedure Call) library based on Sun Microsystems' public domain RPC library
- The standard VSI TCP/IP $QIO interface
- A $QIO interface compatible with the HPE TCP/IP Services for OpenVMS

Consult the *VSI TCP/IP for OpenVMS Programmer's Reference* for additional information about the socket library and the VSI TCP/IP $QIO programming interface. The Sun RPC library routines are documented in the Sun Microsystems guide, *Networking on the Sun Workstation*.

VSI TCP/IP Correspondence with the OSI Reference Model	
Application, Presentation, Session, Transport	FTP, TELNET, FINGER, other applications
Network	TCP, UDP IP, ICMP Ethernet, ARP, X.25, IMP
Data Link	802.2 Coax, Fiber, Luminiferous Ether, and so on
Physical	

Note

The protocol suites implemented under VSI TCP/IP do not map one-to-one onto the OSI reference model. In particular, each TCP/IP application protocol generally handles the functions normally ascribed to the Application and Presentation layers of the OSI model.

Appendix B. Trademark and Copyright Notifications

This appendix contains a complete listing of trademarks and copyright notification contained in this manual.

The material in this document is for informational purposes only and is subject to change without notice. It should not be construed as a commitment by VMS Software, Inc. VMS Software, Inc. assumes no responsibility for any errors that may appear in this document.

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

The following third-party software may be included with your product and will be subject to the software license agreement.

Network Time Protocol (NTP). Copyright © 1992-2004 by David L. Mills. The University of Delaware makes no representations about the suitability of this software for any purpose.

Point-to-Point Protocol. Copyright © 1989 by Carnegie-Mellon University. All rights reserved. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated in all such forms and that any documentation, advertising materials, and other materials related to such distribution and use acknowledge that the software was developed by Carnegie Mellon University. The name of the University may not be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED ``AS IS'' AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTIBILITY AND FITNESS FOR A PARTICULAR PURPOSE.

RES_RANDOM.C. Copyright © 1997 by Niels Provos <provos@physnet.uni-hamburg.de> All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: This product includes software developed by Niels Provos.

4. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

Copyright © 1990 by John Robert LoVerso. All rights reserved. Redistribution and use in source and binary forms are permitted provided that the above copyright notice and this paragraph are duplicated

RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, TORT (INCLUDING NEGLIGENCE) OR STRICT LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions Copyright © 1980, 1982, 1985, 1986, 1988, 1989, 1990, 1993 by The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   This product includes software developed by the University of California, Berkeley and its contributors.

4. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

   THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions Copyright © 1993 by Hewlett-Packard Corporation.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies, and that the name of Hewlett-Packard Corporation not be used in advertising or publicity pertaining to distribution of the document or software without specific, written prior permission. THE SOFTWARE IS PROVIDED "AS IS" AND HEWLETT-PACKARD CORP. DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL HEWLETT-PACKARD CORPORATION BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions Copyright © 1995 by International Business Machines, Inc.

International Business Machines, Inc. (hereinafter called IBM) grants permission under its copyrights to use, copy, modify, and distribute this Software with or without fee, provided that the above copyright notice and all paragraphs of this notice appear in all copies, and that the name of IBM not be used in connection with the marketing of any product incorporating the Software or modifications thereof, without specific, written prior permission. To the extent it has a right to do so, IBM grants an immunity from suit under its patents, if any, for the use, sale or manufacture of products to the extent that such products are used for performing Domain Name System dynamic updates in TCP/IP networks by means of the Software. No immunity is granted for any product per se or for any other function of any product. THE SOFTWARE IS PROVIDED "AS IS", AND IBM DISCLAIMS ALL WARRANTIES, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL IBM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE, EVEN IF IBM IS APPRISED OF THE POSSIBILITY OF SUCH DAMAGES.

Portions Copyright © 1995, 1996, 1997, 1998, 1999, 2000 by Internet Software Consortium. All Rights Reserved. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies. THE SOFTWARE IS PROVIDED "AS IS" AND INTERNET SOFTWARE CONSORTIUM DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL INTERNET SOFTWARE CONSORTIUM BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright (c) 1996-2000 Internet Software Consortium.

Use is subject to license terms which appear in the file named ISC-LICENSE that should have accompanied this file when you received it. If a file named ISC-LICENSE did not accompany this file, or you are not sure the one you have is correct, you may obtain an applicable copy of the license at: http://www.isc.org/isc-license-1.0.html.

This file is part of the ISC DHCP distribution. The documentation associated with this file is listed in the file DOCUMENTATION, included in the top-level directory of this release. Support and other services are available for ISC products - see http://www.isc.org for more information.

ISC LICENSE, Version 1.0

1. This license covers any file containing a statement following its copyright message indicating that it is covered by this license. It also covers any text or binary file, executable, electronic or printed image that is derived from a file that is covered by this license, or is a modified version of a file covered by this license, whether such works exist now or in the future. Hereafter, such works will be referred to as "works covered by this license," or "covered works."

2. Each source file covered by this license contains a sequence of text starting with the copyright message and ending with "Support and other services are available for ISC products - see http://www.isc.org for more information." This will hereafter be referred to as the file's Bootstrap License.

3. If you take significant portions of any source file covered by this license and include those portions in some other file, then you must also copy the Bootstrap License into that other file, and

that file becomes a covered file. You may make a good-faith judgement as to where in this file the bootstrap license should appear.

4. The acronym "ISC", when used in this license or generally in the context of works covered by this license, is an abbreviation for the words "Internet Software Consortium."

5. A distribution, as referred to hereafter, is any file, collection of printed text, CD ROM, boxed set, or other collection, physical or electronic, which can be distributed as a single object and which contains one or more works covered by this license.

6. You may make distributions containing covered files and provide copies of such distributions to whomever you choose, with or without charge, as long as you obey the other terms of this license. Except as stated in (9), you may include as many or as few covered files as you choose in such distributions.

7. When making copies of covered works to distribute to others, you must not remove or alter the Bootstrap License. You may not place your own copyright message, license, or similar statements in the file prior to the original copyright message or anywhere within the Bootstrap License. Object files and executable files are exempt from the restrictions specified in this clause.

8. If the version of a covered source file as you received it, when compiled, would normally produce executable code that would print a copyright message followed by a message referring to an ISC web page or other ISC documentation, you may not modify the file in such a way that, when compiled, it no longer produces executable code to print such a message.

9. Any source file covered by this license will specify within the Bootstrap License the name of the ISC distribution from which it came, as well as a list of associated documentation files. The associated documentation for a binary file is the same as the associated documentation for the source file or files from which it was derived. Associated documentation files contain human-readable documentation which the ISC intends to accompany any distribution.

   If you produce a distribution, then for every covered file in that distribution, you must include all of the associated documentation files for that file. You need only include one copy of each such documentation file in such distributions.

   Absence of required documentation files from a distribution you receive or absence of the list of documentation files from a source file covered by this license does not excuse you from this requirement. If the distribution you receive does not contain these files, you must obtain them from the ISC and include them in any redistribution of any work covered by this license. For information on how to obtain required documentation not included with your distribution, see: http://www.isc.org/getting-documentation.html.

   If the list of documentation files was removed from your copy of a covered work, you must obtain such a list from the ISC. The web page at http://www.isc.org/getting-documentation.html contains pointers to lists of files for each ISC distribution covered by this license.

   It is permissible in a source or binary distribution containing covered works to include reformatted versions of the documentation files. It is also permissible to add to or modify the documentation files, as long as the formatting is similar in legibility, readability, font, and font size to other documentation in the derived product, as long as any sections labeled CONTRIBUTIONS in these files are unchanged except with respect to formatting, as long as the order in which the CONTRIBUTIONS section appears in these files is not changed, and as long as the manual page which describes how to contribute to the Internet Software Consortium (hereafter referred to as the Contributions Manual Page) is unchanged except with respect to formatting.

Documentation that has been translated into another natural language may be included in place of or in addition to the required documentation, so long as the CONTRIBUTIONS section and the Contributions Manual Page are either left in their original language or translated into the new language with such care and diligence as is required to preserve the original meaning.

10. You must include this license with any distribution that you make, in such a way that it is clearly associated with such covered works as are present in that distribution. In any electronic distribution, the license must be in a file called "ISC-LICENSE".

    If you make a distribution that contains works from more than one ISC distribution, you may either include a copy of the ISC-LICENSE file that accompanied each such ISC distribution in such a way that works covered by each license are all clearly grouped with that license, or you may include the single copy of the ISC-LICENSE that has the highest version number of all the ISC-LICENSE files included with such distributions, in which case all covered works will be covered by that single license file. The version number of a license appears at the top of the file containing the text of that license, or if in printed form, at the top of the first page of that license.

11. If the list of associated documentation is in a separated file, you must include that file with any distribution you make, in such a way that the relationship between that file and the files that refer to it is clear. It is not permissible to merge such files in the event that you make a distribution including files from more than one ISC distribution, unless all the Bootstrap Licenses refer to files for their lists of associated documentation, and those references all list the same filename.

12. If a distribution that includes covered works includes a mechanism for automatically installing covered works, following that installation process must not cause the person following that process to violate this license, knowingly or unknowingly. In the event that the producer of a distribution containing covered files accidentally or wilfully violates this clause, persons other than the producer of such a distribution shall not be held liable for such violations, but are not otherwise excused from any requirement of this license.

13. COVERED WORKS ARE PROVIDED "AS IS". ISC DISCLAIMS ALL WARRANTIES WITH REGARD TO COVERED WORKS INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

14. IN NO EVENT SHALL ISC BE LIABLE FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OF COVERED WORKS.

Use of covered works under different terms is prohibited unless you have first obtained a license from ISC granting use pursuant to different terms. Such terms may be negotiated by contacting ISC as follows:

Internet Software Consortium

950 Charter Street

Redwood City, CA 94063

Tel: 1-888-868-1001 (toll free in U.S.)

Tel: 1-650-779-7091

Fax: 1-650-779-7055

Email: info@isc.org

Email: licensing@isc.org

DNSSAFE LICENSE TERMS

This BIND software includes the DNSsafe software from RSA Data Security, Inc., which is copyrighted software that can only be distributed under the terms of this license agreement.

The DNSsafe software cannot be used or distributed separately from the BIND software. You only have the right to use it or distribute it as a bundled, integrated product.

The DNSsafe software can ONLY be used to provide authentication for resource records in the Domain Name System, as specified in RFC 2065 and successors. You cannot modify the BIND software to use the DNSsafe software for other purposes, or to make its cryptographic functions available to end-users for other uses.

If you modify the DNSsafe software itself, you cannot modify its documented API, and you must grant RSA Data Security the right to use, modify, and distribute your modifications, including the right to use any patents or other intellectual property that your modifications depend upon.

You must not remove, alter, or destroy any of RSA's copyright notices or license information. When distributing the software to the Federal Government, it must be licensed to them as "commercial computer software" protected under 48 CFR 12.212 of the FAR, or 48 CFR 227.7202.1 of the DFARS.

You must not violate United States export control laws by distributing the DNSsafe software or information about it, when such distribution is prohibited by law.

THE DNSSAFE SOFTWARE IS PROVIDED "AS IS" WITHOUT ANY WARRANTY WHATSOEVER. RSA HAS NO OBLIGATION TO SUPPORT, CORRECT, UPDATE OR MAINTAIN THE RSA SOFTWARE. RSA DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO ANY MATTER WHATSOEVER, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

If you desire to use DNSsafe in ways that these terms do not permit, please contact:

RSA Data Security, Inc.

100 Marine Parkway

Redwood City, California 94065, USA