IBM

Program Directory for

TCP/IP for z/VM®

Level 430

Program Number 5739-A03

for Use with
z/VM Version 4 Release 3.0

Document Date: March 2003

GI10-4741-00

> **Note!**
>
> Before using this information and the product it supports, be sure to read the general information under "Notices" on page 119.

Contents

Figures

1.0 Introduction

This program directory is intended for the system programmer responsible for program installation and maintenance. It contains information that corresponds to the material and procedures for installation and service of the following:

- TCP/IP for z/VM

Note: It is recommended that you review this program directory in its entirety before you install or service this program, then keep this document for future reference.

The program directory contains the following sections:

- 2.0, "Program Materials" on page 4 identifies basic (and optional) TCP/IP for z/VM program materials and documentation

- 3.0, "Program Support" on page 7 describes the IBM support available for TCP/IP for z/VM

- 4.0, "Program and Service Level Information" on page 10 lists APARs (program level fixes) that have been incorporated within TCP/IP for z/VM

- 5.0, "Installation Requirements and Considerations" on page 12 identifies resources and considerations for installing and using TCP/IP for z/VM

- 6.0, "Installation Instructions" on page 31 provides detailed installation instructions for TCP/IP for z/VM

- 7.0, "Service Instructions" on page 43 provides detailed servicing instructions for TCP/IP for z/VM

- Appendix A, "TCP/IP Installation and Service Utilities" on page 73 provides information about the TCP2PROD command (supplied for placing TCP/IP for z/VM files into production), its related CATALOG files, and other TCP/IP service aids

- Appendix B, "Modifying TCP/IP for z/VM CATALOG Files" on page 85 provides information about how to use VMSES/E local modifications to alter TCP/IP for z/VM CATALOG files

- Appendix D, "Modifying the TCP/IP for z/VM Default Installation" on page 92 presents considerations for changes that affect the installation (and service of) TCP/IP for z/VM

- Appendix C, "PTF Commit Procedure for TCP/IP for z/VM System Files" on page 89 provides information for committing PTF service that is associated with TCP/IP for z/VM (binary) system files

- Appendix E, "Making Local Modifications to TCP/IP for z/VM Modules" on page 94 provides information to help you implement local modifications to TCP/IP for z/VM modules

- Appendix F, "Modifying TCP/IP for z/VM VMNFS Code" on page 104 provides information specific to TCP/IP NFS server module local modifications

- Appendix G, "TCP/IP for z/VM Build Lists" on page 106 provides information about the VMSES/E build lists used to maintain TCP/IP for z/VM

- Appendix H, "Moving TCP/IP for z/VM to SFS Directories" on page 108 provides instructions for changing the TCP/IP for z/VM service environment to use Shared File System directories instead of default minidisks

1

- Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 provides considerations and optional instructions for copying client files to the system Product Code minidisk

- Appendix J, "z/VM Express Service Instructions" on page 116 provides alternative instructions for installing RSU and COR service using z/VM Express Service support

---

Obtaining Updated Planning Information

Before you install TCP/IP for z/VM, read 3.1, "Preventive Service Planning" on page 7. This section describes how to obtain any updates to the information and procedures presented within this program directory.

1.1 Program Description

TCP/IP (Transmission Control Protocol/Internet Protocol) enables z/VM customers to participate in a multivendor, open networking environment using the TCP/IP protocol suite for communications and interoperability. The applications included in TCP/IP provide the ability to transfer files, send mail, log on a remote host, allow access from any other TCP/IP node in the network, and perform other network client and server functions.

Transmission Control Protocol/Internet Protocol for z/VM, Level 430, (TCP/IP for z/VM) contains the functions provided by TCP/IP for z/VM, Level 420, and provides the following enhancements:

- TCP/IP (*stack*) protocol server performance improvements which optimize high-use paths, improve algorithms, and implement other performance-related features. With these enhancements, existing server behavior is maintained while achieving higher performance, especially in environments where the TCP/IP stack acts as a router on behalf of other guests.

- Updated support for Queued Direct I/O (QDIO) and HiperSockets devices, which:

  - detects and enables Multicast capability for HiperSockets devices

  - detects and configures Broadcast capability for QDIO devices

  - handles QDIO VM Guest LAN "LAN Not Ready" responses and asynchronous Start LAN (STRTLAN) conditions.

- Addition of Equal-Cost Multipath support for the TCP/IP (*stack*) and MPROUTE servers. With this support, up to four equal-cost paths to a particular destination can be maintained within the TCP/IP routing table, to facilitate load balancing and to improve performance.

- Expanded TCP/IP (*stack*) server reliability improvements to facilitate the prevention of several Denial of Service (DoS) attacks. Newly addressed attacks are:

  - IGMP-based attacks which attempt to absorb TCP/IP Large Envelope resources (KOD, KOX)

  - The sending of TCP packets which lack established header flag settings (STREAM, R4P3D)

  - The sending of packets which contain identical machine source and destination information (BLAT)

– Resource depletion through the withholding of connection state change acknowledgements (SYNFLOOD).

- The **NETSTAT** command includes support for a new **OBEY** option which can be used to make dynamic changes to the TCP/IP stack server configuration. Any data that is appropriate for use with the OBEYFILE command can be processed using a NETSTAT OBEY command (to the extent that data can be provided via the CMS command line).

- A new **IFCONFIG** command, which facilitates dynamic configuration changes to network interfaces for the z/VM TCP/IP stack in a manner similar to the Linux and UNIX `ifconfig` commands.

  The IFCONFIG command also provide much of the underlying support for the z/VM TCP/IP configuration "wizard" (IPWIZARD) which provides a menu-driven interface for defining and implementing a basic network configuration.

- Miscellaneous documentation improvements

- Incorporation of miscellaneous service APARs for previous IBM TCP/IP for VM releases.

2.0 Program Materials

An IBM program is identified by a program number. The program number for TCP/IP for z/VM is 5739-A03.

The z/VM Version 4 Release 3.0 program announcement material provides detailed information about features supported by TCP/IP for z/VM. If you have not already received a copy of this information, contact your IBM marketing representative.

The following sections identify:

- basic and optional program materials that are applicable to this program

- publications useful during installation and service.

2.1 Basic Machine-Readable Material

TCP/IP for z/VM is distributed as part of the z/VM Version 4 Release 3.0 System DDR. Refer to the *z/VM V4.3 Expands Industry Leading Virtualization Technology* Software Announcement, 202-104, for information about ordering z/VM Version 4 Release 3.0 and its features.

2.2 Optional Machine-Readable Material

There are no features or optional machine-readable materials associated with TCP/IP for z/VM.

2.3 Program Publications

The following sections identify the basic publications associated with TCP/IP for z/VM. There are no optional publications for this component of z/VM.

2.3.1 Basic Program Publications

Publications associated with TCP/IP for z/VM are listed in Figure 1:

Figure 1 (Page 1 of 2). Basic Material: Unlicensed Publications

Publication Title	Form Number
z/VM: TCP/IP Level 430 Planning and Customization	SC24-6019
z/VM: TCP/IP Level 430 User's Guide	SC24-6020
z/VM: TCP/IP Level 430 Programmer's Reference	SC24-6021
z/VM: TCP/IP Level 430 Messages and Codes	GC24-6022

Figure 1 (Page 2 of 2). Basic Material: Unlicensed Publications

Publication Title	Form Number
z/VM: TCP/IP Level 430 Diagnosis Guide	GC24-6023

One copy of the following program publication is provided at no charge to licencees of TCP/IP for z/VM:

- z/VM: TCP/IP Level 430 Planning and Customization (SC24-6019)

2.3.2 Softcopy Publications

TCP/IP for z/VM softcopy publications are provided as part of the *VM Collection* libraries on CD-ROM (SK2T-2067). In addition, softcopy publications are available at the TCP/IP for z/VM home page on the World Wide Web; the URL for this home page is:

```
http://www.vm.ibm.com/related/tcpip/
```

2.4 Microfiche Support

There is no microfiche for TCP/IP for z/VM.

2.5 Publications Useful During Installation and Service

The publications listed in Figure 2 may be useful during the installation and servicing of TCP/IP for z/VM. To order copies, contact your IBM representative.

Figure 2 (Page 1 of 2). Publications Useful During Installation / Service on z/VM Version 4 Release 3.0

Publication Title	Form Number
z/VM: VMSES/E Introduction and Reference	GC24-5994
z/VM: Service Guide	GC24-5993
z/VM: CMS Planning and Administration	SC24-6042
z/VM: CP Planning and Administration	SC24-6043
z/VM: CMS Command and Utility Reference	SC24-6010
z/VM: CMS File Pool Planning, Administration, and Operation	SC24-5949
z/VM: System Messages and Codes - CMS	GC24-6031
z/VM: System Messages and Codes - Other Components	GC24-6032
z/VM: REXX/VM Reference	SC24-6035
z/VM: Language Environment 1.8 C Run-Time Library Reference	SC24-6038
z/VM: CMS Callable Services Reference	SC24-6004
z/VM: TCP/IP Level 430 Planning and Customization	SC24-6019

Figure 2 (Page 2 of 2). Publications Useful During Installation / Service on z/VM Version 4 Release 3.0

Publication Title	Form Number
z/VM: TCP/IP Level 430 User's Guide	SC24-6020

3.0 Program Support

This section describes the IBM support available for TCP/IP for z/VM.

3.1 Preventive Service Planning

Before you install TCP/IP for z/VM, check with your IBM Support Center or use IBMLink™ to determine if Preventive Service Planning (PSP) information is available that you should know. To obtain this information, specify the appropriate UPGRADE and SUBSET values listed in Figure 3:

Figure 3. PSP Upgrade and Subset ID

| **RETAIN™** | | | | |
COMPID	**Release**	**Upgrade**	**Subset**	**Component Name**
5735FAL00	430	TCPIP430	VM430	TCP/IP for z/VM
5735FAL00	430	TCPIP430	*yynn*RSU	RSU Service Recommendations

RSU-BY-LVL information can be obtained from the VM service RSU web site at this URL:
http://www.ibm.com/eserver/zseries/zvm/service/rsu

3.2 Statement of Support Procedures

With TCP/IP for z/VM, you are entitled to support under the basic warranty for z/VM Version 4 Release 3.0. Also, note that z/VM Software Subscription and Support is *automatically* added when you order z/VM — this provides zSeries service to which you are likely accustomed.

Note: You must take specific action when you order z/VM to decline z/VM Software Subscription and Support.

Report any difficulties you have using this program to your IBM Support Center. If an APAR (Authorized Program Analysis Report) is required, the Support Center will provide the address to which any needed documentation can be sent.

Figure 4 identifies IBM RETAIN information — the Component ID (COMPID), Release, and Field Engineering Service Number (FESN) — that corresponds to TCP/IP for z/VM:

Figure 4. Component IDs

| **RETAIN** | | | |
COMPID	**Release**	**Component Name**	**FESN**
5735FAL00	430	TCP/IP for z/VM	0461035

3.3 Service Information

The IBM Software Support Center provides telephone assistance for problem diagnosis and resolution. You can call the IBM Software Support Center at any time; you will receive a return call within eight business hours (Monday—Friday, 8:00 a.m.—5:00 p.m., local customer time). The number to call is:

1-800-237-5511

Outside of the United States or Puerto Rico, contact your local IBM representative or your authorized supplier.

Various installation and service-related items, such as the Preventive Service Planning (PSP) "bucket" and current RSU status/content information, are available at the TCP/IP for z/VM home page on the World Wide Web. The URL for this home page is:

`http://www.vm.ibm.com/related/tcpip/`

3.3.1 Problem Documentation

When working with TCP/IP for z/VM support personnel on problems associated with an active Problem Management Record (PMR), diagnostic information may occasionally be requested. In such cases, the support staff will work with you to determine how to best provide any requested documentation. For reference, addresses that can be used to submit various documentation formats are listed in this section.

Note: The addresses that follow should not be used as a problem reporting facility. All requests for problem assistance must be processed through the IBM Software Support Center, as previously described. Documentation submitted to any of these addresses will be reviewed only if it is associated with an active PMR.

Figure 5. Problem Documentation Addresses

Format	Address
Internet	vmtcpdoc@vnet.ibm.com
IBMLink	GDLVM7(TCPLVL2)
Carrier Service	IBM Corporation
	Attention: *IBM contact name*
	Dept. G79G
	1701 North St.
	Endicott, NY 13760

3.3.2 Communicating Your Comments to IBM

If you have comments about or suggestions for improving the TCP/IP for z/VM program product, or this Program Directory, please provide them to IBM through one of the following means:

- If you prefer to send comments by mail, use the address provided with the Reader's Comments form (RCF) at the back of this document

- If you prefer to send comments electronically, use this Internet e-mail address:

  `vmtcpip@vnet.ibm.com`

If you send documentation-related comments, please include:

- the title of this publication
- the section title, section number, or topic to which your comment applies.

4.0 Program and Service Level Information

This section identifies the program level and any relevant service levels of TCP/IP for z/VM. In this context, *program level* refers to APAR fixes incorporated within the TCP/IP for z/VM program; *service level* refers to PTFs that are supplied with this product. Information about the TCP/IP for z/VM cumulative service tape is provided as well.

4.1 Program Level Information - TCP/IP for z/VM

APAR fixes (for previous levels of IBM TCP/IP for VM) that have been incorporated into this level of TCP/IP for z/VM are:

```
PQ51738  PQ52180  PQ52577  PQ52706  PQ53028  PQ53361  PQ53382  PQ53384
PQ53385  PQ53386  PQ53392  PQ53393  PQ53394  PQ53395  PQ53969  PQ54062
PQ54643  PQ54649  PQ54661  PQ54709  PQ54859  PQ55099  PQ55469  PQ55587
PQ56226  PQ56305  PQ56359  PQ57524  PQ58102  PQ58360
```

4.2 Service Level Information

Before you install and configure TCP/IP for z/VM, you should review the TCPIP430 PSP (Preventive Service Planning) "bucket" for updated installation information that you should be aware of, or for information about PTFs that should be installed. Specify upgrade and subset values of **TCPIP430** and **VM430**, respectively, when you request or obtain this information.

4.3 Cumulative Service (RSU) Tape Information

Cumulative service for TCP/IP for z/VM is available through a periodic, preventive service tape, the Recommended Service Upgrade (RSU). The RSU tape is used to provide service updates for multiple z/VM components (including TCP/IP for z/VM) and is often referred to as a *stacked* RSU tape.

The current-level of the stacked z/VM RSU can be obtained using the information provided in Figure 6:

Figure 6. Cumulative Service (RSU) Tape Information

	RETAIN	
COMPID	Release	PTF
568411202	RSU	**UM97430**

Note: Current RSU status and content information is available at the TCP/IP for z/VM home page on the World Wide Web. The URL for this home page is:

```
http://www.vm.ibm.com/related/tcpip/
```

5.0 Installation Requirements and Considerations

The following sections identify system requirements for installing TCP/IP for z/VM.

5.1 Hardware Requirements

There are no special hardware requirements to install TCP/IP for z/VM. Additional hardware requirements for exploiting specific functions of TCP/IP for z/VM are documented in the announcement material and in *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019).

5.2 Program Considerations

The following sections list programming considerations for installing TCP/IP for z/VM.

5.2.1 Operating System Requirements

TCP/IP for z/VM requires the following operating system:

- z/VM Version 4 Release 3.0
- CMS Level 19

5.2.2 Other Program Product Requirements

5.2.2.1 Other Program Product Requirements - TCP/IP for z/VM

IBM C for VM/ESA Version 3 Release 1 (5654-033) has been used to build the C components which provide the TCP/IP for z/VM services that follow. To use these services, the IBM Language Environment® run-time library must be available:

- Domain Name Server virtual machine (NAMESRV)
- Internet Message Access Protocol server (IMAP)
- Multiple Protocol ROUTE server (MPROUTE)
- Network Data Base servers (NDBPMGR and NDBSRV*nn*)
- Portmapper server (PORTMAP)
- Remote Execution daemon (REXECD and RXAGENT*n*)
- RouteD server (ROUTED)
- SNMP Query Engine and Agent (SNMPD and SNMPQE)
- Sockets Applications Programming Interface
- Network File System server (VMNFS)
- Kerberos Authentication and Administrator Servers (VMKERB and ADMSERV)

Various client functions require Language Environment run-time library support as well. Representative of these are:

- CMSRESOL and CMSRESXA
- DIG
- NFS (client)
- NSLOOKUP
- RPCGEN and RPCINFO
- NDBSRVS

The Language Environment run-time library necessary to use these C components is supplied with z/VM Version 4 Release 3.0.

Note: By default, the code for the above services is installed when you install TCP/IP for z/VM, regardless of whether you intend to use all, or only a subset, of these services.

Additional software requirements for exploiting specific TCP/IP for z/VM functions are documented in the announcement material and in *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019).

5.2.2.2 Other Program Product Requirements - SSL Server

To use the Secure Socket Layer (SSL) server, a suitably configured Linux kernel and file system must be installed on your z/VM system. Information about Linux requirements and preparing Linux for use with the SSL server is available at the TCP/IP for z/VM home page on the World Wide Web. The URL for this home page is:

`http://www.vm.ibm.com/related/tcpip/`

5.2.3 Program Installation/Service Considerations

This section describes items that should be considered before you install or service TCP/IP for z/VM

- VMSES/E is required to install and service this product.

- If multiple users install and maintain licensed products on your system, you may encounter problems when you attempt to gain write access to the Software Inventory (MAINT 51D) minidisk. Write access contention to this minidisk can be eliminated by converting the Software Inventory to use CMS Shared File System (SFS) directories instead of minidisks. See the *z/VM: VMSES/E Introduction and Reference*, Chapter 18, "Changing the Software Inventory to an SFS Directory," for information about how to make this change.

- The **4TCPIP30** user ID is the IBM-supplied user ID for installing and servicing TCP/IP for z/VM. If you choose to use a different user ID or you elect to use different minidisks and/or SFS directories for TCP/IP for z/VM maintenance purposes, suitable PPF override changes must be implemented to reflect your changes.

  Note: **If z/VM Express Service support will be used for applying service for your system, the TCP/IP for z/VM installation user ID (4TCPIP30) must not be changed**.

- TCP/IP for z/VM is supplied with two IBM-defined components. When you perform installation and service tasks, you need to make use of the appropriate TCP/IP component, based on whether the installation and service environment is maintained using minidisks (the default) or using Shared File System (SFS) directories. Select and use the appropriate component for your environment from those listed here: environment.

  The IBM-defined components for TCP/IP for z/VM are:

  TCPIP   Used if TCP/IP for z/VM is installed and serviced using minidisks

  TCPIPSFS   Used if TCP/IP for z/VM is installed and serviced using the Shared File System.

  Note! Minidisk Requirement

  Certain minidisks **must** be defined and used with TCP/IP server machines, even when TCP/IP for z/VM service minidisks are converted to Shared File System directories. This requirement is explained further in item 5 of 5.3, "DASD Storage and User ID Requirements" on page 22.

- On occasion, you may need to perform additional installation or service processing steps to account for problems or errors that cannot be corrected through conventional means. If such steps are required, appropriate explanatory notes and text will be provided in updated levels of this document.

- TCP/IP for z/VM source files are supplied in packed format. Use the CMS COPYFILE command (with its UNPACK option) to unpack TCP/IP source files prior to their use.

- To use the Secure Socket Layer (SSL) server, a suitably configured Linux kernel and file system must be installed on your z/VM system. Information about Linux requirements and preparing Linux for use with the SSL server is available at the TCP/IP for z/VM home page on the World Wide Web. The URL for this home page is:

  `http://www.vm.ibm.com/related/tcpip/`

5.2.4 Migration Considerations

This section provides general information about changes to TCP/IP for z/VM that you should be aware of prior to its installation and use. The changes described here have been implemented to accommodate the introduction of new functions and improvements to existing functions. In some cases, existing functions may have been removed or replaced by alternative functions.

Note: TCP/IP Level 430 is supported on corresponding releases of CP and CMS <u>only</u>.

5.2.4.1 Packaging

5.2.4.1.1 General Information About TCP/IP Level 430

- TCP/IP Level 430 is included as a pre-installed component of the z/VM product; its use is governed by your license for z/VM.

- TCP/IP Level 430 is **not** separately orderable from the z/VM product, although it is serviced separately from z/VM.

- This level of TCP/IP relies on the presence of certain functions in CP and CMS. Abends and incorrect results are possible if you attempt to use it with an older level of CP or CMS.

- TCP/IP softcopy publications are provided in the same manner as other z/VM softcopy publications, and are included with these z/VM publications.

- The **Pascal**-based client and server functions supplied with TCP/IP Level 430 (for example, FTP MODULE or SMTP MODULE) are **not backward compatible** with prior-level TCP/IP "stack" servers, due to internal control block changes that have been implemented in this new level of TCP/IP. Conversely, it should be possible to use older Pascal-based functions in concert with a TCP/IP Level 430 "stack" module in most environments.

  For a summary of TCP/IP Level 430 Pascal component compatibility, see Figure 7.

Figure 7. TCP/IP Level 430 Pascal Module Compatibility		
	TCP/IP Level 430 Pascal Functions	**Prior-level TCP/IP Pascal Functions**
TCP/IP Level 430 Stack Server	Compatible	Compatible
Prior-level TCP/IP Stack Server	**Not Compatible**	Compatible

5.2.4.1.2 Changes Introduced in TCP/IP Level 420

- TCP/IP Level 420 is included as a pre-installed component of the z/VM product; its use is governed by your license for z/VM.

- TCP/IP Level 420 is **not** separately orderable from the z/VM product, although it is serviced separately from z/VM.

- This level of TCP/IP relies on the presence of certain functions in CP and CMS. Abends and incorrect results are possible if you attempt to use it with an older level of CP or CMS.

- TCP/IP softcopy publications are provided in the same manner as other z/VM softcopy publications, and are included with these z/VM publications.

5.2.4.1.3 Changes Introduced in TCP/IP Level 410

- TCP/IP for z/VM is no longer a priced feature of z/VM Version 4 Release 3.0. No action is required to activate (enable) this component of z/VM.

- The TCP/IP NFS Server Feature for z/VM is no longer a priced feature of TCP/IP for z/VM. No action is required to activate (enable) the TCP/IP NFS server.

- The TCP/IP Source Feature for z/VM is no longer a priced feature of TCP/IP for z/VM. TCP/IP for z/VM source files are now supplied and installed as part of the z/VM Version 4 Release 3.0 System DDR.

5.2.4.1.4 Changes Introduced in TCP/IP Level 3A0

- DES Kerberos functions are now incorporated in the base TCP/IP Feature for z/VM (non-DES Kerberos functions are no longer available). Therefore, customers who have a Kerberos database that was created in a non-DES environment **must** delete and then rebuild that database using the supplied DES Kerberos functions. Refer to the chapter titled "Configuring the Kerberos Server" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information about building the Kerberos database.

- The various servers and code that provide support for the Network Computing System (NCS) are no longer provided.

- The installation user ID-owned "Sample" (2C2) minidisk and its corresponding sample files are no longer provided. The sample files provided on this disk in prior levels of TCP/IP can be obtained via the IBM TCP/IP for VM Internet home page, for which the URL is:

    ```
    http://www.vm.ibm.com/related/tcpip/
    ```

5.2.4.1.5 Changes Introduced in TCP/IP Function Level 310

- TCP/IP-based user e-mail functions are included in CMS. TCP/IP-specific versions of NOTE and SENDFILE are **not** provided with the TCP/IP Feature for z/VM.

- A subset of RSCS Version 3 Release 2 function is available with the z/VM product to provide enhanced TCP/IP client and server print capabilities. RSCS functions not related to TCP/IP printing enhancements require a separate RSCS license for use. Refer to the *RSCS Version 3 Release 2 Program Directory* for additional information.

- Non-DES Kerberos functions are included in the base TCP/IP Feature for z/VM. DES Kerberos functions are available as a separate feature that must be separately installed.

5.2.4.2 General TCP/IP Usage

5.2.4.2.1 Changes Introduced in TCP/IP Level 430

- The **Pascal**-based client and server functions supplied with TCP/IP Level 430 (for example, FTP MODULE or SMTP MODULE) are **not backward compatible** with prior-level TCP/IP "stack" servers, due to internal control block changes that have been implemented in this new level of TCP/IP. Conversely, it should be possible to use older Pascal-based functions in concert with a TCP/IP Level 430 "stack" module in most environments.

    For a summary of TCP/IP Level 430 Pascal component compatibility, see Figure 7 on page 15.

5.2.4.2.2 Changes Introduced in TCP/IP Level 420

- TCPIP has been changed to recognize a TCP/IP loopback address of *only* **127.0.0.1**. TCPIP no longer supports an alternative loopback address of 14.0.0.0.

    Existing TCPIP DATA files should be reviewed for **NSINTERADDR** statements that currently include a 14.0.0.0 loopback address. All such statements should be modified to instead make use of the conventional 127.0.0.1 address that is now supported/in use.

5.2.4.3 FTP Client

5.2.4.3.1 Changes Introduced in TCP/IP Level 430

- The FTP client includes support for a **WIDTH** command option that allows more user control over how FTP console output is formatted.

5.2.4.3.2 Changes Introduced in TCP/IP Level 3A0

- The FTP client includes support for the **SIZE** subcommand, which allows a client to obtain the size of a file before it is retrieved.

5.2.4.3.3 Changes Introduced in TCP/IP Function Level 320

- The FTP client has been enhanced to make use of login information present in a NETRC DATA file. By default, when a connection is made to a foreign host, user ID and password information in the NETRC DATA file that is specific to that host are used as part of an automatic FTP login process. Automatic FTP login can be suppressed using the new **NONETRC** option, or through use of the new **NETRC** subcommand.

5.2.4.4 Printing

5.2.4.4.1 Changes Introduced in TCP/IP Function Level 310

- The LPR command can now route print files to RSCS for printing, freeing a user's virtual machine for other work. This may introduce the need for users to learn a limited set of RSCS commands in order to determine the status of their print files. Refer to the *z/VM: TCP/IP Level 430 User's Guide* for more information.

- RSCS can be used instead of LPSERVE to provide enhanced printer daemon (LPD) capabilities. The operation and administration characteristics of RSCS are very different from LPSERVE. Refer to the chapter titled "Configuring the RSCS Print Server" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information.

5.2.4.5 General TCP/IP Server Configuration

5.2.4.5.1 Changes Introduced in TCP/IP Function Level 310

- Separate server initialization execs (TCPIPXIT, FTPDEXIT, etc.) are no longer used. All server parameters and features are controlled by entries contained in a DTCPARMS file. There is support for you to supply an exit which is specific to a particular server, or an exit that is used by all servers, for customization needs that cannot be met using a customized SYSTEM DTCPARMS file.

- Changing server names (when defining a second set of TCP/IP servers, for example) no longer requires changes to IBM-provided execs. All information related to the user ID of a particular server is kept in the TCPIP DATA file, or is part of the server configuration and is maintained in the DTCPARMS file.

- External Security Manager (ESM) interfaces have been standardized across all servers. The RPIUCMS command is used to initialize the RACROUTE environment, and RPIVAL is used to validate

user IDs and passwords supplied by clients. These commands can be changed using a customized SYSTEM DTCPARMS file.

- The ESM environment is automatically established for a server when `:ESM_Enable.YES` is specified for that server in a customized SYSTEM DTCPARMS file.

Refer to the chapter titled "TCP/IP Server Configuration" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information.

5.2.4.6 DNS Server

5.2.4.6.1 Changes Introduced in TCP/IP Level 3A0

- The use of a cache file (as used with a CACHINGONLY name server configuration) has been expanded to the PRIMARY and SECONDARY configurations. A new **ROOTCACHE** statement allows a cache file to be specified in a manner similar to that which can be specified on the CACHINGONLY statement. A corresponding sample root cache file, NSMAIN SCACHE, supplies several configuration recommendations, a list of root name servers, and the Internet site from which the most current list can be obtained.

- A **FORWARDERS** statement has been added to improve name resolution capability for z/VM hosts which are connected to other hosts that provide network firewall services. This statement can be used to identify other name server hosts which can perform name resolution outside the firewall system.

- The algorithms used in the caching subsystem have been improved to facilitate faster access to cached information and to more quickly determine when cached information does not exist.

5.2.4.7 FTP Server

5.2.4.7.1 Changes Introduced in TCP/IP Level 3A0

- The FTP server now makes use of a server-specific configuration file (**SRVRFTP CONFIG** by default) for its startup information, in a manner more consistent with other TCP/IP servers. Therefore, **:parms.** entries within the DTCPARMS file that are associated with the FTP server need be modified to account for this change. Any SRVRFTP operands currently specified using a `:parms.` entry need to be removed, with appropriate modifications made to either the FTP server configuration file or the DTCPARMS file (for example, to account for use of the ANONYMOU or RACF command operands).

  In addition, the FTP server no longer makes use of the client-oriented FTP DATA file.

- The FTP server has been enhanced to better accommodate web browser and graphical FTP clients. Changes for this support include the ability to provide UNIX-format list information in response to client DIR subcommand requests, through the use of the **LISTFORMAT** configuration statement. In addition, the **AUTOTRANS** statement can be used to configure the server to perform automatic EBCDIC-ASCII data translation for transferred files, based on file types (or, file extensions), as determined by **VMFILETYPE** statements which are maintained in the TCPIP DATA file.

- Support for **AUTOTRANS** and **LISTFORMAT** operands to client-supplied SITE subcommands has been implemented as well, to provide client control over automatic EBCDIC-ASCII data translation and list information formats.

- **AUTOTRANS** and **LISTFORMAT** characteristics can now be established when specific FTP users login, when the CHKIPADR exit exec is customized for use.

- Support for additional SMSG interface commands has been added to allow for dynamic server configuration changes, console and tracing control, and shutdown/restart capability.

5.2.4.7.2 Changes Introduced in TCP/IP Function Level 320

- The FTP server has been enhanced to exploit new CP and CMS user authorization facilities provided with VM/ESA Version 2 Release 4.0. These enhancements allow an FTP user to access minidisks they own without the need for minidisk passwords to be defined or supplied during an FTP session. Thus, the link results achieved when FTP users access minidisk resources may noticeably differ from those seen using prior levels of TCP/IP for VM.

  For example, if a user establishes a read-only link to a minidisk (through the use of an explicit "**ACCT** *read_password*" command), a subsequent PUT command that initiates a write request to that minidisk may now succeed *if* the login user ID owns the minidisk in question. By comparison, with a prior TCP/IP level, such a request would cause an FTP user to be prompted to supply a Read/Write (or Multiple Read) password through use of the **ACCT** subcommand in order to first gain write access to the minidisk.

  The use of these new user authorization facilities also allows users with LOGONBY authority for a given *base* virtual machine to exercise that same authority during an FTP session. This is accomplished through use of the CP Diagnose X'88' command by the FTP server, which allows access to *base* user ID resources without requiring the password for that *base* user ID to be supplied. This kind of access is achieved through use of a new "/BY/*byuser_id*" parameter of the FTP **USER** subcommand. Additionally, the CP directory entry for the FTP server must include an OPTION DIAG88 statement, to allow use of the Diagnose X'88' command.

- FTP virtual reader support has been added, which allows an FTP client to use the virtual reader of a VM user ID as the current directory. For such users to issue DELETE and DIR or LS commands against a reader directory, the FTP server virtual machine must have class D privileges. To allow user to PUT files to a reader directory, the **RDR** parameter must be included with the FTP server startup command, SRVRFTP.

5.2.4.8 Kerberos Server

5.2.4.8.1 Changes Introduced in TCP/IP Level 3A0

- The non-DES Kerberos functions that were provided with previous levels of IBM TCP/IP for VM are no longer available. Instead, only a DES Kerberos system is supported, with the DES Kerberos functions incorporated as part of the TCP/IP Feature for z/VM.

  Any Kerberos database that was created in a non-DES environment will not work with the DES Kerberos functions supplied as part of TCP/IP Feature for z/VM. The existing non-DES Kerberos database **must** be deleted and then rebuilt using the supplied DES Kerberos functions. Refer to the chapter titled "Configuring the Kerberos Server" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information about building the Kerberos database.

5.2.4.9 MPROUTE Server

5.2.4.9.1 Changes Introduced in TCP/IP Level 3A0

- The MPROUTE and RouteD servers cannot be concurrently used with the same TCP/IP stack server.

5.2.4.10 RouteD Server

5.2.4.10.1 Changes Introduced in TCP/IP Level 3A0

- The RouteD and MPROUTE servers cannot be concurrently used with the same TCP/IP stack server.

5.2.4.10.2 Changes Introduced in TCP/IP Function Level 320

- The **#CP EXTERNAL** command can no longer be used to terminate RouteD server. The command to stop RouteD is **#CP SMSG * SHUTDOWN**.

5.2.4.11 SMTP Server

5.2.4.11.1 Changes Introduced in TCP/IP Level 420

- Support for new **FILESPERCONN** and **MAXCONNPERSITE** statements has been added. These statements allow more control over the establishment of connections with remote SMTP servers, and may aid with adjusting mail delivery throughput.

- An **SMSG REFRESH** command can now be used by authorized users to dynamically update SMTP security or nickname table information, depending on the configuration of the SMTP server.

5.2.4.11.2 Changes Introduced in TCP/IP Level 3A0

- Timing values for the **RETRYAGE** and **WARNINGAGE** statements can now be specified in hours or minutes (in addition to a number of *days*), through the use of additional **H** or **M** statement operands. Existing defaults (specified in days, for which a **D** statement operand can now be specified) remain unchanged.

5.2.4.11.3 Changes Introduced in TCP/IP Function Level 320

- Support for the SMTP **EHLO** command has been added, as has support for the Message Size Declaration and 8-bit MIME transport service extensions. With this support, SMTP now accepts and handles the SIZE and BODY parameters on MAIL FROM: commands.

5.2.4.11.4 Changes Introduced in TCP/IP Function Level 310

- The **DEBUG** configuration file statement is no longer supported. This statement has been replaced by the **TRACE** statement which, when specified with the DEBUG parameter, provides identical function as the DEBUG statement, except that output goes to only the console; there is no debug file support. Refer to the chapter titled "Configuring the SMTP Server" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information about the TRACE statement and additional parameters that can be specified.

- The default for the **WARNINGAGE** configuration file statement has been reduced from 3 days to 1 day.

- Due to the introduction of new and changed trace options, the **TRACE** SMSG command now requires additional options to be specified. In TCP/IP for VM releases prior to TCP/IP Function Level 310, the **TRACE** SMSG command was used to trace host name resolution via name severs. This SMSG command has been replaced with the **TRACE RESOLVER** SMSG command. Similar results can be achieved by specifying the TRACE RESOLVER statement in the SMTP configuration file. Refer to the chapter titled "Configuring the SMTP Server" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information about configuration file changes, and to the *z/VM: TCP/IP Level 430 User's Guide* for SMTP SMSG command interface changes.

- The SMTP DATA file is no longer needed. The **ATSIGN** statement previously supported using this file has been incorporated within the TCPIP DATA file.

- The sample SMTP configuration file (SMTP SCONFIG) now specifies "NETDATA" as the default for both the **LOCALFORMAT** and **RSCSFORMAT** statements, to reflect the operational default value associated with these statements.

- Host names for mail items can now be re-resolved (when required) by using the **REPROCESS** SMSG command. Refer to the chapter titled "Configuring the SMTP Server" in *z/VM: TCP/IP Level 430 Planning and Customization* for more information. The formerly documented SMTP-EXP EXEC is no longer needed for this purpose and **should not be used**.

- SMTP work files (NOTE and ADDRBLOK files) have new formats and names; the file types for these files have been changed to NOTEFILE and ADDRFILE, respectively. When migrating from IBM TCP/IP for VM Version 2 Release 4, any NOTE or ADDRBLOK files present on the SMTP server A-disk are converted to the new format and renamed; such converted files cannot be processed by previous versions of TCP/IP for VM.

5.2.4.12 TCP/IP (Stack) Server

5.2.4.12.1 Changes Introduced in TCP/IP Level 430

- The **NETSTAT** command now supports an **OBEY** option, which can be used to make dynamic changes to the TCP/IP stack server configuration. Any data that is appropriate for use with the OBEYFILE command can be processed using a NETSTAT OBEY command (to the extent that data can be provided via the CMS command line).

- The **IFCONFIG** command introduced with this level of TCP/IP can be used to make dynamic configuration changes to network interfaces for the z/VM TCP/IP stack, or to display the current configuration. Note that network changes made using IFCONFIG are *not permanent*. However, IFCONFIG can produce data that is compatible with the TCP/IP server configuration file, which then can be used for permanent modifications.

5.2.4.12.2 Changes Introduced in TCP/IP Level 420

- TCPIP has been changed to recognize a TCP/IP loopback address of *only* **127.0.0.1**. TCPIP no longer supports an alternative loopback address of 14.0.0.0.

Existing TCPIP DATA files should be reviewed for **NSINTERADDR** statements that currently include a 14.0.0.0 loopback address. All such statements should be modified to instead make use of the conventional 127.0.0.1 address that is now supported/in use.

- The number of **DEVICE** statements that can be specified within the TCP/IP server configuration file is no longer limited to 100 such statements. The number of DEVICE statements that can be supported is now determined by the virtual storage size of the TCPIP virtual machine.

- Proxy ARP support (activated through use of the **PROXYARP** operand of the **ASSORTEDPARMS** statement) can now be used with OSA Direct (OSD) devices, as well as for more traditional point-to-point connections.

- The default for the **SCANINTERVAL** operand of the **INTERNALCLIENTPARMS** statement (used for Telnet server configuration) has been changed from 120 to 60 seconds.

5.2.4.12.3 Changes Introduced in TCP/IP Function Level 320

- The **TIMESTAMP** statement default has been changed from TIMESTAMP 0 to TIMESTAMP PREFIX, which specifies that a time stamp is to preface every trace and console message. This change helps in diagnosing problems and isolating error conditions. Therefore, it is recommended that any existing TCP/IP server configuration files be changed to specify TIMESTAMP PREFIX to aid in problem determination.

- The Telnet session connection exit interface has been changed to pass the LU name supplied by a client (if any) as an additional parameter. Existing exits may need to be changed to accommodate this behavior.

- The Telnet printer management exit is called for any printer session, regardless of whether the client LU name and IP address are defined by a TN3270E statement in the TCP/IP configuration file. Existing exits may need to be changed to accommodate this behavior.

5.3 DASD Storage and User ID Requirements

Figure 10 on page 26 lists the user IDs and minidisks used to install and service TCP/IP for z/VM.

Important Installation Notes:

1. The user IDs necessary for installing and using TCP/IP for z/VM have been defined as part of the installed z/VM Version 4 Release 3.0 System DDR. Likewise, all required minidisks have been defined. These resources have been listed in Figure 10 so you are aware of the resources that have been allocated on your behalf.

   For information about specific user ID directory entry requirements, consult the **4TCPIP30 PLANINFO** file. This file is located on the 4TCPIP30 191 minidisk.

> **Note — z/VM Express Service Support**
>
> If z/VM Express Service support will be used for applying service for your system, the TCP/IP for z/VM user IDs and minidisks already defined must not be modified. Such changes, if made, will negate use of the z/VM Express Service procedures.

2. The **4TCPIP30** user ID is the IBM-supplied user ID for installing and servicing TCP/IP for z/VM. If you choose to use a different user ID or you elect to use different minidisks and/or SFS directories for TCP/IP for z/VM maintenance purposes, review the information presented in Appendix D, "Modifying the TCP/IP for z/VM Default Installation" on page 92 prior to making any changes.

3. The minidisks that are associated with the **4TCPIP30**, **TCPMAINT**, and **TCPIP** user IDs (or your chosen equivalents) **must** be maintained in order to provide TCP/IP services within your environment. The remaining user IDs listed in Figure 10 are associated with servers and clients that provide optional TCP/IP for z/VM services. If you choose to not use a particular optional service, you need not maintain the user IDs and production minidisks associated with that service. However, be sure to review the information presented in Appendix D, "Modifying the TCP/IP for z/VM Default Installation" on page 92 prior to making any changes.

4. If you choose to use user IDs for TCP/IP server virtual machines that differ from the IBM-supplied user IDs shown in Figure 10, review the section titled "Implications of Assigning Different Server Virtual Machine Names" in Chapter 1 of *z/VM: TCP/IP Level 430 Planning and Customization*. Also, review the information presented in Appendix D, "Modifying the TCP/IP for z/VM Default Installation" on page 92.

5. Note the following, with regard to the user ID and minidisk information provided in Figure 10:

> **Specific Minidisk Requirements**
>
> Certain minidisks **must** be defined for the TCP/IP server machines you choose to use, as well as for maintaining TCP/IP for z/VM in your environment. Such minidisks **cannot** be replaced with an equivalent SFS directory.
>
> Minidisks to which this requirement applies are listed in Figure 10 with **boldface** virtual device numbers. In addition, no default SFS directory names or SFS 4K block values are provided for these minidisks.

6. **All** 4TCPIP30 *test build* minidisks **must** be maintained. If the 4TCPIP30 minidisks for optional services are not maintained, problems will be encountered during installation and service.

   If you choose to eliminate any of the resources which correspond to TCP/IP services that are not required for your installation, review the considerations presented in Appendix D, "Modifying the TCP/IP for z/VM Default Installation" on page 92 prior to making any changes.

7. Additional storage may need to be allocated for a given user ID or server minidisk, depending on your environment. Some examples of minidisks that may need to increased, and possible reasons for so doing, are listed in Figure 8. Note that certain minidisks (not cited here) may also need to be increased to accommodate the logging of tracing or other activities.

Figure 8. Alternative Minidisk Storage Requirements

User ID / Minidisk	Rationale for Storage Revision
SMTP 191	Allow for SMTP processing of a high volume of e-mail
VMNFS 191	Provide support for a large number of NFS clients
DHCPD 191	Provide support a large number of DHCP clients
LPSERVE 191	Allow for LPD processing of sizeable print jobs
SSLSERV 201	Facilitate installation of a Linux kernel and file system
4TCPIP30 2D2	Facilitate maintenance of multiple service levels of (binary) system files used by the SSL server

8. The storage requirements for various TCP/IP minidisks may be revised over time to account for TCP/IP for z/VM content changes. In some cases, this may require the size of existing minidisks to be increased.

   When applicable, storage requirement changes for specific minidisks will be identified in Figure 10, in updated levels of this document. Such changes will be noted through the use of revision characters (usually a vertical bar — "I" in the left margin of a page).

   For other minidisks, storage requirement changes will need to be assessed locally, for your specific environment. For example, the capacity of the **DELTA** minidisk (**4TCPIP30 2D2**, by default), periodically may need to be increased, based on the specific preventive and corrective service applied to your system.

9. If you choose to provide Network Database services, you may find the need to define multiple virtual machines, named NDBSRV01, NDBSRV02, etc. Each NDBSRV*nn* virtual machine you create should be defined similar to NDBSRV01, which has been defined as part of the installed z/VM Version 4 Release 3.0 System DDR.

10. If you choose to provide remote execution services through use of the rexec daemon (REXECD), you may find the need to define multiple agent virtual machines, named RXAGENT1, RXAGENT2, etc. Each RXAGENT*n* virtual machine you create should be defined similar to RXAGENT1, which has been defined as part of the installed z/VM Version 4 Release 3.0 System DDR. However, note that the RXAGENT*n* virtual machines do not "own" any minidisks.

11. To use the Secure Socket Layer (SSL) server, a suitably configured Linux kernel and file system must be installed on your z/VM system. Information about Linux requirements and preparing Linux for use with the SSL server is available at the TCP/IP for z/VM home page on the World Wide Web. The URL for this home page is:

    `http://www.vm.ibm.com/related/tcpip/`

12. Source files are supplied in packed format. If you intend to unpack source files after installation, ensure sufficient space is allocated for the unpacked files. Revised storage requirements for storing unpacked files on the TCP/IP **SOURCE** minidisk (**4TCPIP30 2B3**, by default) are listed in Figure 9:

Figure 9. 4TCPIP30 2B3 Minidisk Storage Requirements — Unpacked Source Files

Type of Storage	Revised Storage Requirement
3390 DASD	80 cylinders
3380 DASD	96 cylinders
9345 DASD	96 cylinders
FBA Device	114080 FB-512 blocks
SFS Directory	14260 SFS 4K blocks

13. For information about copying client code to the Product Code minidisk, see Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113.

5.3.1 DASD Requirements for TCP/IP for z/VM

Minidisk owner (User ID)	Default Device Number	Storage in Cylinders		FB-512 Blocks	SFS 4K Blocks	Usage Default SFS Directory Name
		DASD	CYLS			
4TCPIP30	191	3390	50	72000	9000	4TCPIP30 user ID 191 minidisk
		3380	60			
		9345	60			**VMSYS:4TCPIP30**
4TCPIP30	2B2	3390	86	123200	15400	Contains all base code shipped with TCP/IP for z/VM
		3380	103			
		9345	103			**VMSYS:4TCPIP30.TCPIP.OBJECT**
4TCPIP30	2B3	3390	55	78400	9800	Source files disk. (3*)
		3380	66			
		9345	66			**VMSYS:4TCPIP30.TCPIP.SOURCE**
4TCPIP30	2C4	3390	5	6000	750	Contains local modifications
		3380	5			
		9345	5			**VMSYS:4TCPIP30.TCPIP.LOCAL**
4TCPIP30	2D2	3390	117	167200	20900	Contains serviced files (3*)
		3380	140			
		9345	140			**VMSYS:4TCPIP30.TCPIP.DELTA**
4TCPIP30	2A6	3390	5	6000	750	Contains AUX files and software inventory tables that represent the test service level of TCP/IP for z/VM
		3380	5			
		9345	5			**VMSYS:4TCPIP30.TCPIP.APPLYALT**
4TCPIP30	2A2	3390	5	6000	750	Contains AUX files and software inventory tables that represent the service level of TCP/IP for z/VM that is currently in production
		3380	5			
		9345	5			**VMSYS:4TCPIP30.TCPIP.APPLYPROD**

Figure 10 (Page 1 of 5). DASD Storage Requirements for Target Minidisks - TCP/IP for z/VM

Notes:

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

2. For installation to SFS directories, a total of **54150** 4K blocks are required.

3. Additional storage may need to be allocated for some minidisks, depending on your environment. For more information, see the accompanying notes on page 22.

4. See Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for information about copying client code to the Product Code minidisk

| Minidisk owner (User ID) | Default Device Number | Storage in Cylinders | | FB-512 Blocks | SFS 4K Blocks | Usage |
		DASD	CYLS			Default SFS Directory Name
4TCPIP30	**491**	3390 3380 9345	38 45 45	53600	_____	**Test** build disk for server code; files from this disk are copied to a production disk (TCPMAINT 591) which also requires this amount of free space
4TCPIP30	**492**	3390 3380 9345	67 80 80	96000	_____	**Test** build disk for client code; files from this disk are copied to a production disk (TCPMAINT 592) which also requires this amount of free space
TCPMAINT	**191**	3390 3380 9345	7 8 8	9600	_____	TCPMAINT user ID 191 minidisk
TCPMAINT	**198**	3390 3380 9345	9 10 10	12000	_____	Contains configuration files for clients and servers.
TCPMAINT	**591**	3390 3380 9345	38 45 45	53600	_____	**Production** build disk for server code
TCPMAINT	**592**	3390 3380 9345	67 80 80	96000	_____	**Production** build disk for client code
ADMSERV	**191**	3390 3380 9345	5 5 5	6000	_____	ADMSERV user ID 191 minidisk
BOOTPD	**191**	3390 3380 9345	2 2 2	2400	_____	BOOTPD user ID 191 minidisk
DHCPD	**191**	3390 3380 9345	2 2 2	2400	_____	DHCPD user ID 191 minidisk (3*)

Figure 10 (Page 2 of 5). DASD Storage Requirements for Target Minidisks - TCP/IP for z/VM

Notes:

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

2. For installation to SFS directories, a total of **54150** 4K blocks are required.

3. Additional storage may need to be allocated for some minidisks, depending on your environment. For more information, see the accompanying notes on page 22.

4. See Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for information about copying client code to the Product Code minidisk

| Minidisk owner (User ID) | Default Device Number | Storage in Cylinders | | FB-512 Blocks | SFS 4K Blocks | Usage |
		DASD	CYLS			Default SFS Directory Name
FTPSERVE	**191**	3390 3380 9345	9 10 10	12000	_____	FTPSERVE user ID 191 minidisk
IMAP	**191**	3390 3380 9345	1 1 1	1200	_____	IMAP user ID 191 minidisk
LPSERVE	**191**	3390 3380 9345	2 2 2	2400	_____	LPSERVE user ID 191 minidisk (3*)
MPROUTE	**191**	3390 3380 9345	2 2 2	2400	_____	MPROUTE user ID 191 minidisk
NAMESRV	**191**	3390 3380 9345	2 2 2	2400	_____	NAMESRV user ID 191 minidisk
NDBPMGR	**191**	3390 3380 9345	1 1 1	1200	_____	NDBPMGR user ID 191 minidisk
NDBSRV01	**191**	3390 3380 9345	1 1 1	1200	_____	NDBSRV01 user ID 191 minidisk
PORTMAP	**191**	3390 3380 9345	2 2 2	2400	_____	PORTMAP user ID 191 minidisk
REXECD	**191**	3390 3380 9345	2 2 2	2400	_____	REXECD user ID 191 minidisk

Figure 10 (Page 3 of 5). DASD Storage Requirements for Target Minidisks - TCP/IP for z/VM

Notes:

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

2. For installation to SFS directories, a total of **54150** 4K blocks are required.

3. Additional storage may need to be allocated for some minidisks, depending on your environment. For more information, see the accompanying notes on page 22.

4. See Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for information about copying client code to the Product Code minidisk

| Minidisk owner (User ID) | Default Device Number | Storage in Cylinders | | FB-512 Blocks | SFS 4K Blocks | Usage |
		DASD	CYLS			Default SFS Directory Name
ROUTED	**191**	3390 3380 9345	2 2 2	2400	_____	ROUTED user ID 191 minidisk
RXAGENT1	_____	3390 3380 9345	__ __ __	_____	_____	REXEC agent (a 191 minidisk is **not** required; REXEC agents utilize the REXECD 191 minidisk)
SMTP	**191**	3390 3380 9345	25 30 30	36000	_____	SMTP user ID 191 minidisk (3*)
SNALNKA	**191**	3390 3380 9345	3 3 3	3600	_____	SNALNKA user ID 191 minidisk
SNMPD	**191**	3390 3380 9345	2 2 2	2400	_____	SNMPD user ID 191 minidisk
SNMPQE	**191**	3390 3380 9345	2 2 2	2400	_____	SNMPQE user ID 191 minidisk
SSLSERV	**191**	3390 3380 9345	1 1 1	1200	_____	SSLSERV user ID 191 minidisk
SSLSERV	**201**	3390 3380 9345	1 1 1	1200	_____	SSLSERV user ID 201 minidisk (3*)
SSLSERV	**202**	3390 3380 9345	30 36 36	42400	_____	SSLSERV user ID 202 minidisk

Figure 10 (Page 4 of 5). DASD Storage Requirements for Target Minidisks - TCP/IP for z/VM

Notes:

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

2. For installation to SFS directories, a total of **54150** 4K blocks are required.

3. Additional storage may need to be allocated for some minidisks, depending on your environment. For more information, see the accompanying notes on page 22.

4. See Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for information about copying client code to the Product Code minidisk

| Minidisk owner (User ID) | Default Device Number | Storage in Cylinders | | FB-512 Blocks | SFS 4K Blocks | Usage |
		DASD	CYLS			Default SFS Directory Name
SSLSERV	**203**	3390	1	1400	_____	SSLSERV user ID 203 minidisk
		3380	2			
		9345	2			
SSLSERV	**204**	3390	15	21200	_____	SSLSERV user ID 204 minidisk
		3380	18			
		9345	18			
SSLSERV	**205**	3390	140	200800	_____	SSLSERV user ID 205 minidisk
		3380	168			
		9345	168			
TCPIP	**191**	3390	5	6000	_____	TCPIP user ID 191 minidisk
		3380	5			
		9345	5			
TFTPD	**191**	3390	2	2400	_____	TFTPD user ID 191 minidisk
		3380	2			
		9345	2			
UFTD	**191**	3390	2	2400	_____	UFTD user ID 191 minidisk
		3380	2			
		9345	2			
VMKERB	**191**	3390	6	8400	_____	VMKERB user ID 191 minidisk
		3380	7			
		9345	7			
VMNFS	**191**	3390	9	12000	_____	VMNFS user ID 191 minidisk [3*]
		3380	10			
		9345	10			
X25IPI	**191**	3390	2	2400	_____	X25IPI user ID 191 minidisk
		3380	2			
		9345	2			

Figure 10 (Page 5 of 5). DASD Storage Requirements for Target Minidisks - TCP/IP for z/VM

Notes:

1. Cylinder values defined in this table are based on a 4K block size. FB-512 block and SFS values are derived from the 3380 cylinder values in this table. FBA minidisk sizes are shown in 512-byte blocks; these minidisks should be CMS formatted at 1K size.

2. For installation to SFS directories, a total of **54150** 4K blocks are required.

3. Additional storage may need to be allocated for some minidisks, depending on your environment. For more information, see the accompanying notes on page 22.

4. See Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for information about copying client code to the Product Code minidisk

6.0 Installation Instructions

This section describes the method by which TCP/IP for z/VM is installed and provides step-by-step procedures to complete the installation process.

The procedures that follow are presented in two-column format, where the steps to be performed are identified using numbered, **boldface** headings. Any sub-steps that correspond to a given procedure are presented on the right side of each page and are ordered using bold numerals, while the commands associated with these steps are presented on the left side of a page. Pertinent command information may exist to the right of a given command. For more information about the two-column format see "Understanding Dialogs with the System" in the *z/VM: Installation Guide*.

Each step of these installation instructions must be followed. Do not skip any step unless directed otherwise.

Throughout these instructions, the use of IBM-supplied default minidisk device numbers and user IDs is assumed. If different user IDs, device numbers, or SFS directories are used to install TCP/IP for z/VM in your environment, adapt these instructions as needed.

Note!

Any sample console output presented throughout these instructions is based on a z/VM Version 4 Release 3.0 system; this output reflects an installation environment in which default values (PPF and component names, user IDs, and minidisks) are in use.

6.1 TCP/IP for z/VM Installation Process Overview

A brief description of the steps necessary to complete the installation of TCP/IP for z/VM follows:

- **Review the Default Installation** — Various resources have been defined and allocated for TCP/IP for z/VM, as part of the installed z/VM Version 4 Release 3.0 System DDR. This default environment should be reviewed and, if necessary, modified for your installation.

- **Review TCP/IP for z/VM Content and Changes** — Review the topics presented in 5.2.4, "Migration Considerations" on page 14, so you are aware of changes that may affect your customization and use of TCP/IP Level 430.

- **Configure TCP/IP for z/VM** — The configuration files associated with the various TCP/IP services must be customized to effectively use TCP/IP for z/VM.

For a complete description of all VMSES/E installation commands, operands and options, refer to:

- *z/VM: VMSES/E Introduction and Reference* (GC24-5994)

6.2 Installing TCP/IP for z/VM

┌─ Note — All z/VM Customers ───┐

The material presented in the next few sections is provided mostly for informational and references
purposes. To complete the installation of TCP/IP for z/VM, continue with the instructions in section
6.2.2.1, "Configure TCP/IP for z/VM for Your Installation" on page 36.

└──┘

6.2.1 Review the TCP/IP for z/VM Default Installation Environment

Because TCP/IP for z/VM has been installed as part of the z/VM Version 4 Release 3.0 system DDR,
several installation steps have already been performed on your behalf. Among these are the:

- addition of TCP/IP-specific user ID entries and PROFILES to the z/VM Version 4 Release 3.0 system
  directory
- creation of a simplified PROFILE EXEC for the 4TCPIP30 user ID
- allocation of TCP/IP-required minidisks
- loading of TCP/IP for z/VM product files (run-time and sample configuration files) to test build *and*
  production minidisks, using VMSES/E commands.

6.2.1.1 PPF Override and Other Modification Considerations

The file name (or, *ppfname*) of IBM-supplied Product Parameter File (PPF) for TCP/IP for z/VM is
4TCPIP30. This file has been installed (and used) as part of the z/VM Version 4 Release 3.0 system
DDR installation. The **4TCPIP30** *ppfname* is also assumed and referenced throughout section 7.0,
"Service Instructions."

If you create your own TCP/IP for z/VM PPF override file, use the *ppfname* of your override file (instead of
4TCPIP30) throughout any procedures that require this file to be identified, unless noted otherwise.

**Note: PPF Overrides for TCP/IP for z/VM must not be used on systems that use z/VM Express
Service support**.

As *z/VM: TCP/IP Level 430 Planning and Customization* is reviewed and used to configure TCP/IP, you
may also identify TCP/IP services that are not required for your installation. If you choose to eliminate the
resources that correspond to such services, review the considerations presented in Appendix D,
"Modifying the TCP/IP for z/VM Default Installation" on page 92 prior to making any changes.

6.2.1.2 TCP/IP for z/VM Directory PROFILES and User IDs

Two system directory PROFILE entries (PROFILE TCPCMSU and PROFILE TCPGCSU) have been
added to the z/VM Version 4 Release 3.0 system directory for TCP/IP for z/VM; these entries are shown in
Figure 11. Each directory entry supplied for a TCP/IP for z/VM service virtual machine includes one of
these profiles.

Figure 11. TCP/IP for z/VM System Directory Profiles

```
PROFILE TCPCMSU                          PROFILE TCPGCSU
  IPL CMS                                  IPL GCSXA PARM AUTOLOG
  MACHINE XA                               MACHINE XA
  SPOOL 00C 2540 READER *                  NAMESAVE GCS
  SPOOL 00D 2540 PUNCH  A                   SPOOL 00C 2540 READER *
  SPOOL 00E 1403 A                         SPOOL 00D 2540 PUNCH  A
  CONSOLE 009 3215 T                       SPOOL 00E 1403 A
  LINK MAINT 190 190 RR                    CONSOLE 009 3215 T
  LINK MAINT 19D 19D RR                    LINK MAINT 190 190 RR
  LINK MAINT 19E 19E RR                    LINK MAINT 19D 19D RR
  LINK MAINT 0401 0401 RR                  LINK MAINT 19E 19E RR
  LINK MAINT 0402 0402 RR                  LINK MAINT 0401 0401 RR
  LINK MAINT 0405 0405 RR                  LINK MAINT 0402 0402 RR
                                           LINK MAINT 0405 0405 RR
```

Notes:

1. Links to the MAINT 401, 402 and 405 minidisks are established to facilitate the use of CMS Kanji, German and Upper Case American English HELP files, for those environments in which these may be required.

2. The `NAMESAVE GCS` statement can be removed from PROFILE TCPGCSU if the GCS saved segment is not restricted.

The user IDs that have been defined for TCP/IP for z/VM are listed in

Note: When installation of the z/VM Version 4 Release 3.0 System DDR has been completed, the login password for a given TCP/IP user ID is identical to that same user ID. If you have not already done so, change these passwords to valid passwords, in accordance with your security guidelines.

Additional User ID Notes:

1. For information about specific user ID directory entry requirements, consult the **4TCPIP30 PLANINFO** file. This file is located on the 4TCPIP30 191 minidisk.

2. The directory entry for the TCPIP virtual machine includes the statement: `SHARE RELATIVE 3000`

   For most installations, the relative CPU share allocation of 3000 should be suitable. However, you are free to change this value to conform to local guidelines established for defining server and guest virtual machine share settings.

3. If you create additional RXAGENT*n* machines, duplicate the RXAGENT1 directory entry for each server you add.

4. If you create additional NDBSRV*nn* machines, duplicate the NDBSRV01 directory entry for each server you add, and include an appropriate LINK statement in the 4TCPIP30 directory entry for each new server 191 minidisk that is created.

```
┌─ Formatting Reminder ──────────────────────────────────────────────────────────
│
│  Any additional minidisks you create must be formatted before you continue with the installation of
│  TCP/IP for z/VM.
│
└──────────────────────────────────────────────────────────────────────────────────
```

TCP/IP User ID	Associated TCP/IP Function
Figure 12 (Page 1 of 2). Default User IDs - TCP/IP for z/VM	
4TCPIP30	Maintains the TCP/IP system.
TCPMAINT	TCP/IP system ownership.
ADMSERV	Runs the Kerberos database remote administration server.
BOOTPD	Responds to client requests for boot information using data defined in a BOOTP machine file.
DHCPD	Responds to client requests for boot information using data defined in a DHCPD machine file.
FTPSERVE	Implements the File Transfer Protocol (FTP) daemon, which controls access to files on the local host.
IMAP	Implements the Internet Message Access Protocol (IMAP) daemon, which allows a client to access and manipulate electronic mail messages on a server.
LPSERVE	Implements the Line Printer Daemon (LPD), which handles client requests to print a file.
MPROUTE	Implements the Multiple Protocol ROUTE (MPROUTE) server, which uses OSPF and/or RIP protocols to manage network routing information.
NAMESRV	Implements the Domain Name System (DNS) server.
NDBPMGR	Provides Network Database (NDB) Port Manager support, and is used in conjunction with the NDBSRV01 server.
NDBSRV01 [1*]	Provides Network Database (NDB) System support, in conjunction with the NDB Port Manager (NDBPMGR) server.
PORTMAP	Runs the Portmapper function for RPC systems that support the Network File System protocol.
REXECD	Provides remote execution services for TCP/IP hosts that support the REXEC client.
ROUTED	Implements the RouteD server, which manages IP route table entries.
RXAGENT1 [1*]	Agent virtual machine used by REXECD to process anonymous rexec client requests.
SMTP	Implements the Simple Mail Transfer Protocol (SMTP) server, which provides TCP/IP electronic mail support.
SNALNKA	Provides SNA LU 0 connections between multiple hosts.
SNMPD	Virtual machine for the SNMP Agent.
SNMPQE	Virtual machine for the SNMP Query Engine.
SSLSERV	Provides Secure Sockets Layer (SSL) protocol support for TCP/IP servers.
Notes:	
1. Additional changes may need to be made for some user IDs, depending on your environment. For more information, see the accompanying notes on page 33.	

Figure 12 (Page 2 of 2). Default User IDs - TCP/IP for z/VM	
TCP/IP User ID	**Associated TCP/IP Function**
TCPIP (1*)	Primary virtual machine that provides TCP/IP and Telnet services.
TFTPD	Transfers files between the Byte File System (BFS) and TFTP clients.
UFTD	Implements the Unsolicited File Transfer (UFT) server.
VMKERB	Runs the Kerberos authentication server.
VMNFS	Implements the Network File System (NFS) server.
X25IPI	Provides an interface which allows the TCPIP virtual machine to communicate with hosts that use the X.25 protocol.
Notes:	
1. Additional changes may need to be made for some user IDs, depending on your environment. For more information, see the accompanying notes on page 33.	

6.2.2 Move TCP/IP for z/VM to SFS Directories (Optional)

By default, TCP/IP for z/VM is installed to minidisks, as part of the z/VM Version 4 Release 3.0 System DDR. However, you can move TCP/IP for z/VM service minidisks to Shared File System (SFS) directories. See Appendix H, "Moving TCP/IP for z/VM to SFS Directories" on page 108 for instructions on how this can be done.

6.2.2.1 Configure TCP/IP for z/VM for Your Installation

TCP/IP for z/VM has been installed with copies of *sample* configuration files that are used by various TCP/IP servers and clients. Before any TCP/IP services are used, these configuration files must be customized for your environment.

See *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019) for detailed information about the various TCP/IP services that can be configured for your installation, and for details about the configuration files that are associated with each service. For reference, the sample files supplied by IBM are summarized in Figure 14 on page 40.

IPWIZARD Considerations

If the IPWIZARD command has been used to create an initial TCP/IP configuration, the following files have been customized:

- PROFILE TCPIP
- SYSTEM DTCPARMS
- TCPIP DATA

These files enable basic network connectivity for your z/VM system; their content is based on information supplied via the IPWIZARD panels. If you intend to provide more comprehensive TCP/IP services for your installation, the previously listed files will require further customization, as will other TCP/IP configuration files.

If the IPWIZARD command has **not** been used, the PROFILE TCPIP and TCPIP DATA files are present, but contain only sample data; the SYSTEM DTCPARMS file does not yet exist.

6.2.2.2 Initialize TCP/IP Services

Once TCP/IP for z/VM has been (fully) configured for your environment, the appropriate TCP/IP servers must be initialized. For more information, see the section that discusses "Starting and Stopping TCP/IP Servers" in the chapter titled "General TCP/IP Server Configuration," of *z/VM: TCP/IP Level 430 Planning and Customization*.

6.2.2.3 Copy TCP/IP Client Code to the z/VM Product Code Disk (Optional)

After TCP/IP for z/VM has been configured for your environment, you may want to consider copying TCP/IP client code (or a subset of this) to the z/VM Product Code minidisk. See Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for additional information and instructions concerning this process.

6.2.2.4 TCP/IP for z/VM Product and Sample Configuration Files

Figure 13 on page 38 lists the name and location of TCP/IP for z/VM product run-time files provided by IBM, and the names and location of these files as placed into production by the VMSES/E **PUT2PROD** command.

Note: The *target* minidisk link device numbers listed in this table correspond to LINK statement defaults; these defaults are defined for the 4TCPIP30 user ID in the system (CP) directory entry that is supplied as part of the z/VM Version 4 Release 3.0 System DDR.

Figure 13 (Page 1 of 2). TCP/IP for z/VM Production Run-Time Files				
4TCPIP30 Source Minidisk	4TCPIP30 Link Device Number	Source File Name / Type	Production File Name / Type	Target Minidisk
491	591	— **All Files** —	(no change)	———
492	592	— **All Files** —	(no change)	———
491	262	TCPROFIL EXEC	PROFILE EXEC	TCPIP 191
491	263	TCPROFIL EXEC	PROFILE EXEC	ADMSERV 191
491	264	TCPROFIL EXEC	PROFILE EXEC	BOOTPD 191
491	265	TCPROFIL EXEC	PROFILE EXEC	DHCPD 191
491	266	TCPROFIL EXEC	PROFILE EXEC	FTPSERVE 191
491	267	TCPROFIL EXEC	PROFILE EXEC	LPSERVE 191
491	268	TCPROFIL EXEC	PROFILE EXEC	MPROUTE 191
491	269	TCPROFIL EXEC	PROFILE EXEC	NAMESRV 191
491	26A	TCPROFIL EXEC	PROFILE EXEC	NDBPMGR 191
491	26B	TCPROFIL EXEC [2*]	PROFILE EXEC	NDBSRV01 191
491	26C	TCPROFIL EXEC	PROFILE EXEC	PORTMAP 191
491	26D	TCPROFIL EXEC	PROFILE EXEC	REXECD 191
491	26E	TCPROFIL EXEC	PROFILE EXEC	ROUTED 191
491	26F	TCPROFIL EXEC	PROFILE EXEC	SMTP 191
491	270	TCPROFIL GCS [1*]	PROFILE GCS	SNALNKA 191
491	271	TCPROFIL EXEC	PROFILE EXEC	SNMPD 191
491	272	TCPROFIL EXEC	PROFILE EXEC	SNMPQE 191
491	273	TCPROFIL EXEC	PROFILE EXEC	SSLSERV 191

Notes:

1. The TCPROFIL GCS profiles are not interchangeable with the TCPROFIL EXEC used for other TCP/IP servers.

2. TCPROFIL EXEC should be copied to the 191 disk of any additional NDBSRV*nn* servers that are installed.

Figure 13 (Page 2 of 2). TCP/IP for z/VM Production Run-Time Files				
4TCPIP30 Source Minidisk	**4TCPIP30 Link Device Number**	**Source File Name / Type**	**Production File Name / Type**	**Target Minidisk**
491	279	TCPROFIL EXEC	PROFILE EXEC	TFTPD 191
491	27A	TCPROFIL EXEC	PROFILE EXEC	UFTD 191
491	27B	TCPROFIL EXEC	PROFILE EXEC	VMKERB 191
491	27C	TCPROFIL EXEC	PROFILE EXEC	VMNFS 191
491	27D	TCPROFIL GCS (1*)	PROFILE GCS	X25IPI 191
491	27E	TCPROFIL EXEC	PROFILE EXEC	IMAP 191
Notes:				
1. The TCPROFIL GCS profiles are not interchangeable with the TCPROFIL EXEC used for other TCP/IP servers.				
2. TCPROFIL EXEC should be copied to the 191 disk of any additional NDBSRV*nn* servers that are installed.				

Figure 14 on page 40 lists the name and location of TCP/IP for z/VM sample configuration files provided by IBM (and the default names and location for their customized counterparts) as placed into production by the VMSES/E **PUT2PROD** command. The server or user groups for which a given file is required (or may be used) are identified as well.

Note: Unless otherwise noted, the **Sample** and **Configured** disks cited in this table are minidisks owned by the TCPMAINT user ID.

Figure 14 (Page 1 of 2). TCP/IP for z/VM Sample/Production Configuration Files

Sample Disk	Configured Disk	Sample File Name / Type	Configured File Name / Type	Usage
591	198	PROFILE STCPIP	PROFILE TCPIP	TCPIP
592	198	HOSTS SLOCAL	HOSTS LOCAL	TCPIP
591	198	SCEXIT SEXEC	SCEXIT EXEC	TCPIP
591	198	SCEXIT SAMPASM	SCEXIT ASSEMBLE	TCPIP
591	198	PMEXIT SEXEC	PMEXIT EXEC	TCPIP
591	198	PMEXIT SAMPASM	PMEXIT ASSEMBLE	TCPIP
591	191 (1*)	ADM@AADD SAMPAUTH	ADM@ACL ADD	ADMSERV
591	191 (1*)	ADM@AGET SAMPAUTH	ADM@ACL GET	ADMSERV
591	191 (1*)	ADM@AMOD SAMPAUTH	ADM@ACL MOD	ADMSERV
591	198	BOOTPTAB SAMPLE	ETC BOOTPTAB	BOOTPD
591	198	CHKIPADR SEXEC	CHKIPADR EXEC	FTPSERVE
591	198	FTPEXIT SEXEC	FTPEXIT EXEC	FTPSERVE
591	198	FTPEXIT SAMPASM	FTPEXIT ASSEMBLE	FTPSERVE
591	198	SRVRFTP SCONFIG	SRVRFTP CONFIG	FTPSERVE
591	198	IMAP SCONFIG	IMAP CONFIG	IMAP
591	198	TCPVMIPC SAMPNAME	$SERVER$ NAMES	IMAP
591	198	LPD SCONFIG	LPD CONFIG	LPSERVE
591	198	MPROUTE SCONFIG	MPROUTE CONFIG	MPROUTE
591	198	NSMAIN SCACHE	NSMAIN CACHE	NAMESRV
591	198	NSMAIN SDATA	NSMAIN DATA	NAMESRV
591	198	VALIDUSR SEXEC	VALIDUSR EXEC	NAMESRV
591	198	ROUTED SCONFIG	ROUTED CONFIG	ROUTED
591	198	RSCSTCP SCONFIG	RSCSTCP CONFIG	RSCS
591	198	RSCSLPD SCONFIG	RSCSLPD CONFIG	RSCS (LPD)
591	198	RSCSLPR SCONFIG	RSCSLPR CONFIG	RSCS (LPD)
591	198	RSCSLPRP SCONFIG	RSCSLPRP CONFIG	RSCS (LPD)
591	198	RSCSUFT SCONFIG	RSCSUFT CONFIG	RSCS (UFT)
591	198	SMTP SCONFIG	SMTP CONFIG	SMTP
591	198	SMTPCMDX SEXEC	SMTPCMDX EXEC	SMTP

Notes:

1. The ADMSERV 191 minidisk.

2. TCPIP DATA is used by **all** TCP/IP servers and client programs.

3. KRB CONF is used by client applications that utilize Kerberos authentication servers.

Figure 14 (Page 2 of 2). TCP/IP for z/VM Sample/Production Configuration Files

Sample Disk	Configured Disk	Sample File Name / Type	Configured File Name / Type	Usage
591	198	SMTPCMDX SAMPASM	SMTPCMDX ASSEMBLE	SMTP
591	198	SMTPVERX SEXEC	SMTPVERX EXEC	SMTP
591	198	SMTPVERX SAMPASM	SMTPVERX ASSEMBLE	SMTP
591	198	SMTPFWDX SEXEC	SMTPFWD EXEC	SMTP
591	198	SMTPFWDX SAMPASM	SMTPFWDX ASSEMBLE	SMTP
591	198	SMTPMEMO SAMPLE	SECURITY MEMO	SMTP
591	198	SMTPSECT SAMPTABL	SMTP SECTABLE	SMTP
591	198	SNALNKA SAMPGCS	SNALNKA GCS	SNALNKA
591	198	MIB_DESC SDATA	MIB_DESC DATA	SNMPQE
591	198	UFTD SCONFIG	UFTD CONFIG	UFTD
591	198	UFTCMDX SEXEC	UFTCMDX EXEC	UFTD
591	198	UFTNSLKX SEXEC	UFTNSLKX EXEC	UFTD
591	198	VMNFS SCONFIG	VMNFS CONFIG	VMNFS
591	198	VMNFSCMS SEXEC	VMNFSCMS EXEC	VMNFS
591	198	VMNFSSMG SEXEC	VMNFSSMG EXEC	VMNFS
591	198	VMNFSMON SEXEC	VMNFSMON EXEC	VMNFS
591	198	X25IPI SCONFIG	X25IPI CONFIG	X25IPI
591	198	X25IPI SAMPGCS	X25IPI GCS	X25IPI
592	592	TCPIP SDATA	TCPIP DATA [2*]	All Users
592	592	ETC SAMPSERV	ETC SERVICES	All Users
592	592	FTP SDATA	FTP DATA	FTP Users
592	592	KRB SCONFIG	KRB CONF [3*]	Clients

Notes:

1. The ADMSERV 191 minidisk.

2. TCPIP DATA is used by **all** TCP/IP servers and client programs.

3. KRB CONF is used by client applications that utilize Kerberos authentication servers.

Figure 15 lists the name and location of TCP/IP for z/VM (binary) system files provided by IBM, and the default names and location of these files as placed into production by the VMSES/E **PUT2PROD** command. The server that makes use of a given file is identified as well.

Note: The *target* minidisk link device numbers listed in this table correspond to LINK statement defaults; these defaults are defined for the 4TCPIP30 user ID in the system (CP) directory entry that is supplied as part of the z/VM Version 4 Release 3.0 System DDR.

Figure 15. TCP/IP for z/VM Production System Files

4TCPIP30 Source Disk	4TCPIP30 Link Device Number	Source File Name / Type	Production File Name / Type	Target Minidisk
2D2	276	DTCSSLF FLSBIN	DTCSSLF DATABIN	SSLSERV 203
2D2	275	DTCSSLS FLSBIN	DTCSSLS DATABIN	SSLSERV 202

Notes:

1. Files listed in this table are used by only the Secure Socket Layer (SSL) Server.

TCP/IP for z/VM is now installed and built on your system.

7.0 Service Instructions

Note — z/VM Express Service Support

If you have chosen to use z/VM Express Service support for applying service to your z/VM Version 4 Release 3.0 system, you must use the instructions documented in:

- the *z/VM Installation and Service Summary* pamphlet (provided with the *z/VM: Installation Guide*), or

- Appendix J, "z/VM Express Service Instructions" on page 116

to install preventive (RSU) or COR service.

This section describes the method by which TCP/IP for z/VM is serviced; it provides step-by-step procedures to install corrective (COR) and preventive service for TCP/IP for z/VM, using VMSES/E. Preventive service for TCP/IP for z/VM is delivered via a Recommended Service Upgrade (RSU) tape.

To become more familiar with service using VMSES/E, read the introductory chapters in:

- *z/VM: VMSES/E Introduction and Reference* (GC24-5994)

This publication also contains command syntax for the VMSES/E commands cited throughout these instructions.

Each step of these service instructions must be followed. Do not skip any step unless directed otherwise.

Throughout these instructions, the use of IBM-supplied default minidisk device numbers and user IDs is assumed. If different user IDs, device numbers, or SFS directories are used to install TCP/IP for z/VM in your environment, adapt these instructions as needed.

Note!

Any sample console output presented throughout these instructions is based on a z/VM Version 4 Release 3.0 system; this output reflects an installation environment in which default values (PPF and component names, user IDs, and minidisks) are in use.

7.1 VMSES/E Service Process Overview

A brief description of the steps required to service TCP/IP for z/VM, using VMSES/E, follows:

- **Merge Existing Service** — The VMFMRDSK command is used to clear the alternate apply disk before receiving new service. This allows you to easily remove new service if a serious problem is encountered during its testing or use.

- **Receive New Service** — The VMFREC command is used to receive service from the delivery media and place it on the DELTA disk.

- **Apply the Service** — The VMFAPPLY command is used to update the version vector table (VVT), which identifies the service level of all serviced parts. In addition, AUX files are generated for parts that require them, based on the content of the VVT.

- **Reapply Local Service** (if applicable) — All local service (*user modifications*, or "mods") must be entered into the software inventory to allow VMSES/E to track these changes and build them into the system. Refer to Chapter 7 of the *z/VM: Service Guide* for this procedure.

- **Build New Levels** — The build task generates the serviced level of an object and places the new object on a test BUILD disk.

- **Place the New Service into Production** — Once all service has been satisfactorily tested, it is placed into production by copying the new service to the production disks. Note that customized files that have been serviced may require additional customization.

7.2 Servicing TCP/IP for z/VM

7.2.1 Important Service Notes

Before you service TCP/IP for z/VM, you should review the information that follows and take appropriate action to ensure that your service environment is correct.

7.2.1.1 PPF Override Considerations

A *ppfname* of **4TCPIP30** is cited throughout these service instructions, which assumes the PPF supplied by IBM for TCP/IP for z/VM is in use. If you have created your own TCP/IP for z/VM PPF override file, use the *ppfname* of your override file (instead of 4TCPIP30) **throughout** this procedure, unless noted otherwise.

7.2.1.2 Language Environment Run-time Library Considerations

If you service TCP/IP for z/VM C components, the Language Environment run-time library must be available when you build serviced objects using the VMFBLD command. If the Language Environment run-time library does not reside on a system minidisk automatically accessed by VMSES/E (such as the MAINT 19E minidisk), you need to ensure the appropriate minidisk is available (perhaps through the use of a PPF override).

7.2.1.3 Installing RSU and COR Service — Where to Begin

```
┌─ Quick Index for Service Instructions ─────────────────────────────────────┐
│                                                                             │
│  Select the service instructions you should use, based on the type of service you are installing: │
│                                                                             │
│   • TCP/IP for z/VM RSU Service — Begin with 7.2.2, "Preventive (RSU) Service for TCP/IP for z/VM" │
│     on page  46                                                             │
│                                                                             │
│   • TCP/IP for z/VM COR Service — Begin with 7.2.3, "Corrective (COR) Service for TCP/IP for z/VM" │
│     on page  55                                                             │
│                                                                             │
└─────────────────────────────────────────────────────────────────────────────┘
```

7.2.2 Preventive (RSU) Service for TCP/IP for z/VM

Preventive service is available periodically on the Recommended Service Upgrade (RSU) tape. Each RSU is cumulative and contains selected, important PTFs. The service on the RSU is in pre-applied, pre-built format and includes serviced files, the objects that were rebuilt using these files, and an updated software inventory.

The RSU content allows for installing new service more quickly than an equivalent group of Corrective (COR) PTFs. However, the installation of an RSU requires that you reapply any reach-ahead service that is already installed (service you have applied to your system that is not on the RSU).

RSU tapes are in *install* format; thus, the VMFINS command is used to load an RSU tape.

7.2.2.1 Prepare to Receive Service

Electronic Service (Envelope File)

If you have received the RSU electronically or on CD-ROM, follow the appropriate instructions to retrieve and decompress the envelope file to your A-disk. Decompression is currently done by using the DETERSE module, which produces a decompressed file that has a file name format of RPTF*num* and a file type of SERVLINK. You will need to supply this file name for the VMFINS commands that follow.

---

1 Log on the TCP/IP for z/VM service user ID, **4TCPIP30**.

The PROFILE EXEC provided for this user ID (as part of the z/VM Version 4 Release 3.0 System DDR) contains ACCESS commands for the required VMSES/E minidisks — the VMSES/E code minidisk (MAINT 5E5, by default) and the VMSES/E Software Inventory minidisk (MAINT 51D, by default).

2 Issue the CMS QUERY DISK command to verify the VMSES/E code and Software Inventory minidisks are correctly linked and accessed.

query disk   Verify the MAINT 5E5 minidisk is accessed as file mode **B**, and is linked **R/O**.

Verify the MAINT 51D minidisk is accessed as file mode **D**, and is linked **R/W**.

Note: If another user has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the 51D disk in read-only (RR) mode, after which you need issue the appropriate LINK and ACCESS commands for the 51D minidisk. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

3 If necessary, establish the appropriate access to the VMSES/E minidisks.

    a Establish read access to the VMSES/E code minidisk.

        link MAINT 5e5 5e5 rr
        access 5e5 b

    b Establish write access to the Software Inventory minidisk.

        link maint 51d 51d mr
        access 51d d

4 Mount the TCP/IP for z/VM RSU tape on an appropriate device and ensure this device has been attached to the **4TCPIP30** user ID using virtual device number 181.

5 Receive the product documentation (4TCPIP30 MEMO) to the 51D minidisk, then identify the products and components for which service is included on the RSU tape. The MEMO documentation identifies the amount of storage necessary to receive the service present on the RSU. Use this information to ensure that your service disks or directories have adequate storage for this purpose.

    a If receiving the RSU from **tape**, issue:

vmfins install info (nomemo   **nomemo** will load but not print the memo.

    b If receiving the RSU from an **envelope** file, issue:

vmfins install info (nomemo env rptf*num*   **nomemo** will load but not print the memo.

6 Clear the alternate APPLY disk to ensure that a clean minidisk exists for receipt of the new service.

vmfmrdsk 4tcpip30 {tcpip | tcpipsfs} apply (setup

                                  Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

                                  This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

7 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview mrd

8 Invoke the VMFPSU command to obtain additional information about the service contained on the RSU and how it will affect your local modifications. This command creates an output file (*appid* **PSUPLAN**) that you should review. See the *z/VM: Service Guide* for an explanation of this file and its content.

vmfpsu 4tcpip30 {tcpip | tcpipsfs} Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

This command produces an output file which compares service present on the RSU to the service on your system. The file name is *appid* **PSUPLAN**, where *appid* is as specified in the PPF file.

7.2.2.2 Receive the Service

1 Receive the service on the RSU.

Because the RSU contains pre-applied, pre-built service in *install* format, the VMFINS command is used to load:

- new service to the DELTA disk,
- an updated apply service inventory to the APPLY disk, and
- pre-built objects to the appropriate test build disks.

a If receiving the RSU from **tape**, issue:

vmfins install ppf 4tcpip30 {tcpip | tcpipsfs} (nomemo nolink

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

nolink prevents VMFINS from linking required minidisks; it will only access these minidisks if they are not accessed.

b If receiving the RSU from an **envelope** file, issue:

vmfins install ppf 4tcpip30 {tcpip | tcpipsfs} (nomemo nolink env rptf*num* override no

> Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.
>
> *nolink* prevents VMFINS from linking required minidisks; it will only access these minidisks if they are not accessed.

2 Review the install message log ($VMFINS $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview install

7.2.2.3 Apply the Service

Because service on the RSU is pre-applied, this step reapplies applicable reach-ahead service (service you have applied to your system that is not on the RSU).

1 Reapply reach-ahead service.

vmfapply ppf 4tcpip30 {tcpip | tcpipsfs}

> Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.
>
> This command reapplies the reach-ahead service on your system. The version vector table (VVT) is updated with serviced part information and all necessary AUX files are generated on the alternate apply disk.

2 Review the apply message log ($VMFAPP $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview apply

3 Re-work and reapply local service, if it has been affected by RSU-provided service.

    a. The output file created by the VMFPSU command (invoked in step 8 on page 48) identifies local modifications that are affected by RSU service.

    b. For information on re-working local modifications, refer to Chapter 7 in the *z/VM: Service Guide* and follow the steps that are applicable to your local modification(s).

    Note that when using this information, the following substitutions may need to be made:

- **zvm** should be: **4tcpip30**
- *compname* should be: **tcpip** or **tcpipsfs**
- *appid* should be: **4tcpip30**
- *fm-local* should be the file mode of the 2C4 minidisk
- *fm-applyalt* should be the file mode of the 2A6 minidisk

Keep in mind that when you reach this step in the *z/VM: Service Guide*:

- "Rebuilding Objects"

you should return to using this program directory and continue with step 7.2.2.4, "Update the Build Status Table."

7.2.2.4 Update the Build Status Table

1 Update the Build Status Table for serviced parts.

vmfbld ppf 4tcpip30 {tcpip | tcpipsfs} (status   Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

        This command updates the Build Status Table to determine what objects (if any) remain to be built.

If a $PPF file has been serviced you will receive this prompt:

```
VMFBLD2185R The following source product parameter files have been
            serviced:
VMFBLD2185R 4TCPIP30 $PPF
VMFBLD2185R When source product parameter files are serviced, all
            product parameter files built from them must be recompiled
            using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
            parameter files built to your A-disk and exit VMFBLD so
            you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
            recompiled your product parameter files with VMFPPF.
```

0   Enter a zero (**0**) and complete the steps given here before you continue.

```
VMFBLD2188I Building 4TCPIP30 $PPF
            on 191 (A) from level $PFnnnnn
```

vmfppf 4tcpip30 *   **Note:** If you've created your own PPF override, use your PPF name (for this command *only*) instead of 4TCPIP30.

copyfile 4tcpip30 $ppf a = = d (olddate replace

erase 4tcpip30 $ppf a   **Note:** **Do not** use your own PPF name in place of 4TCPIP30 for these COPYFILE and ERASE commands.

vmfbld ppf 4tcpip30 {tcpip | tcpipsfs} (status

1   Re-issue VMFBLD to complete updates to the build status table.

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

When you receive the previously displayed prompt, enter a one (**1**) and continue to the next step.

2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

vmfview build

7.2.2.5 Build Serviced Objects

Because service on the RSU is pre-built, this step builds only objects which have been affected by any reach-ahead and local service that has been reapplied.

1 Ensure Language Environment run-time library support is available if you are applying service to functions that require this support.

2 Rebuild TCP/IP for z/VM serviced parts.

vmfbld ppf 4tcpip30 {tcpip | tcpipsfs} (serviced   Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

3 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview build   **Note:** The message that follows will be included in the build message log for each TCP/IP for z/VM (binary) system file that has been serviced. This is a normal occurrence, since system files are built by the TCP2PROD command when they are placed into production.

```
ST:VMFBLD1851I Cannot process TCPBLxxx EXCnnnnn because it is flagged to
ST:         be bypassed
```

7.2.2.6 Test the New Service

All new service should be thoroughly test before it is placed into production. A suggested method for doing this is to temporarily link and then access the appropriate *test* build minidisks ahead of their *production* counterparts. Representative LINK and ACCESS statements for this purpose are illustrated here:

```
LINK TCPMAINT 198 198 RR
LINK 4TCPIP30 491 491 RR
LINK 4TCPIP30 492 492 RR
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
...
ACCESS 198 fm198                    /* fm198 is 'D' (perhaps) */
ACCESS 491 fm491                    /* fm491 is 'E' (perhaps) */
ACCESS 492 fm492                    /* fm492 is 'F' (perhaps) */
ACCESS 591 fm591                    /* fm591 is 'G' (perhaps) */
ACCESS 592 fm592                    /* fm592 is 'H' (perhaps) */
```

- To facilitate the testing of new service that affects TCP/IP **server virtual machines**, a TCP/IP *server profile exit* (or the supplied sample *global profile exit*, TCPRUNXT SEXEC) could be used to establish a suitable environment, as part of server initialization (**SETUP**) processing. For information on these exits, see the chapter titled "General TCP/IP Server Configuration" in *z/VM: TCP/IP Level 430 Planning and Customization*.

  When TCP/IP server and administrative functions are tested, ensure that the TCPMAINT 198, 4TCPIP30 491, and TCPMAINT 591 minidisks are accessed (in addition to the 4TCPIP30 492 and TCPMAINT 592 minidisks) by the TCP/IP service virtual machines and administrative user IDs involved in this activity.

- To facilitate the testing of new service that affects TCP/IP **client functions** (for example, the NETSTAT or FTP command), only the 4TCPIP30 492 and TCPMAINT 592 minidisks need to be accessed (with respect to TCP/IP-specific minidisk requirements).

When new service is tested, consult any applicable documentation (for example, that provided with the APARs which comprise the service) to account for changes specific to new or changed function. Your testing may also require TCP/IP services affected by service to be shutdown and restarted, possibly more than once.

Note - Shutting Down TCP/IP Services

Before you shutdown any TCP/IP servers, ensure that applicable conditions or guidelines for your environment have been followed.

For information on shutting down TCP/IP servers, see the section that discusses "Starting and Stopping TCP/IP Servers" in the chapter titled "General TCP/IP Server Configuration," of *z/VM: TCP/IP Level 430 Planning and Customization*.

7.2.2.7 Place the Service into Production

Once the new service has been thoroughly tested, it needs to be placed into production; complete the appropriate TCP/IP service instructions that remain to accomplish this.

RSU Service Instructions — Where To Next...

See 7.2.4, "Place the New TCP/IP for z/VM Service Into Production" on page 64 for instructions for completing the installation of RSU service.

7.2.3 Corrective (COR) Service for TCP/IP for z/VM

Corrective service for TCP/IP for z/VM is provided in COR format via tape or electronic envelope. It is installed using the VMSES/E VMFREC, VMFAPPLY, and VMFBLD commands.

┌─ **Electronic Service (Envelope File)** ──────────────────────────────────────┐

If you have received service electronically or on CD-ROM, follow the appropriate instructions to retrieve and decompress the envelope files to your A-disk. Decompression is currently done by using the DETERSE module, which will produce decompressed files that have the following file name formats:

- VLST*num*, for the documentation envelope
- VPTF*num*, for the service envelope.

The file type for both of these files is (and must be) SERVLINK. You will need to supply these file names with the VMFREC commands cited in these instructions.

└──┘

┌─ **TCP/IP for z/VM System File Considerations** ──────────────────────────────┐

If the corrective service you intend to apply affects TCP/IP for z/VM (binary) system files, you may first need to *commit* the current level of service for these files **before** the relevant new service can be received and applied. For more information, see Appendix C, "PTF Commit Procedure for TCP/IP for z/VM System Files" on page 89.

└──┘

7.2.3.1 Prepare to Receive Service

1 Log on the TCP/IP for z/VM service user ID, **4TCPIP30**.

The PROFILE EXEC provided for this user ID (as part of the z/VM Version 4 Release 3.0 System DDR) contains ACCESS commands for the required VMSES/E minidisks — the VMSES/E code minidisk (MAINT 5E5, by default) and the VMSES/E Software Inventory minidisk (MAINT 51D, by default).

2 Issue the CMS QUERY DISK command to verify the VMSES/E code and Software Inventory minidisks are correctly linked and accessed.

query disk

Verify the MAINT 5E5 minidisk is accessed as file mode **B**, and is linked **R/O**.

Verify the MAINT 51D minidisk is accessed as file mode **D**, and is linked **R/W**.

Note: If another user has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the 51D disk in read-only (RR) mode, after which you

need issue the appropriate LINK and ACCESS commands for the 51D
minidisk. Do not continue with these procedures until a R/W link is
established to the 51D minidisk.

3 If necessary, establish the appropriate access to the VMSES/E minidisks.

    a Establish read access to the VMSES/E code minidisk.

       link MAINT 5e5 5e5 rr
       access 5e5 b

    b Establish write access to the Software Inventory minidisk.

       link maint 51d 51d mr
       access 51d d

4 Have the TCP/IP for z/VM corrective (COR) service tape mounted and
attached as 181 to **4TCPIP30**.

5 Mount the TCP/IP for z/VM corrective (COR) tape on an appropriate device
and ensure this device has been attached to the **4TCPIP30** user ID using
virtual device number 181.

6 Establish the correct minidisk access order.

**vmfsetup 4tcpip30 {tcpip	tcpipsfs}**	Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

7 Receive the service documentation. The VMFREC command (with its INFO
option) loads the service documentation and displays a list of all products for
which service is present on the tape.

    a If receiving service from **tape**, issue:

vmfrec info	This command loads the service memo to the 4TCPIP30 191 minidisk.

b If receiving service from an **envelope** file, issue:

vmfrec info (env vlst*num*   This command loads the service memo to the 4TCPIP30 191 minidisk.

8 Review the receive message log ($VMFREC $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview receive   Also, note the products and components for which service has been supplied on the tape. To do this, use the PF5 key to show all "status" messages, which will identify these products and components.

9 Clear the alternate APPLY disk to ensure that a clean minidisk exists for receipt of the new service. for new service.

vmfmrdsk 4tcpip30 {tcpip | tcpipsfs} apply   Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

This command copies the alternate APPLY disk to the production APPLY disk and then clears the alternate APPLY disk.

10 Review the merge message log ($VMFMRD $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview mrd

7.2.3.2 Receive the Service

1 Receive the service.

a If receiving service from **tape**, issue:

vmfrec ppf 4tcpip30 {tcpip | tcpipsfs}

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

This command receives service from the service tape. All new service is loaded to the DELTA disk.

b If receiving service from an **envelope** file, issue:

vmfrec ppf 4tcpip30 {tcpip | tcpipsfs} (env vptf*num*

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

This command receives service from the service envelope. All new service is loaded to the DELTA disk.

2 Review the receive message log ($VMFREC $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview receive

7.2.3.3 Apply the Service

1 Apply the new service.

vmfapply ppf 4tcpip30 {tcpip | tcpipsfs}

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

This command applies the just-recieved service. The version vector table (VVT) is updated with serviced part information and all necessary AUX files are generated on the alternate apply disk.

2 Review the apply message log ($VMFAPP $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview apply

┌─ **Note - Local Modifications** ─────────────────────────────────────

If you receive message VMFAPP2120W, you need to reapply any local modifications before building an updated level of TCP/IP for z/VM. For information on re-working local modifications, refer to Chapter 7 in the *z/VM: Service Guide* and follow the steps that are applicable to your local modification(s).

Note that when using this information, the following substitutions may need to be made:

- **zvm** should be: **4tcpip30**
- *compname* should be: **tcpip** or **tcpipsfs**
- *appid* should be: **4tcpip30**
- *fm-local* should be the file mode of the 2C4 minidisk
- *fm-applyalt* should be the file mode of the 2A6 minidisk

Keep in mind that when you reach this step in the *z/VM: Service Guide*:

- "Rebuilding Objects"

you should return to using this program directory and continue with step 7.2.3.4, "Update the Build Status Table" on page 60.

└──

7.2.3.4 Update the Build Status Table

1 Update the Build Status Table for serviced parts.

vmfbld ppf 4tcpip30 {tcpip | tcpipsfs} (status

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

This command updates the Build Status Table to identify objects that need to be built as a result of applying new service.

If a $PPF file has been serviced you will receive this prompt:

```
VMFBLD2185R The following source product parameter files have been
            serviced:
VMFBLD2185R 4TCPIP30 $PPF
VMFBLD2185R When source product parameter files are serviced, all
            product parameter files built from them must be recompiled
            using VMFPPF before VMFBLD can be run.
VMFBLD2185R Enter zero (0) to have the serviced source product
            parameter files built to your A-disk and exit VMFBLD so
            you can recompile your product parameter files with VMFPPF.
VMFBLD2185R Enter one (1) to continue only if you have already
            recompiled your product parameter files with VMFPPF.
```

0   Enter a zero (**0**) and complete the steps given here before you continue.

```
VMFBLD2188I Building 4TCPIP30 $PPF
            on 191 (A) from level $PFnnnnn
```

vmfppf 4tcpip30 *   **Note:** If you've created your own PPF override, use your PPF name (for this command *only*) instead of 4TCPIP30.

copyfile 4tcpip30 $ppf a = = d (olddate replace

erase 4tcpip30 $ppf a   **Note:** **Do not** use your own PPF name in place of 4TCPIP30 for these COPYFILE and ERASE commands.

vmfbld ppf 4tcpip30 {tcpip | tcpipsfs} (status setup

1   Re-issue VMFBLD to complete updates to the build status table.

                                             Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

                                             When you receive the previously displayed prompt, enter a one (**1**) and continue to the next step.

2 Use VMFVIEW to review the build status messages, and see what objects need to be built.

vmfview build

7.2.3.5 Build Serviced Objects

1 Ensure Language Environment run-time library support is available if you are applying service to functions that require this support.

2 Rebuild TCP/IP for z/VM serviced parts.

vmfbld ppf 4tcpip30 {tcpip | tcpipsfs} (serviced　　　Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

3 Review the build message log ($VMFBLD $MSGLOG). If necessary, correct any problems before you proceed with the next step. For information about handling specific error messages, see *z/VM: System Messages and Codes - Other Components* or use on-line HELP.

vmfview build　　　**Note:** The message that follows will be included in the build message log for each TCP/IP for z/VM (binary) system file that has been serviced. This is a normal occurrence, since system files are built by the TCP2PROD command when they are placed into production.

```
ST:VMFBLD1851I Cannot process TCPBLxxx EXCnnnnn because it is flagged to
ST:            be bypassed
```

7.2.3.6 Test the New Service

All new service should be thoroughly test before it is placed into production. A suggested method for doing this is to temporarily link and then access the appropriate *test* build minidisks ahead of their *production* counterparts. Representative LINK and ACCESS statements for this purpose are illustrated here:

```
LINK TCPMAINT 198 198 RR
LINK 4TCPIP30 491 491 RR
LINK 4TCPIP30 492 492 RR
LINK TCPMAINT 591 591 RR
LINK TCPMAINT 592 592 RR
...
ACCESS 198 fm198                    /* fm198 is 'D' (perhaps) */
ACCESS 491 fm491                    /* fm491 is 'E' (perhaps) */
ACCESS 492 fm492                    /* fm492 is 'F' (perhaps) */
ACCESS 591 fm591                    /* fm591 is 'G' (perhaps) */
ACCESS 592 fm592                    /* fm592 is 'H' (perhaps) */
```

- To facilitate the testing of new service that affects TCP/IP **server virtual machines**, a TCP/IP *server profile exit* (or the supplied sample *global profile exit*, TCPRUNXT SEXEC) could be used to establish a suitable environment, as part of server initialization (**SETUP**) processing. For information on these exits, see the chapter titled "General TCP/IP Server Configuration" in *z/VM: TCP/IP Level 430 Planning and Customization*.

  When TCP/IP server and administrative functions are tested, ensure that the TCPMAINT 198, 4TCPIP30 491, and TCPMAINT 591 minidisks are accessed (in addition to the 4TCPIP30 492 and TCPMAINT 592 minidisks) by the TCP/IP service virtual machines and administrative user IDs involved in this activity.

- To facilitate the testing of new service that affects TCP/IP **client functions** (for example, the NETSTAT or FTP command), only the 4TCPIP30 492 and TCPMAINT 592 minidisks need to be accessed (with respect to TCP/IP-specific minidisk requirements).

When new service is tested, consult any applicable documentation (for example, that provided with the APARs which comprise the service) to account for changes specific to new or changed function. Your testing may also require TCP/IP services affected by service to be shutdown and restarted, possibly more than once.

Note - Shutting Down TCP/IP Services

Before you shutdown any TCP/IP servers, ensure that applicable conditions or guidelines for your environment have been followed.

For information on shutting down TCP/IP servers, see the section that discusses "Starting and Stopping TCP/IP Servers" in the chapter titled "General TCP/IP Server Configuration," of *z/VM: TCP/IP Level 430 Planning and Customization*.

7.2.4 Place the New TCP/IP for z/VM Service Into Production

7.2.4.1 4TCPIP30 CATALOG File Modification Notes

Before you continue with the steps in this section, it is recommended that you verify the correctness of any 4TCPIP30 CATALOG file modifications that may have been made for your environment; this file is used by the TCP2PROD command to copy files to TCP/IP for z/VM minidisks. See Appendix A, "TCP/IP Installation and Service Utilities" on page 73 for detailed information about the TCP2PROD command and TCP/IP for z/VM catalog files.

Any new files that can be processed will be identified within a serviced 4TCPIP30 CATALOG file.

The various definition sections of the 4TCPIP30 CATALOG file, and the files associated with each section, are briefly described here:

Section	Description
TCPRUN	Non-customizable TCP/IP for z/VM run-time files. For reference, files that can be processed using the TCPRUN section are listed in Figure 13 on page 38 of 6.2.2.4, "TCP/IP for z/VM Product and Sample Configuration Files" on page 37.
TCPCONFIG	Customizable sample files. For reference, files that can be processed using the TCPCONFIG section are listed in Figure 14 on page 40.
TCPSYSFILE	Non-customizable TCP/IP (binary) system files. For reference, files that can be processed using the TCPSYSFILE section are listed in Figure 15 on page 42.

7.2.4.2 Copy Serviced TCP/IP for z/VM Run-time Files Into Production

Once the new service has been thoroughly tested, it needs to be placed into production (that is, copied to TCP/IP for z/VM production build minidisks).

1 Shutdown TCP/IP services.

> ┌─ **Note - Shutting Down TCP/IP Services** ─────────────────────
>
> Before you shutdown any TCP/IP servers, ensure any applicable conditions or guidelines for your environment have been followed.

For information on shutting down TCP/IP servers, see the section that discusses "Starting and Stopping TCP/IP Servers" in the chapter titled "General TCP/IP Server Configuration," of *z/VM: TCP/IP Level 430 Planning and Customization*.

2 Log on the TCP/IP for z/VM service user ID, **4TCPIP30**.

The PROFILE EXEC provided for this user ID (as part of the z/VM Version 4 Release 3.0 System DDR) contains ACCESS commands for the required VMSES/E minidisks — the VMSES/E code minidisk (MAINT 5E5, by default) and the VMSES/E Software Inventory minidisk (MAINT 51D, by default).

3 Issue the CMS QUERY DISK command to verify the VMSES/E code and Software Inventory minidisks are correctly linked and accessed.

query disk   Verify the MAINT 5E5 minidisk is accessed as file mode **B**, and is linked **R/O**.

Verify the MAINT 51D minidisk is accessed as file mode **D**, and is linked **R/W**.

Note: If another user has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the 51D disk in read-only (RR) mode, after which you need issue the appropriate LINK and ACCESS commands for the 51D minidisk. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

4 If necessary, establish the appropriate access to the VMSES/E minidisks.

a Establish read access to the VMSES/E code minidisk.

link MAINT 5e5 5e5 rr
access 5e5 b

b Establish write access to the Software Inventory minidisk.

link maint 51d 51d mr
access 51d d

5 Access the 4TCPIP30 491 minidisk.

access 491 i   The 491 minidisk is where the TCP2PROD EXEC and 4TCPIP30 CATALOG files reside.

6 Review the 4TCPIP30 CATALOG file to verify its correctness, as recommended in 7.2.4.1, "4TCPIP30 CATALOG File Modification Notes" on page 64. Ensure that any changes you may have made to this file remain in effect, and that any local service used to customize this file has been properly re-worked and applied.

7 (*Optional*) Establish write links to any TCP/IP for z/VM production or server minidisks which are not yet linked in this mode.

LINK statements for the various TCP/IP for z/VM minidisks are present in the 4TCPIP30 directory entry (supplied as part of the installed z/VM Version 4 Release 3.0 System DDR).

If you have changed the default installation user ID or use different minidisk device numbers in your environment, you may need to manually link the necessary TCP/IP production and server minidisks. See Figure 13 on page 38 and Figure 14 on page 40 for device link information. If you created a PPF override that has changed any of these device numbers, use your values.

link *tcpipid vdev1 vdev2* **mr**

> **Note:** If another user has the *vdev1* minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the *vdev1* disk in read-only (RR) mode, after which you need to re-issue the above LINK command. Do not continue with these procedures until a R/W link is established to the *vdev1* minidisk.

8 Copy serviced TCP/IP for z/VM files into production using the TCP2PROD command. The command cited below process files that are identified in the TCPRUN section of the 4TCPIP30 CATALOG file. See Appendix A, "TCP/IP Installation and Service Utilities" on page 73 for information about this command and TCP/IP for z/VM catalog files.

> ┌─ **Verifying Your Environment** ─────────────────────────────
>
> When you perform this step, it is suggested that you first invoke TCP2PROD as illustrated, but with the **TEST** option also specified. This will verify that all resources can be accessed and that the appropriate files will be processed.
>
> With the **TEST** option in effect, **no files are copied into production**.
>
> Resolve any reported problems, then invoke TCP2PROD (without the TEST option) as illustrated.

tcp2prod 4tcpip30 {tcpip | tcpipsfs} 4tcpip30 tcprun (setup

> Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

9 Review the TCP2PROD message log (TCP2PROD $MSGLOG). If necessary, correct any problems before you proceed with the next step.

7.2.4.3 Copy New TCP/IP for z/VM Configuration Files Into Production

Use the TCP2PROD command, as described in this section, to place any new IBM-supplied sample configuration files into production.

Note

When you perform this step, only *new* configuration files are copied into production. Existing (that is, customized) configuration files are *not* replaced.

Note: For step 2 below, it's assumed that the TCPCONFIG section of the 4TCPIP30 CATALOG has been verified, as recommended in 7.2.4.1, "4TCPIP30 CATALOG File Modification Notes" on page 64. If this is not the case, you should make any necessary changes to this section of the 4TCPIP30 CATALOG file before you continue with the steps that follow.

1 If necessary, establish the appropriate environment, as described by steps 1 through 5, in 7.2.4.2, "Copy Serviced TCP/IP for z/VM Run-time Files Into Production" on page 64.

2 Copy serviced TCP/IP for z/VM configuration files into production using the TCP2PROD command. For reference, files that can be processed using the TCPCONFIG section are listed in Figure 14 on page 40.

Verifying Your Environment

When you perform this step, it is suggested that you first invoke TCP2PROD as illustrated, but with the **TEST** option also specified. This will verify that all resources can be accessed and that the appropriate files will be processed.

With the **TEST** option in effect, **no files are copied into production**.

Resolve any reported problems, then invoke TCP2PROD (without the TEST option) as illustrated.

---

tcp2prod 4tcpip30 {tcpip | tcpipsfs} 4tcpip30 tcpconfig (setup

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

3 Review the TCP2PROD message log (TCP2PROD $MSGLOG). If necessary, correct any problems before you proceed with the next step.

vmfview tcp2prod

RSU Service Instructions — Where To Next...

If you are installing **RSU** service, skip the section that follows and continue with the instructions provided in 7.2.4.5, "Configure TCP/IP for z/VM" on page 70.

7.2.4.4 Build and Copy Serviced TCP/IP for z/VM System Files Into Production

Use the TCP2PROD command, as described below, to build and then place any serviced TCP/IP for z/VM system files into production.

> **Corrective Service-specific Processing**
>
> The steps described in this section are necessary **only** if Corrective (**COR**) service has been applied that has affected one or more TCP/IP (binary) system files (as listed in Figure 15 on page 42).

Note: For step 2 that follows, it's assumed that the TCPSYSFILE section of the 4TCPIP30 CATALOG has been verified, as recommended in 7.2.4.1, "4TCPIP30 CATALOG File Modification Notes" on page 64. If this is not the case, you should make any necessary changes to this section of the 4TCPIP30 CATALOG file before you continue with the steps that follow.

1 If necessary, establish the appropriate environment, as described by steps 1 through 5, in 7.2.4.2, "Copy Serviced TCP/IP for z/VM Run-time Files Into Production" on page 64.

2 Copy serviced TCP/IP for z/VM system files into production using the TCP2PROD command.

> **Verifying Your Environment**
>
> When you perform this step, it is suggested that you first invoke TCP2PROD as illustrated, but with the **TEST** option also specified. This will verify that all resources can be accessed and that the appropriate files will be processed.
>
> With the **TEST** option in effect, **no files are copied into production**.
>
> Note that TEST mode processing for TCP/IP for z/VM system files is somewhat limited (as compared with conventional run-time and configuration files) due to the processing requirements and characteristics of these files.
>
> Resolve any reported problems, then invoke TCP2PROD (without the TEST option) as illustrated.

tcp2prod 4tcpip30 {tcpip | tcpipsfs} 4tcpip30 tcpsysfile (setup

Use **tcpip** if the TCP/IP for z/VM default minidisk environment has been maintained; use **tcpipsfs** if the service minidisks were moved to Shared File System directories.

Note: For each *serviced* system file that is processed, the prompt shown below is issued before a production-level system file is created.

Enter a one (**1**) in response to this prompt to allow for creation of an updated system file.

```
DTCPRD3004I Accessing target resource...
DTCPRD3097I Accessing resource vaddr (owner_id vaddr) with file mode fm2
DTCPRD3062I Decompacting file filename FLBnnnnn fm1 to file mode A...
DTCPRD3055I About to RESERVE target minidisk owner_id vaddr for system file:
            filename DATABIN fm2
            This will erase all files on this disk. Do you want to continue?
            Enter 0 (No), 1 (Yes), 2 (Exit)
1
DTCPRD3060I Installing file filename DATABIN fm2 on minidisk owner_id vaddr
DTCPRD3068I Updating build status information for file: filename FLSBIN
DTCPRD3068I Updating build status information for file: TCPBLxxx (BLDLIST)
```

3 Review the TCP2PROD message log (TCP2PROD $MSGLOG). If necessary, correct any problems before you proceed with the next step.

vmfview tcp2prod

7.2.4.5 Configure TCP/IP for z/VM

If any of the configuration files you have previously customized have been affected by service, you should review the updated *sample* equivalents of the those files, and determine if any changes are required to your production files.

See *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019) for detailed information about the content and use of these files, and how to configure specific TCP/IP servers for your environment. To accommodate service-related changes, you may at times need to consult APAR-specific documentation.

7.2.4.6 Re-Initialize TCP/IP Services

Once you have completed any necessary configuration changes, the appropriate
TCP/IP servers must be initialized. For more information, see the section that
discusses "Starting and Stopping TCP/IP Servers" in the chapter titled "General
TCP/IP Server Configuration," of   *z/VM: TCP/IP Level 430 Planning and
Customization*.

7.2.4.7 Copy Serviced TCP/IP Client Code to the z/VM Product Code Disk (Optional)

If you previously copied TCP/IP for z/VM client code to the z/VM product code disk,
you should replace the appropriate files with their serviced counterparts. See
Appendix I, "Copying TCP/IP for z/VM Client Code to the Y-Disk" on page 113 for
additional information and instructions concerning this process.

7.2.4.8 Verify the RSU Service Level (Optional)

The procedures in this section describe how to determine the current TCP/IP for
z/VM RSU service level by using either of these commands:

*   TCP/IP **NETSTAT LEVEL**
*   VMSES/E **VMFSIM QUERY**

The service contained on each RSU constitutes a new service level; this service
level is updated in the system inventory when RSU service is installed.

Note: Corrective (COR) service does not affect the TCP/IP for z/VM RSU service
level. However, you should record and use this service level when corrective
(COR) service is ordered.

To use the TCP/IP **NETSTAT LEVEL** command to obtain the TCP/IP for z/VM RSU
service level, the TCPIP server (or, the TCP/IP *stack*) must be operating and the
NETSTAT command must be available.

access 592 *fm*   where *fm* is an available file mode.
netstat level

```
VM TCP/IP Netstat Level 430
IBM 2064; z/VM Version 4 Release 3.0, service level 0201 (nn-BIT), VM TCP/IP Level
430; RSU 0201
```

The last part of the returned output indicates the TCP/IP RSU service level; for this example, this is: **0201**

Note that any user ID that has the TCPMAINT 592 minidisk accessed can use the NETSTAT LEVEL command:

To use the **VMFSIM QUERY** command, the VMSES/E code (MAINT 5E5) and Software Inventory (MAINT 51D) minidisks must be accessed. Thus, this command is more readily issued while logged on the TCP/IP installation user ID, **4TCPIP30**.

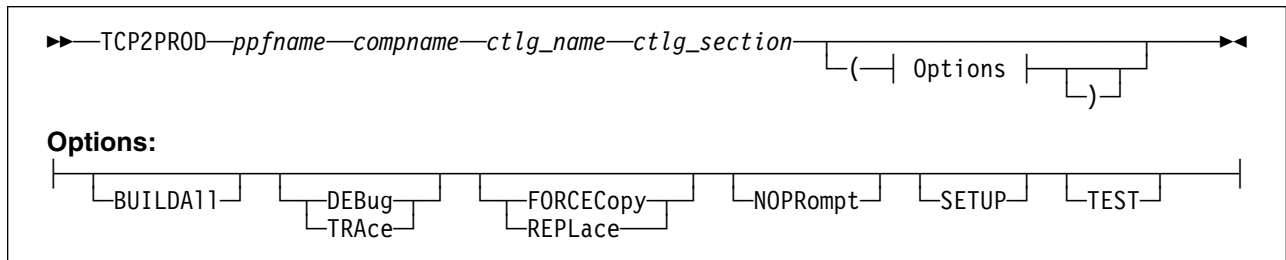**vmfsim query vm sysrecs tdata :ppf 4tcpip30 :stat**

```
VMFSIP2408I RESULTS FOR
           TDATA :PPF 4TCPIP30 :STAT
:PPF 4TCPIP30 TCPIP
  :STAT RECEIVED.04/01/02.01:23:45.4TCPIP30.RSU-0201
```

The last part of the status line indicates the TCP/IP RSU service level; for this example, this is: **0201**

You have finished servicing TCP/IP for z/VM.

Appendix A. TCP/IP Installation and Service Utilities

A.1 TCP2PROD

```
►►──TCP2PROD──ppfname──compname──ctlg_name──ctlg_section──────────────────────►◄
                                                      └─(─┤ Options ├──────┘
                                                                  └─)─┘
```

Options:

```
├──────────────┬────────┬─────────────┬──────────┬────────┬───────┤
    └─BUILDAll─┘  ┌─DEBug─┐  ┌─FORCECopy─┐  └─NOPRompt─┘  └─SETUP─┘  └─TEST─┘
                  └─TRAce─┘  └─REPLace──┘
```

A.1.1 Purpose

Use the TCP2PROD command to help you place TCP/IP for z/VM files into production. The TCP2PROD command uses a TCP/IP for z/VM CATALOG file to determine which product files are to be placed into production, as well as which minidisks and SFS directories are to be used for this process.

When TCP/IP for z/VM (binary) system files are processed, TCP2PROD also serves as a VMSES/E *part handler*, in that it manages build status information (maintained in the 4TCPIP30 SRVBLDS file) for these types of files.

Note: The TCP2PROD command is intended for use by the 4TCPIP30 user ID, and should only be used when you install or service TCP/IP for z/VM.

A.1.2 Operands

ppfname   is the name of the usable form product parameter file used for installing and maintaining TCP/IP for z/VM; the file type must be **PPF**.

compname   is the name of the component as specified on the :COMPNAME. tag in the product parameter file; *compname* is a 1- to 16-character alphanumeric identifier.

The PPF Variable Declarations (:DCL.) section defined for *compname* determines from which source minidisks and SFS directories product files are copied; likewise for the target minidisks to which these files are copied.

ctlg_name   is the name of the product catalog file to be processed. The file type must be CATALOG.

ctlg_section   is the definition section of the catalog file to be processed. The value
specified by *ctlg_section* is used as a "root" for the *begin* and *end*
tags that define each section of grouped entries (records) within a
catalog file.

A.1.3 Options

BUILDAll

    causes TCP2PROD to rebuild (binary) system files that correspond to entries
defined within a *system file* section of the TCP/IP for z/VM CATALOG file,
regardless of the VMSES/E build status for these objects. This option is has
no effect when other types of catalog entries are processed.

DEBUG
TRACE

    causes additional messages to be issued to provide information for debugging
purposes. The TRACE option is synonymous with the DEBUG option.

FORCECopy
REPLace

    causes files identified within a CATALOG *configuration file* definition section to
be copied to their configured name and type, regardless of whether such a
configuration file already exists. The REPLACE option is synonymous with the
FORCECOPY option.

NOPRompt

    prevents TCP2PROD from issuing prompts through the course of its
processing. When this option is used, TCP2PROD assumes an *affirmative
response* (**1**) for any prompts that are bypassed. In addition, appropriate
informational messages are issued in place of the usual prompts.

SETUP

    causes a VMSES/E VMFSETUP command to be issued as part of TCP2PROD
processing, so that the correct operational environment is established. The
ppfname and *compname* supplied when TCP2PROD is invoked are used as
operands for the VMFSETUP command.

TEST

    causes processing to be performed so that no files are placed into production
via the VMSES/E VMFCOPY command (for conventional files), or through
TCP2PROD-inherent processing (used for system files). This option allows you
to verify that minidisks and SFS directories can be accessed as required, and
that the appropriate catalog file entries will be processed. Additional messages
are issued in some cases to clarify what processing would be performed if the
TEST option was not used.

    Note: TEST mode processing for TCP/IP for z/VM system files is somewhat
limited (as compared with that for conventional run-time and configuration files)
due to the processing requirements and characteristics of these files.

A.1.4 Usage Notes

- **TCP2PROD does not issue LINK commands** for minidisks as part of its processing.

  If necessary, the appropriate LINK statements should be added to the 4TCPIP30 directory entry. Alternatively, the required LINK commands can be added to the PROFILE EXEC of the 4TCPIP30 user ID.

- The first CATALOG file found in the CMS search order that matches the name specified by *ctlg_name* is used.

- CATALOG file entries which are found to be unusable are bypassed. Warning messages are issued when entries are bypassed for this reason.

- A minidisk or directory must be accessed at file mode A with Read/Write (R/W) status, for use as temporary work space and for message logging.

- By default, a file identified within a *configuration file* definition section of a CATALOG file is copied to its configured name and type only if the configured file does not already exist.

  A catalog section is considered to be a *configuration file* definition section if its corresponding *begin* tag contains the string, **CONFIG**. For more information about CATALOG file structure and content, see A.2, "TCP/IP for z/VM CATALOG Files" on page 77.

- TCP2PROD processes system files by first decompressing service-level files to a temporary work space (file mode A) and then creating an appropriate binary image file under a unique CMS file name (for which a corresponding CMS minidisk has been reserved). At the same time, TCP2PROD serves as a VMSES/E *part handler*, in that it manages build status information (maintained in the 4TCPIP30 SRVBLDS file) for these types of files. By design, only files that are identified within a *system file* definition section of a CATALOG file are processed in this manner.

  A catalog section is considered to be a *system file* definition section if its corresponding *begin* tag contains the string, **SYSFILE**. For more information about CATALOG file structure and content, see A.2, "TCP/IP for z/VM CATALOG Files" on page 77.

- If TCP2PROD is invoked with *ppfname* specified as a question mark (**?**), the command syntax is displayed.

A.1.5 The TCP2PROD $MSGLOG File

Informational, warning and error messages issued to the console by TCP2PROD are maintained in a message log, TCP2PROD $MSGLOG. This log file is written to the minidisk or directory accessed at file mode A, and can be viewed using the VMSES/E **VMFVIEW** command.

The TCP2PROD $MSGLOG is cumulative, with the most recent entries appended
at the **top** of the file. Separator headers that include date and time stamps are
inserted in the log with each TCP2PROD invocation so that newer log entries can
be distinguished from older ones.

No messages are logged until an initial validation of TCP2PROD operands is
complete.

A.1.6 Return Codes

The return codes that can be produced by TCP2PROD are listed and explained
here:

0

Explanation: Successful execution; no processing errors were encountered.

1

Explanation: Incorrect invocation. TCP2PROD was invoked with an incorrect number of
operands. A message that identifies the missing operand is displayed, in addition to the
command syntax.

2

Explanation: Internal error. If return code 2 is returned, processing status is unknown.
Contact the TCP/IP for z/VM support group for problem determination and assistance in
addressing this type of error.

4

Explanation: Errors encountered, with warnings issued. The errors encountered may have
caused processing to complete with only partial success. Review the TCP2PROD
$MSGLOG for warning messages which identify any problems that were encountered.

8

Explanation: Errors encountered; processing has not completed successfully. Review the
TCP2PROD $MSGLOG for messages regarding the problems encountered.

A.2 TCP/IP for z/VM CATALOG Files

A.2.1 Purpose

THE TCP/IP for z/VM CATALOG file is used by TCP2PROD to determine which TCP/IP for z/VM product files are to be placed into production, as well as which minidisks and SFS directories are to be used for this process.

CATALOG files provided for use with TCP/IP for z/VM are listed in Figure 16.

Figure 16. TCP/IP for z/VM CATALOG Files	
Catalog File Name / Type	**Associated Files**
4TCPIP30 CATALOG	TCP/IP for z/VM files

Distinct sections within the CATALOG file are used to identify *product* files, such as the TCPIP MODULE, and *sample* files which need to be customized, such as the TCPIP SDATA file. Each definition section is delimited by a pair of *begin* and *end* tags. For example, the definition section for TCP/IP for z/VM sample configuration files is defined using the `:TCPCONFIG.` and `:ETCPCONFIG.` tags.

The entries within each section provide information sufficient for TCP2PROD to process a specific group of TCP/IP files.

Unless marked otherwise, all entries within a given section are processed and the corresponding files are placed into production via the VMSES/E VMFCOPY command (for conventional files) or through additional processing inherent to the TCP2PROD command (for system files).

A.2.2 CATALOG Record Syntax

The syntax for TCP/IP for z/VM CATALOG file entries is indicated below.

```
►►─────┬───┬───────src_dclvar──prd_dclvar──────────────────────────────►

    └─ - ─┘

►────┬──src_fname──┬──┬──src_ftype──┬────┬──prd_fname──┬──────────────►
     │      *      │  │      *      │    │      =      │
     └─sysfile_name──bldlist_name──(1)┘

►───┬──prd_ftype──┬──────────────────────────────────────────►◄
    │     =   (2) │      └─comment─┘
    └─────────────┘
```

Notes:

[1] The *sysfile_name* and *bldlist_name* operands are only valid within a *system file* definition section.

[2] Not valid within a *system file* definition section.
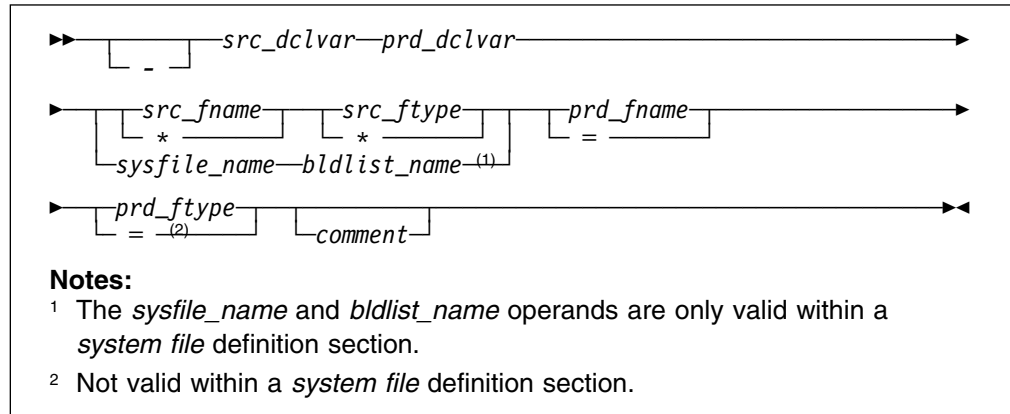
A.2.3 Operands

-   is the entry bypass character, a hyphen (-). Use this bypass character at the beginning of a file entry to indicate it should *not* be processed by TCP2PROD.

         Note: It's recommended that all file entries remain present within any supplied TCP/IP for z/VM CATALOG files. If a TCP/IP for z/VM file is not used in your environment, its corresponding entry should be bypassed as described above. Doing so will also help readily distinguish comment lines from entries that correspond to TCP/IP for z/VM product files.

src_dclvar   is a PPF :DCL. variable name for the minidisk or SFS directory where the source file resides.

prd_dclvar   is a PPF :DCL. variable name for the minidisk or SFS directory where the production or configured file is to reside.

src_fname   is the source file name. An asterisk (*) can be used to indicate that all files of the type specified by *src_ftype* are to be processed. When this wildcard (*) is used, the production or configured file name remains unchanged from the source file name.

src_ftype   is the source file type. An asterisk (*) can be used to indicate that all files of the name specified by *src_fname* are to be processed. When this wildcard (*) is used, the production or configured file type remains unchanged from the source file type.

prd_fname   is the configured file name. If the source file name is specified using an asterisk (*), *prd_fname* must be specified to maintain the correct entry format. In such cases, an equal sign (=) should be used; a value other than an equal sign is processed as if an equal sign had been specified.

prd_ftype   is the production or configured file type. If the source file type is specified using an asterisk (*), *prd_ftype* must be specified to maintain the correct entry format. In such cases, an equal sign (=) should be used; a value other than an equal sign is processed as if an equal sign had been specified.

Note: An equal sign cannot be used to specify a production file type within a *system file* definition section; a "conventional" CMS file type must be specified within this type of definition section.

sysfile_name
is the file name of a (binary) source system file. This operand is only valid within a *system file* definition section.

bldlist_name
is the name of the VMSES/E build list with which a preceding *sysfile_name* file is associated. For TCP/IP for z/VM system files, a one-to-one correspondence exists for a given build list and a given system file. This operand is only valid within a *system file* definition section.

comment   commentary text that is ignored by TCP2PROD during processing.

Notes:

1. All operands must be separated by at least one space.

2. Comment lines may be included in CATALOG files. Comment lines must begin with an asterisk (*); these lines are ignored during TCP2PROD processing.

A.2.4 Usage Notes

- TCP/IP for z/VM CATALOG file changes that are required for your environment should be made via a VMSES/E local modification, to allow for the reporting of service-related changes during VMSES/E processing. See Appendix B, "Modifying TCP/IP for z/VM CATALOG Files" on page 85 for more information about how to change a TCP/IP for z/VM CATALOG file in this manner.

- The source and target minidisk/directory variable names used within this file correspond to those used within the TCP/IP for z/VM ($)PPF file (or an override variation of that file). If any changes are made to the Variable Declarations (:DCL.) section of the TCP/IP for z/VM PPF file via a PPF override, you may need to incorporate similar changes within TCP/IP for z/VM CATALOG files

(through separate VMSES/E local modifications) to allow for the correct resolution of PPF :DCL. variable names.

A.2.5 Additional Information

The various definition sections of the 4TCPIP30 CATALOG file, and the files associated with each section, are briefly described here:

Section   **Description**

TCPRUN   Non-customizable TCP/IP for z/VM run-time files. For reference, files that can be processed using the TCPRUN section are listed in Figure 13 on page 38 of 6.2.2.4, "TCP/IP for z/VM Product and Sample Configuration Files" on page 37.

TCPCONFIG Customizable sample files. For reference, files that can be processed using the TCPCONFIG section are listed in Figure 14 on page 40.

TCPSYSFILE Non-customizable TCP/IP (binary) system files. For reference, files that can be processed using the TCPSYSFILE section are listed in Figure 15 on page 42.

Notes:

1. To ensure you are notified of any service-related changes to the 4TCPIP30 CATALOG file, make changes to this file using a **VMSES/E local modification**. See Appendix B, "Modifying TCP/IP for z/VM CATALOG Files" on page 85 for information about how to change the 4TCPIP30 CATALOG file in this manner.

2. When changes are made, ensure the only files identified for TCP2PROD processing are those associated with the servers defined for your environment.

3. The TCP2PROD command is not used to install files that are associated with TCPSYSFILE section of the 4TCPIP30 CATALOG file; these files are already present in an appropriate state (due to inclusion of TCP/IP for z/VM as part of the z/VM Version 4 Release 3.0 System DDR).

A.3 TCPCMLST

```
►►──TCPCMLST──ppfname──ftype_abbrv──fm──────────────────────────────────────►◄
```

A.3.1 Purpose

Use the TCPCMLST command to generate a file that lists PTF-numbered parts for which VMSES/E COMMIT processing may be applicable. The generated file (*ppfname* $REMLIST) can be used as input to the VMSES/E **VMFREM** command, which commits specific service levels for your maintenance environment.

Note: The TCPCMLST command is intended for use by the 4TCPIP30 user ID, and should only be used when you commit service levels for TCP/IP for z/VM files.

A.3.2 Operands

ppfname   is the name of the usable form product parameter file used for installing and maintaining TCP/IP for z/VM; the file type must be **PPF**.

ftype_abbrv is the 3-character abbreviation used for PTF-numbered files that correspond to the "real" (or, *base*) CMS file types used for TCP/IP for z/VM files. For example, **FLB** is the part-type abbreviation used for TCP/IP parts that have a base file type of **FLSBIN**. The mapping of file type abbreviations and their corresponding base file types can be found in the VM SYSABRVT file.

fm   is the file mode of the minidisk or directory on which PTF-numbered parts of concern are maintained. By convention, this is the TCP/IP for z/VM **DELTA** minidisk (**4TCPIP30 2D2**, by default) or an equivalent SFS directory.

A.3.3 Usage Notes

- A minidisk or directory must be accessed at file mode A with Read/Write (R/W) status, to allow for the creation of files by TCPCMLST.

- TCPCMLST creates the files listed in Figure 17 (dependent upon current maintenance circumstances):

Figure 17. TCPCMLST - Generated Files

File Name / File Type	Content
ppfname $REMLIST	Lists PTFs that are candidates for commit processing. This file is created when PTF-numbered parts exist that correspond to the selected *ftype_abbrv* abbreviation).
ppfname $CMLSTLG	Lists PTFs identified for commit processing through prior TCPCMLST invocations. This file is produced (or updated) when a *ppfname* $REMLIST file already exists and PTF commit candidates are identified by TCPCMLST.
ppfname $BASLIST	Lists base-level parts that can be removed *after* commit processing has been completed for PTFs listed in the *ppfname* $REMLIST file. The base-level parts listed correspond to one (or more) of the listed PTF-numbered parts.
ppfname $CMBASLG	Lists base-level parts identified for removal through prior TCPCMLST invocations. This file is produced (or updated) when a *ppfname* $BASLIST file already exists and TCPCMLST is invoked and base-level removal candidates are identified along with PTF commit candidates.

- If TCPCMLST is invoked with *ppfname* specified as a question mark (**?**), the command syntax is displayed.

A.3.4 Return Codes

The return codes that can be produced by TCPCMLST are listed and explained here:

0

Explanation: Successful execution; no processing errors were encountered.

1

Explanation: Incorrect invocation. TCPCMLST was invoked with an incorrect number of operands. A message that identifies the missing operand is displayed, in addition to the command syntax.

2

Explanation: Internal error. If return code 2 is returned, processing status is unknown. Contact the TCP/IP for z/VM support group for problem determination and assistance in addressing this type of error.

8

Explanation: Errors encountered; processing has not completed successfully.

A.4 TCPSLVL

```
►►──TCPSLVL─*part_fn*─*part_ft*─*part_fm*────────────────────────────────────────►◄
```

A.4.1 Purpose

Use the TCPSLVL command to display service information that is intrinsic to a TCP/IP executable MODULE file. The information presented is obtained from data that is embedded within the various TEXT decks (files) that comprise a given MODULE.

Note: The TCPSLVL command is intended for use as a diagnostic aid, in consultation with the IBM TCP/IP support group.

When TCPSLVL examines the MODULE you specify, it produces an output line for each TEXT deck in which maintenance data is present. Each line begins with the keyword **SLVL**, followed by the name of a TEXT deck, and its corresponding service indicator. This indicator may reflect either an Authorized Program Analysis Report (APAR) number or an IBM development tracking number. This information, taken as a whole, then can provide an overall (or perhaps, "rule of thumb") indication of the service that is incorporated within a given module.

As an example, this TCPSLVL command will check the service content of the LPR MODULE that resides at file mode "P":

`tcpslvl lpr module p`

For this command, the results produced might be:

```
  SLVL LPRP     PQ39574
  SLVL CMHOSTN  MT01181
  SLVL CMRESOL  PQ28862
```

In this example, APAR level numbers for the LPRP and CMRESOL TEXT files are listed; for the CMHOSTN TEXT file, an internal IBM development tracking number is shown.

A.4.2 Operands

part_fn   the file name of the TCP/IP executable file from which service information is to be obtained.

part_ft   the file type of the TCP/IP file from which service information is to be obtained. Internal service information is available for only TCP/IP MODULE files.

part_fm   the file mode of the minidisk or directory on which the file of interest resides. When specified as an asterisk (*), the first file present in the current search order, which matches the provided *part_fn* and *part_fn*, is evaluated.

A.4.3 Usage Notes

The TCPSLVL command and the information it provides are intended to supplement the information and files that are maintained or used by VMSES/E and its various utilities. TCPSLVL data should be used, at most, only to form generalizations about the service content of TCP/IP modules.

A.4.4 Return Codes

The return codes that can be produced by TCPSLVL are listed and explained here:

0

Explanation: Successful execution; no processing errors were encountered.

1

Explanation: Incorrect invocation / "Help" was requested. TCPSLVL was invoked with an incorrect number of operands, or was invoked with a question mark (?) as the first (or only) operand. In response, the command syntax is displayed.

---

nn

Explanation: Processing error. A nonzero return code (other than 1) indicates an problem was encountered when the file was evaluated. Such a return code will be presented when the specified file cannot be located, or when an error occurs when file contents are examined.

Appendix B. Modifying TCP/IP for z/VM CATALOG Files

This appendix describes how to create a VMSES/E local modification for TCP/IP for z/VM CATALOG files, which are provided as replacement-maintained (or, "full-source") objects. This procedure can also be used to modify other replacement-maintained TCP/IP for z/VM files, such as sample files.

The example provided in this appendix describes how to modify the 4TCPIP30 CATALOG file to prevent files associated with the BOOTP server will from being processed by the TCP2PROD EXEC.

For more information about installing and maintaining local modifications, see Chapters 5, 6, and 7 of the *z/VM: Service Guide* (GC24-5993).

Notes:

1. Throughout this procedure, references are made to several TCP/IP for z/VM installation minidisks. Default device numbers (and equivalent SFS directories, when applicable) for these minidisks are listed in the tables in 5.3, "DASD Storage and User ID Requirements" on page 22.

2. VMSES/E local modifications require a modification identifier (or, *modid*) to be associated with the parts affected by a change; *modid* is a locally-determined value. It should begin with **L**, and is followed by up to 4 alphanumeric characters that identify a specific local modification. For example, L0001.

1 Log on the installation user ID, **4TCPIP30**.

The PROFILE EXEC provided for this user ID (as part of the z/VM Version 4 Release 3.0 System DDR) contains ACCESS commands for the required VMSES/E minidisks — the VMSES/E code minidisk (MAINT 5E5, by default) and the VMSES/E Software Inventory minidisk (MAINT 51D, by default).

2 Issue the CMS QUERY DISK command to verify the VMSES/E code and Software Inventory minidisks are correctly linked and accessed.

query disk   Verify the MAINT 5E5 minidisk is accessed as file mode **B**, and is linked **R/O**.

Verify the MAINT 51D minidisk is accessed as file mode **D**, and is linked **R/W**.

Note: If another user has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the 51D disk in read-only (RR) mode, after which you need issue the appropriate LINK and ACCESS commands for the 51D minidisk. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

3 If necessary, establish the appropriate access to the VMSES/E minidisks.

    a Establish read access to the VMSES/E code minidisk.

      link MAINT 5e5 5e5 rr
      access 5e5 b

    b Establish write access to the Software Inventory minidisk.

      link maint 51d 51d mr
      access 51d d

4 Create a usable PPF file.

Note: This VMFPPF step is necessary **only** if you have made changes to the PPF file since it was last compiled.

vmfppf *ppfname compname*

where:

- *ppfname* is the name of a PPF file that corresponds to the configuration file to be modified. For example, **4tcpip30**. If you have created an override PPF file, use your override file name.

- *compname* is the installed component name with which the configuration file is associated. For example, for TCP/IP for z/VM (4TCPIP30), you would use either: **tcpip** or **tcpipsfs**

5 Establish the correct minidisk access order.

vmfsetup *ppfname compname*

ppfname and *compname* are the same as described in step 4.

6 Copy the highest level of the file to be modified to the local modification (LOCALMOD) minidisk, with the correct file type for the replacement part.

vmfrepl *fn ft ppfname compname* **($select logmod** *modid* **outmode localmod**

where:

- *fn* and *ft* are the real file name and file type of the file being modified. For example: 4TCPIP30 CATALOG

- *ppfname* and *compname* are the same as described in step 4 on page 86.

- *modid* is the local modification identifier for this part. (*modid* is described in note 2 on page 85).

- **localmod** is the symbolic name for the local modification disk in the :MDA. section of the 4TCPIP30 PPF file.

7 XEDIT the part just copied to the LOCALMOD disk, so you can make (and then file) your changes.

xedit *fn ftabbrv-modid fm-local*

The file ID (*fn ftabbrv-modid fm-local*) is returned in message VMFREP2507I. For example: 4TCPIP30 CATL0001 E

8 Make your changes to the displayed file.

For example, assume you've chosen to not use the BOOTP server provided with TCP/IP for z/VM, because you are instead using the supplied DHCP server. To avoid TCP2PROD processing of files associated with the BOOTP server, you need to bypass the 4TCPIP30 CATALOG file shown here:

```
:TCPRUN.
    ...
- &BLD1Z   &DISK4    TCPROFIL EXEC    PROFILE =          BOOTPD
    ...
:ETCPRUN.
    ...
    ...
:TCPCONFIG.
    ...
- &BLD0Z   &BLD4Z    BOOTPTAB SAMPLE  ETC      BOOTPTAB   BOOTPD
    ...
:ETCPCONFIG.
```

9 When you have completed your changes, save the replacement part.

====> file

10 Rebuild the Modified Objects

> **Note**
>
> When you have completed steps 6 through 9 for all of the local modifications necessary for this component, then rebuild these objects.

Build the modified (or, serviced) configuration file(s).

vmfbld ppf *ppfname compname* **(serviced**

Appendix C. PTF Commit Procedure for TCP/IP for z/VM System Files

As TCP/IP for z/VM (binary) system files are updated through corrective service, the storage space required to maintain multiple PTF levels for these parts is likely to become constrained. To allow new service for these files to be received and applied, it will at times be necessary to *commit* existing PTF service — this causes a service status of 'COMMITTED' to be reflected for a PTF, and its corresponding parts are deleted.

Note: For some environments, it may be necessary to commit existing maintenance prior to receiving *any* new corrective service that affects TCP/IP for z/VM system files, depending on how much **DELTA** disk storage has been allocated for maintaining serviced files.

TCP/IP for z/VM system files for which commit processing is likely to be required are listed in Figure 18:

Figure 18. TCP/IP for z/VM System Files
System File Identifier
DTCSSLF FLSBIN DTCSSLS FLSBIN

The procedure that follows illustrates how to use the VMSES/E VMFREM command to commit service associated with a specific PTF for a TCP/IP for z/VM system file.

1 Log on the installation user ID, **4TCPIP30**.

The PROFILE EXEC provided for this user ID (as part of the z/VM Version 4 Release 3.0 System DDR) contains ACCESS commands for the required VMSES/E minidisks — the VMSES/E code minidisk (MAINT 5E5, by default) and the VMSES/E Software Inventory minidisk (MAINT 51D, by default).

2 Issue the CMS QUERY DISK command to verify the VMSES/E code and Software Inventory minidisks are correctly linked and accessed.

query disk

Verify the MAINT 5E5 minidisk is accessed as file mode **B**, and is linked **R/O**.

Verify the MAINT 51D minidisk is accessed as file mode **D**, and is linked **R/W**.

Note: If another user has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the 51D disk in read-only (RR) mode, after which you need issue the appropriate LINK and ACCESS commands for the 51D

minidisk. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

3 If necessary, establish the appropriate access to the VMSES/E minidisks.

    a Establish read access to the VMSES/E code minidisk.

        link MAINT 5e5 5e5 rr
        access 5e5 b

    b Establish write access to the Software Inventory minidisk.

        link maint 51d 51d mr
        access 51d d

4 Create a usable PPF file.

Note: This VMFPPF step is necessary **only** if you have made changes to the PPF file since it was last compiled.

vmfppf *ppfname compname*	where:
	• *ppfname* is the name of the PPF file used for TCP/IP maintenance purposes at your installation. For example, **4tcpip30**. If you have created an override for the PPF file in question, use your override file name.
	• *compname* is the installed component name with which the source file is associated. For example, for TCP/IP for z/VM (4TCPIP30), you would use either: **tcpip** or **tcpipsfs**

5 Establish the correct minidisk access order.

vmfsetup *ppfname compname*	where *ppfname* and *compname* are the same as described in step 4.

6 Identify the PTF-numbered parts (files) for which commit processing is appropriate and generate a file that lists the PTFs that are associated with these parts. The TCPCMLST command creates the *ppfname* $REMLIST file that will be used in the next step.

tcpcmlst *ppfname* **flb** *fm-delta*

where:

- *ppfname* is the same as described in step 4.

- *fm-delta* is the file mode at which the 4TCPIP30 **DELTA** minidisk or directory is accessed.

7 Commit the PTF service for the parts in question, as identified within the *ppfname* $REMLIST file.

vmfrem ppf *ppfname compname* **list** *ppfname* **$remlist a (commit**

where:

- *ppfname* and *compname* are the same as described in step 4 on page 90.

Upon successful completion of the VMFREM command, the parts associated with the PTF in question are erased and its status in the receive software inventory table (*recid* SRVRECS) is set to 'COMMITTED'.

Appendix D. Modifying the TCP/IP for z/VM Default Installation

As the installation and configuration of TCP/IP for z/VM is completed, you may identify one or more TCP/IP services that are not required for your installation. Alternatively, it may be desirable (or necessary) to modify certain aspects of the TCP/IP default installation, to meet requirements that are specific to your organization.

If you choose to modify or eliminate any of the IBM-supplied resources for TCP/IP for z/VM, the changes outlined here should be considered prior to making any changes.

- The **4TCPIP30** user ID is the IBM-supplied user ID for installing and servicing TCP/IP for z/VM. If you choose to use a different user ID or you elect to use different minidisks and/or SFS directories for TCP/IP for z/VM maintenance purposes, suitable PPF override changes must be implemented to reflect your changes.

  Note: If you choose to use an existing or "common-use" user ID (such as the MAINT user ID) to install and maintain TCP/IP for z/VM, minidisks may already be defined which have device numbers that conflict with TCP/IP for z/VM minidisk defaults. If such minidisks exist, you will need to create a PPF override to change the TCP/IP default minidisk device numbers so they are unique within your environment.

- If you choose to use user IDs for TCP/IP server virtual machines that differ from the IBM-supplied user IDs shown in Figure 10, review the section titled "Implications of Assigning Different Server Virtual Machine Names" in Chapter 1 of *z/VM: TCP/IP Level 430 Planning and Customization*.

- Note the following, with regard to the user ID and minidisk information provided in Figure 10 on page 26 of 5.3.1, "DASD Requirements for TCP/IP for z/VM":

  > **Specific Minidisk Requirements**
  >
  > Certain minidisks **must** be defined for the TCP/IP server machines you choose to use, as well as for maintaining TCP/IP for z/VM in your environment. Such minidisks **cannot** be replaced with an equivalent SFS directory.
  >
  > Minidisks to which this requirement applies are listed in Figure 10 with **boldface** virtual device numbers. In addition, no default SFS directory names or SFS 4K block values are provided for these minidisks.

- The minidisks that are associated with the **4TCPIP30**, **TCPMAINT**, and **TCPIP** user IDs (or your chosen equivalents) **must** be maintained in order to provide TCP/IP services within your environment. The remaining user IDs listed in Figure 10 on page 26 of 5.3.1, "DASD Requirements for TCP/IP for z/VM" are associated with servers and clients that provide optional TCP/IP for z/VM services.

- To identify TCP/IP services that are not required for your installation, review the service descriptions provided in *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019). The server-specific configuration chapters of this publication may also help you with determining whether a service or its resources can (or should) be removed.

- Suitable **4TCPIP30 $PPF** file overrides are necessary to avoid references to any TCP/IP user ID that is removed, or to account for any user IDs that are changed. This is also applies to the removal of any minidisks that are uniquely associated with such user IDs.

  If such overrides are created, the **VMFPPF** command must be used to generate a compiled PPF filed that corresponds to your override file name, as well as to create an updated **SERVP2P** PPF file, for use by the VMSES/E SERVICE command.

- The z/VM Version 4 Release 3.0 system directory entry for the **4TCPIP30** user ID (or its equivalent) should be modified such that LINK statements to any removed minidisks are no longer present.

- The z/VM Version 4 Release 3.0 system directory entry for the **MAINT** user ID (or its equivalent) should be modified such that LINK statements to any removed minidisks are no longer present.

- The content of the 4TCPIP30 CATALOG file (used by the TCP2PROD command to copy files into production) should be reviewed for correctness. It may be necessary to modify this file (via a VMSES/E local modification) to bypass references to files that correspond to any TCP/IP user ID that is removed, as well as to any minidisks that are uniquely associated with that user ID. See Appendix A, "TCP/IP Installation and Service Utilities" on page 73 for information about the TCP2PROD command and TCP/IP for z/VM catalog files.

Appendix E. Making Local Modifications to TCP/IP for z/VM Modules

This appendix provides information to assist you in making local modifications to TCP/IP for z/VM modules, and is intended to supplement the local modification process described in Chapter 6, "Procedures for Local Service" of the *z/VM: Service Guide* (GC24-5993). The process described herein is oriented toward local service that is implemented through:

- TEXT file *replacement* — when an existing TCP/IP TEXT file is fully replaced and the file name of the existing TEXT file remains unchanged

- *local updates* to the source for an existing TEXT file (so that a "local replacement" is effectively created)

- TEXT file *substitution* or *addition*, in which an existing TCP/IP TEXT file is fully replaced by one or more new, differently-named TEXT files.

With regard to the latter two types of modification, additional steps to accommodate changes to TCP/IP C and Pascal-based modules or added TEXT files are provided.

When to Use This Information

In general, the VMSES/E local modification process as documented in Chapter 6, "Procedures for Local Service" of the *z/VM: Service Guide* should be used when local modifications are made. *This is especially true* for **common** part types — such as EXEC and ASSEMBLE files — for which VMSES/E itself provides appropriate part handlers.

However, C or Pascal-specific VMSES/E part handlers are not available for TCP/IP **C** and **Pascal** source parts, so local modifications to TCP/IP for z/VM modules based on these languages cannot be processed completely within the VMSES/E environment. Instead, the VMFC and VMFPAS execs provided with TCP/IP for z/VM can be used to facilitate local modifications to its C and Pascal-based modules. Additional steps which make use of these execs, are included in this appendix to help you modify TCP/IP C or Pascal-based objects.

Information about the VMFC and VMFPAS execs can be found in the chapter titled "Using Source Code Libraries," of *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019).

Note that VMFC and VMFPAS are *not* "VMSES/E-compliant" part handlers. They do not process source files in the same manner as VMSES/E part handlers (for example, by using VMFEXUPD); nor do they produce object (TEXT) files that comply with VMSES/E maintenance naming conventions (such as file abbreviations and part numbering). Therefore, several manual steps are required to ensure the TEXT files produced by VMFC and VMFPAS can be used as part of the VMSES/E maintenance process.

For detailed information about installing and maintaining local modifications, see Chapters 5, 6, and 7 of the *z/VM: Service Guide*.

Notes:

1. Throughout this procedure, references are made to several TCP/IP for z/VM installation minidisks. Default device numbers, and equivalent SFS directories (when applicable), for these minidisks are listed in the tables in 5.3, "DASD Storage and User ID Requirements" on page 22.

2. VMSES/E local modifications require a modification identifier, or *modid*, to be associated with the parts affected by a change; *modid* is a locally-determined value. It should begin with **L**, and is followed by up to 4 alphanumeric characters that identify a specific local modification. For example: L0002

E.1.1 Prepare for Local Service

1 Log on the installation user ID, **4TCPIP30**.

The PROFILE EXEC provided for this user ID (as part of the z/VM Version 4 Release 3.0 System DDR) contains ACCESS commands for the required VMSES/E minidisks — the VMSES/E code minidisk (MAINT 5E5, by default) and the VMSES/E Software Inventory minidisk (MAINT 51D, by default).

2 Issue the CMS QUERY DISK command to verify the VMSES/E code and Software Inventory minidisks are correctly linked and accessed.

query disk
    Verify the MAINT 5E5 minidisk is accessed as file mode **B**, and is linked **R/O**.

    Verify the MAINT 51D minidisk is accessed as file mode **D**, and is linked **R/W**.

Note: If another user has the MAINT 51D minidisk linked in write (R/W) mode, you'll obtain only read (R/O) access to this minidisk. If this occurs, have that user re-link the 51D disk in read-only (RR) mode, after which you need issue the appropriate LINK and ACCESS commands for the 51D minidisk. Do not continue with these procedures until a R/W link is established to the 51D minidisk.

3 If necessary, establish the appropriate access to the VMSES/E minidisks.

    a Establish read access to the VMSES/E code minidisk.

        link MAINT 5e5 5e5 rr
        access 5e5 b

    b Establish write access to the Software Inventory minidisk.

        link maint 51d 51d mr
        access 51d d

4 Create a usable PPF file.

> **Note:** This VMFPPF step is necessary **only** if you have made changes to the PPF file since it was last compiled.

vmfppf *ppfname compname*   where:

- *ppfname* is the name of a PPF file that corresponds to the source file to be modified. For example, **4tcpip30**. If you have created an override for the PPF file in question, use your override file name.

- *compname* is the installed component name with which the source file is associated. For example, for TCP/IP for z/VM (4TCPIP30), you would use either: **tcpip** or **tcpipsfs**

5 Establish the correct minidisk access order.

vmfsetup *ppfname compname*   where *ppfname* and *compname* are the same as described in step 4.

6 Access the TCP/IP for z/VM **SOURCE** minidisk (**4TCPIP30 2B3**, by default) or an equivalent SFS directory.

access 2b3 *fm*   where *fm* is an available file mode, perhaps "C."

E.1.2 Receive the Local Service

E.1.2.1 Create the Replacement TEXT File

Create the replacement TEXT file (or, "text deck"). How you perform this step depends on the nature of the TEXT file itself.

- If your modification is comprised of a **replacement** TEXT file, you need only to copy this file to an appropriately named file which VMSES/E can then use.

- If your modification is comprised of a **new** (additional) TEXT file, you need to copy this file to an appropriately named file and update the build list that corresponds to the module affected by your changes.

If either of the above cases are applicable, skip the steps provided for "C and Pascal Source Modifications," and create the required TXT*modid* file, as described in E.1.2.5, "Copy the TEXT File for Use by VMSES/E" on page 99.

- If your modification requires updates to a C or Pascal source file, continue with the steps provided for "C and Pascal Source Modifications."

C and Pascal Source Modifications

E.1.2.2 Add an Update Record to the AUX File

1 Update or create the AUX file for the part being modified (*fn* AUXLCL) and add an entry for the update file; the default AUX file type is "AUXLCL."

Note: Perform this step for each source file you need to modify.

xedit *fn* **auxlcl** *fm-local* **(noprof**
===> **input** *update-ft svclvl* **lc***modid comment*
===> **file**

where:

- *fn* is the file name of the source-maintained part being modified.
- *fm-local* is the file mode of the TCP/IP for z/VM local modification (LOCALMOD) disk.
- *update-ft* is the file type of the update file that contains your local modifications.
- *svclvl* is a service level indicator; by convention, the string "LCL" is most often used for local modifications.
- *modid* is the local modification number (as described in Note 2 on page 95). For example: L0002
- **lc***modid* is the string "LC" concatenated with *modid*; for example: LCL0002

E.1.2.3 Create the Update File for the Part

1 Xedit the source file (*fn* C **or** *fn* PASCAL) with the **CTL** option.

Note: Perform this step for each source file you need to modify.

xedit *fn ft-src fm-src* **(ctl tcpip**

where:

- *fn ft-src fm-src* are the file name, file type and file mode of the source file you are modifying.

- **tcpip** is the name of the TCP/IP control file.

2 Make your changes to the displayed source file. The original source file is *not* changed.

3 When you have completed your changes, save them on the TCP/IP for z/VM local modification (LOCALMOD) disk. When you enter the FILE command in XEDIT, all of your changes are placed in the update file (*fn update-ft*).

===> file = = *fm-local*

where *fm-local* is the file mode of the TCP/IP for z/VM local modification (LOCALMOD) disk.

E.1.2.4 Create an Updated Replacement TEXT File

1 Compile the source to include your updates. Use the VMFC EXEC to compile C source files; use the VMFPAS EXEC to compile PASCAL source files. For more information about the VMFC and VMFPAS execs, see the chapter titled "Using Source Code Libraries," of *z/VM: TCP/IP Level 430 Planning and Customization* (SC24-6019).

Note: Perform this step for each source file you have modified.

a To compile **C**-based source files, issue:

vmfc *fn* **tcpip (** *options*

OR

b To compile **Pascal**-based source files, issue:

vmfpas *fn* **tcpip (** *options*

where:

- *fn* is the name of the source file to be compiled.

- **tcpip** is the name of the TCP/IP control file.

- *options* are C or PASCAL compiler options required for your environment.

After successful completion of the VMFC (or VMFPAS) EXEC, a text file (*fn* TXTLCL A) incorporating your local modifications will exist.

Note: If you choose to use a different compilation method instead of using the VMFC or VMFPAS execs provided with TCP/IP for z/VM, the resulting TEXT file may be named differently than "*fn* TXTLCL."

─────────────── End of C and Pascal Source Modifications ───────────────

E.1.2.5 Copy the TEXT File for Use by VMSES/E

Copy the replacement or new TEXT file (or, "text deck") to the TCP/IP for z/VM local modification (LOCALMOD) disk, with the correct file type for the replacement part.

Note: Perform this step for each TEXT file affected by your modifications.

vmfrepl *fn* **text** *ppfname compname fn ft-txt fm-txt* **($select logmod** *modid* **outmode localmod ftabbr txt**

where:

- *fn* is the file name of the TEXT file in question.

- *ppfname* and *compname* are the same as described in step 4 on page 96.

- *fn ft-txt fm-txt* are the file name, file type and file mode of the TEXT file in question.

- *modid* is the local modification number (as described in Note 2 on page 95). For example: L0002

- **localmod** is the symbolic name for the local modification disk in the :MDA. section of the 4TCPIP30 PPF file.

E.1.3 (Optional) Modify the Build List for Added TEXT Files

If your local modifications do not require the addition of a new TEXT file, skip the steps provided in "Additional Steps for Adding a New TEXT File," and continue with the steps provided in E.1.4, "Rebuild the Modified Objects" on page 102.

If your local modifications are implemented within a unique TEXT file (that is, one not supplied by IBM as part of TCP/IP for z/VM) you need to perform additional steps for this TEXT file to be incorporated by the affected TCP/IP module. These steps, provided in "Additional Steps for Adding a New TEXT File," implement build

list changes that will accommodate the new TEXT file; they would also be required if you find it necessary to replace an IBM-supplied TEXT file with one of a different name.

Additional Steps for Adding a New TEXT File

E.1.3.1 Add an Update Record to the Build List AUX File

1 Update or create the build list AUX file (*fn-blst* AUXLCL) and add an entry for the update file; the default file AUX file type is "AUXLCL."

A complete list of TCP/IP for z/VM build lists is provided in Appendix G, "TCP/IP for z/VM Build Lists" on page 106.

xedit *fn-blst* **auxlcl** *fm-local* **(noprof**
===> input *update-ft svclvl* **lc***modid comment*
===> file

where:

- *fn-blst* is the file name of the build list used to build the affected module (that is, the build list to which the new TEXT file must be added).

- *fm-local* is the file mode of the TCP/IP for z/VM local modification (LOCALMOD) disk.

- *update-ft* is the file type of the update file that contains your local modifications.

- *svclvl* is a service level indicator; by convention, the string "LCL" is most often used for local modifications.

- *modid* is the local modification number (as described in Note 2 on page 95). For example: L0002

- **lc***modid* is the string "LC" concatenated with *modid*; for example: LCL0002

E.1.3.2 Create the Update File for the Build List

1 Xedit the build list source file with the **CTL** option.

xedit *fn-blst* **$exec** *fm-src* **(ctl tcpip**

where:

- *fn-blst* and *fm-src* are the file name and file mode of the build list file you are modifying.

- **$exec** is the file type of the build list.

- **tcpip** is the name of the TCP/IP control file.

2 Make your changes to the displayed source file. The original source file is *not* changed.

Locate the `:OBJNAME.` tag associated with the name of the module that will include your text file. For example, `TCPIP.MODULE`. Add a new `:PARTID.` tag record after the last `:PARTID.` tag for this object (`TCPIP.MODULE`), and before its `:EOBJNAME.` tag.

The new :PARTID. tag should define the file name of the text file you're adding, followed its file type abbreviation, TXT. Use only a single space to separate the tag, the text file name, and the TXT abbreviation, as follows

    :PARTID. *newtxtfn* TXT

In the example that follows, the TCPBLM91 build list is to be updated to add TEXT files to the TCPIP module. To do this, line 176 is copied (or duplicated); then, in the newly created line(s), the string "TCOFFPR" is changed to the name of the added TEXT file(s).

```
 TCPBLM91 EXCnnnnn I2  V 80  Trunc=80 Size=265 Line=178 Col=1 Alt=nn
 ====>
 00087 :EOBJNAME.
 00088 *
 00089 :OBJNAME. TCPIP.MODULE AMODE 31 FROM TCPIP NOMAP
 00090 :BLDREQ. TCPBLCOM.BLDLIST TCPBL492.TCPASCAL.TXTLIB TCPBL492.TCPLAN
 00091          TCPBL492.TCPXXA.TXTLIB
 00092 :GLOBAL. TXTLIB COMMTXT TCPXXA TCPASCAL TCPLANG
 00093 :OPTIONS. CLEAR NOAUTO RLDSAVE NOLIBE NOUNDEF RMODE 24
 00094 :PARTID. TCPIP TXT
 00095 --------------------  80  line(s) not displayed ------------------
 00175 :OPTIONS. LIBE UNDEF RESET VSPASCAL
 00176 :PARTID. TCOFFPR TXT
 00177 :EOBJNAME.
 00178 *
```

3 When you have completed your changes, save them on the TCP/IP for z/VM local modification (LOCALMOD) disk. When you enter the FILE command in XEDIT, all of your changes are placed in the update file (*fn-blst update-ft*).

===> file = = *fm-local*

where *fm-local* is the file mode of the TCP/IP for z/VM local modification (LOCALMOD) disk.

E.1.3.3 Create the Updated Replacement Build List

1 Create a replacement part from the build list $EXEC (source) file. To do this, update the build list file with the VMFEXUPD command.

vmfexupd *fn-blst* **exec** *ppfname compname* **(outmode localmod $select logmod**

where:

- *fn-blst* is the file name of the build list being updated. For example, `TCPBLM91`

- *ppfname* and *compname* are the same as described in step 4 on page 96.

The VMFEXUPD command records the update for the build list in the local version vector table (VVTLCL), adds the update to the $SELECT file, and creates the replacement part (*fn-blst* EXEC).

──────────────── End of Additional Steps for Adding a New TEXT File ────────────────

E.1.4 Rebuild the Modified Objects

┌─ **Note** ──┐
│   │
│ When you have completed the steps described in E.1.2, "Receive the Local Service" through E.1.2.5, │
│ "Copy the TEXT File for Use by VMSES/E" (or E.1.3.3, "Create the Updated Replacement Build List") │
│ for all of the local modifications necessary for this component, then rebuild the modified objects. │
│   │
└──┘

In general, your modification is likely to require other steps associated with the service process to be completed (such as updating the build status table, re-building serviced objects, testing service, and copying the service into production). To complete the service process, continue with one of the steps listed, as appropriate:

- 7.2.2.4, "Update the Build Status Table" on page 50, to complete TCP/IP for z/VM RSU service

- 7.2.3.4, "Update the Build Status Table" on page 60, to complete TCP/IP for z/VM COR service

For reference, a sample VMFBLD command is shown below that can be used to rebuild specific objects within a given build list:

```
vmfbld ppf ppfname compname fn-blst fn-mod.module (serviced
```

where:

- *ppfname* and *compname* are the same as described in step 4 on page 96.

- *fn-blst* is the appropriate TCP/IP for z/VM build list file name. A complete list of TCP/IP for z/VM build lists is provided in Appendix G, "TCP/IP for z/VM Build Lists" on page 106.

- *fn-mod* is the file name of module which incorporates the TEXT file that has been modified.

Also, if you have modified only one module, you may want to manually copy it into production, instead of using the TCP2PROD EXEC. If this is the case, use the VMFCOPY command that follows:

```
vmfcopy fn module fm-bld = = fm-prd (oldd repl sprodid 4tcpip30%tcpip prodid 4tcpip30%tcpip
```

where:

- *fn* is the name of the modified module.

- *fm-bld* is the file mode of the TCP/IP for z/VM build disk on which the module was built.

- *fm-prd* is the file mode of the TCP/IP for z/VM production disk where the module should reside.

The VMFCOPY command will update the VMSES/E PARTCAT file on the appropriate TCP/IP for z/VM production minidisk.

Appendix F. Modifying TCP/IP for z/VM VMNFS Code

This appendix provides information to assist you in making local modifications to the TCP/IP for z/VM NFS server module (VMNFS). Modifications would be required for the NFS server to:

- use of a file handle encryption subroutine different from that in NFSFHCIP ASSEMBLE
- validate SMSG requests in a manner different from its current implementation (affects NFSSMSG C)
- report failed minidisk link attempts in a manner different from its current implementation (affects NFSBADPW C).

Certain modifications may also require changes to the TCPBLC91 EXEC, which is the build list used to build the VMNFS module.

Note: TCP/IP source files are distributed as part of the z/VM Version 4 Release 3.0 System DDR. These files reside on the TCP/IP for z/VM **SOURCE** minidisk (**4TCPIP30 2B3**, by default), or an equivalent SFS directory.

For detailed information about installing and maintaining local modifications, see Chapters 5, 6, and 7 of the *z/VM: Service Guide* (GC24-5993).

F.1 Modifying the NFSHCHIP ASSEMBLE and TCPBLC91 EXEC Files

If you need to modify the NFSHCHIP ASSEMBLE or TCPBLC91 EXEC files, you should follow the steps provided in Chapter 6, "Procedures for Local Service" of the z/VM: Service Guide (GC24-5993). In so doing, the following substitutions may need to be made:

- **zvm** should be: **4tcpip30**
- *compname* should be: **tcpip** or **tcpipsfs**
- *appid* should be: **4tcpip30**
- *fm-local* should be the file mode of the 2C4 minidisk
- *fm-applyalt* should be the file mode of the 2A6 minidisk

You may also find some of the information provided in Appendix E, "Making Local Modifications to TCP/IP for z/VM Modules" on page 94 to be useful, such as the steps provided in E.1.1, "Prepare for Local Service."

Keep in mind that when you get to the following step in the *z/VM: Service Guide*:

- "Rebuilding Objects"

you should return to using this program directory and continue with 7.2.2.4, "Update the Build Status Table" on page 50.

F.2 Modifying VMNFS C Source Files

If you need to modify the source for the NFSSMSG or NFSBADPW C files, use the information provided in Appendix E, "Making Local Modifications to TCP/IP for z/VM Modules" on page 94, especially that which is specific to "C and Pascal Source Modifications."

Appendix G. TCP/IP for z/VM Build Lists

This appendix provides a complete list of the VMSES/E build lists used to maintain TCP/IP for z/VM. This information has been provided to help you determine which build list to use with VMSES/E commands when you need to build or service specific TCP/IP objects, and to assist you with making local modifications. For more information about build list content and formats, see the *z/VM: VMSES/E Introduction and Reference* (GC24-5994).

The build lists identified in the tables that follow can be found on the 4TCPIP30 2B2 (BASE1) minidisk. However, before using the information within a given build list, the 4TCPIP30 2D2 (DELTA) minidisk should be checked for a more current, serviced counterpart; this will ensure the most current build list file is referenced.

Also, note that the minidisks shown under the "Build String" headings are 4TCPIP30 minidisk defaults. If a PPF override has been used in your environment to change Build String minidisks or SFS directories, use your values when you determine which files are affected by a build list.

G.1 TCP/IP for z/VM Build Lists

Figure 19 lists the VMSES/E build lists used for TCP/IP for z/VM, and provides general information about the objects (files) managed by each:

Figure 19 (Page 1 of 2). VMSES/E Build Lists - TCP/IP for z/VM			
Build List Name	**VMSES/E Part Handler**	**Build String (Minidisk)**	**Build List Description / Affected Objects**
TCPBL491	VMFBDCOM	BUILD1 (491)	Full-replacement objects built to the 491 minidisk
TCPBL492	VMFBDCOM	BUILD3 (492)	Full-replacement objects built to the 492 minidisk
TCPBLALL	VMFBDMLB	BUILD3 (492)	ALLMACRO MACLIB build list
TCPBLTRP	VMFBDMLB	BUILD3 (492)	TFTPRP MACLIB build list
TCPBLCOM	VMFBDTLB	BUILD3 (492)	COMMTXT TXTLIB build list
TCPBLGDD	VMFBDTLB	BUILD3 (492)	GDDMXD TXTLIB build list
TCPBLXAW	VMFBDTLB	BUILD3 (492)	XAWLIB TXTLIB build list
TCPBLDPI	VMFBDTLB	BUILD3 (492)	DPILIB TXTLIB build list
TCPBLRPC	VMFBDTLB	BUILD3 (492)	RPCLIB TXTLIB build list
TCPBLRPT	VMFBDPMD	BUILD3 (492)	VMRPC TXTLIB build list
Notes:			
1. Language Environment run-time library support must be available when building objects identified in this build list.			

Figure 19 (Page 2 of 2). VMSES/E Build Lists - TCP/IP for z/VM			
Build List Name	**VMSES/E Part Handler**	**Build String (Minidisk)**	**Build List Description / Affected Objects**
TCPBLOLD	VMFBDTLB	BUILD3 (492)	OLDXLIB TXTLIB build list
TCPBLXTL	VMFBDTLB	BUILD3 (492)	XTLIB TXTLIB build list
TCPBLX11	VMFBDTLB	BUILD3 (492)	X11LIB TXTLIB build list
TCPBLSNA	VMFBDLLB	BUILD1 (491)	SNALINK LOADLIB build list
TCPBLXNX	VMFBDLLB	BUILD1 (491)	XNX25 LOADLIB build list
TCPBLSNM	VMFBDLLB	BUILD1 (491)	SNMPLIB LOADLIB build list
TCPBLM91	VMFBDMOD	BUILD1 (491)	MODULE objects built to the 491 minidisk
TCPBLM92	VMFBDMOD	BUILD3 (492)	MODULE objects built to the 492 minidisk
TCPBLC91 [1*]	VMFBDMOD	BUILD1 (491)	C-based MODULE objects built to the 491 minidisk
TCPBLC92 [1*]	VMFBDMOD	BUILD3 (492)	C-based MODULE objects built to the 492 minidisk
TCPBLP91 [1*]	VMFBDTLB	BUILD1 (491)	VMFBDPMD-dependent MODULE objects built to the 491 minidisk
TCPBLP92 [1*]	VMFBDTLB	BUILD3 (492)	VMFBDPMD-dependent MODULE objects built to the 492 minidisk
TCPBLCSL	VMFBDCLB	BUILD1 (491)	TCPCSLIB CSLIB build list
TCPBLFSF	TCP2PROD	N/A	DTCSSLF system file build list
TCPBLFSS	TCP2PROD	N/A	DTCSSLS system file build list
TCKBLDES	VMFBDTLB	BUILD3 (492)	DES TXTLIB build list
TCKBLKRB	VMFBDTLB	BUILD3 (492)	KRB TXTLIB build list
TCKBLKDB	VMFBDTLB	BUILD3 (492)	KDB TXTLIB build list
TCKBLBPL	VMFBDTLB	BUILD3 (492)	BPLDBM TXTLIB build list
TCKBLC91 [1*]	VMFBDMOD	BUILD1 (491)	Kerberos C-based MODULE objects built to the 491 minidisk
TCKBLC92 [1*]	VMFBDMOD	BUILD3 (492)	Kerberos C-based MODULE objects built to the 492 minidisk
Notes:			
1. Language Environment run-time library support must be available when building objects identified in this build list.			

Appendix H. Moving TCP/IP for z/VM to SFS Directories

By default, TCP/IP for z/VM is installed to minidisks as part of the z/VM Version 4 Release 3.0 System DDR, during the initial install of z/VM Version 4 Release 3.0 itself. However, you can move certain TCP/IP for z/VM minidisks — the **service** minidisks — to Shared File System (SFS) directories. Refer to Figure 10 in 5.3.1, "DASD Requirements for TCP/IP for z/VM" on page 26 to see which minidisks can reside in SFS space. When service disks are moved to SFS directories, you can use either the default file pool (VMSYS) or a file pool of your choosing.

A summary of the steps necessary to move TCP/IP for z/VM service minidisks to SFS space are:

- Allocate space in the user-defined (or default) file pool

- Provide the installation user ID, 4TCPIP30, access to the file pool

- Create the necessary TCP/IP for z/VM subdirectories

- Copy files from minidisks to the new SFS subdirectories

- Create a PPF override, if not using default file pool or subdirectory names.

Where to Next

You need to do one of the following:

- To place TCP/IP for z/VM into SFS directories using **4TCPIP30 PPF file defaults**, you need to continue with instructions provided in the *z/VM: Installation Guide*, in the appendix titled "Moving Components to SFS Directories."

- To place TCP/IP for z/VM into your own **user-defined file pool or SFS directories**, continue with the instructions in H.1.1, "Setup the SFS File Pool and Required Directories" on page 109.

After you have chosen one of the previous options and completed the steps required (whether provided in this appendix or in the *z/VM: Installation Guide*) you need to return to 6.2.2.1, "Configure TCP/IP for z/VM for Your Installation" on page 36 and complete the installation of TCP/IP for z/VM.

H.1.1 Setup the SFS File Pool and Required Directories

Notes:

1. The steps that follow help you determine TCP/IP for z/VM file pool space requirements, enroll the 4TCPIP30 user ID in a file pool, and define the required SFS directories. If not all of these steps are required — for example, you are changing only SFS directory names — adapt these instructions as needed for your environment.

2. The steps that follow assume the use of a user-defined SFS file pool name; if you are using the z/VM default file pool name (VMSYS), you should substitute "VMSYS" in place of the text "user-defined file pool" or the variable *userfilepool*, when appropriate.

3. For information about planning for, generating, and managing a file pool and server, see *z/VM: CMS File Pool Planning, Administration, and Operation* (SC24-5949).

1 Determine the number of 4K blocks required for your SFS directories by adding up the 4K block requirements for each SFS directory you plan to use. If you intend to use *all* of the TCP/IP for z/VM SFS directory defaults, the 4K block requirements are summarized in Figure 10 on page 26.

This information will be used when the 4TCPIP30 user ID is enrolled in the user-defined file pool.

2 Enroll user 4TCPIP30 in the user-defined file pool, using the ENROLL USER command:

```
ENROLL USER 4TCPIP30 userfilepool: (BLOCKS blocks
```

where *blocks* is the number of 4K blocks you calculated in the previous step.

Note: This must be done from a user ID that is an administrator for the user-defined file pool.

3 Determine if there are enough blocks available in the file pool to accommodate TCP/IP for z/VM. This information can be obtained via the QUERY FILEPOOL STATUS command. Near the end of the output from this command is a list of minidisks in the file pool and the number of blocks free. If the number of blocks free is smaller than the total number needed to install TCP/IP for z/VM, you need to add space to the file pool before you continue with this process. See *z/VM: CMS File Pool Planning, Administration, and Operation* for information about adding space to a file pool.

4 Use the CREATE DIRECTORY command to create the required subdirectories; the default subdirectories are listed in Figure 10 on page 26. If necessary, refer to the *z/VM: CMS Command and Utility Reference* for more information about the CREATE DIRECTORY command.

set filepool *userfilepool:*
create directory *dirid*
create directory *dirid*

dirid is the name of the SFS directory you're creating, such as the default names:

```
create directory userfilepool:4tcpip30.tcpip
create directory userfilepool:4tcpip30.tcpip.local
create directory userfilepool:4tcpip30.tcpip.delta
create directory userfilepool:4tcpip30.tcpip.applyalt
create directory userfilepool:4tcpip30.tcpip.applyprod
create directory userfilepool:4tcpip30.tcpip.object
```

5 If you intend to use an SFS directory as the work space for the 4TCPIP30 user ID, include the following IPL control statement in the 4TCPIP30 directory entry (after the `INCLUDE TCPCMSU` statement):

```
IPL CMS PARM FILEPOOL userfilepool
```

This will cause CMS to automatically access the 4TCPIP30's top directory as file mode A.

H.1.2 Copy Minidisk Files to SFS Directories

1 Copy the files from the TCP/IP for z/VM minidisks (or from the VMSYS file pool, if TCP/IP for z/VM is already installed there) to your new user-defined SFS file pool and directories, using the VMFCOPY command.

Note: Repeat the ACCESS and VMFCOPY commands that follow for each minidisk you need to copy. If necessary, see Figure 10 on page 26 for TCP/IP for z/VM default minidisk device numbers and SFS directory names.

access *vdev* **e**
access *dirid* **f**
vmfcopy * * e = = f (prodid 4tcpip30%tcpip olddate replace

where:

- *vdev* is the minidisk from which you are copying files.

- *dirid* is the name of the (target) SFS directory to which you are copying files.

- **4tcpip30%tcpip** is the PRODID defined within the 4TCPIP30 PPF file, for both the minidisk *and* SFS components of TCP/IP for z/VM.

The VMFCOPY command will update the VMSES PARTCAT file on the target directory.

H.1.3 Create a Product Parameter File (PPF) Override

This section provides information to help you create a product parameter file (PPF) override. The example used in this section changes the name of the SFS file pool where TCP/IP for z/VM files reside. See the *z/VM: VMSES/E Introduction and Reference* for more information about PPF file overrides.

Note: Do **not** directly modify the product-supplied 4TCPIP30 $PPF or 4TCPIP30 PPF files to change the VMSYS file pool name or any other installation parameters. If the 4TCPIP30 $PPF file is serviced, the existing $PPF file will be replaced, and any changes to that file will be lost. By creating your own $PPF override, your updates will be preserved.

The following process describes changing the TCP/IP for z/VM default file pool name from "VMSYS" to "MYPOOL1":

1 Create a new $PPF override file or edit an existing override file.

xedit *overname* **$ppf** *fm***2**

overname is the PPF override file name (such as "mytcpip") that you want to use.

fm is an appropriate file mode. If you create this file yourself, specify a file mode of A.

If you modify an existing override file, specify a file mode of A or D, based on where the file currently resides (A being the file mode of a R/W 191 minidisk, or equivalent; D, that of the MAINT 51D minidisk).

2 Create (or modify as required) the Variable Declarations (**:DCL.**) section for the **TCPIPSFS** override area so it resembles the :DCL. section that follows:

```
:OVERLIST. TCPIPSFS
:TCPIPSFS. TCPIP 4TCPIP30
*
*=========================================================================*
* Override for TCPIPSFS SFS Component -- File Pool Name Change            *
*=========================================================================*
:DCL. UPDATE
 &191    DIR   MYPOOL1:4TCPIP30                 * A-disk
 &LMODZ  DIR   MYPOOL1:4TCPIP30.TCPIP.LOCAL     * Local modifications
 &DELTZ  DIR   MYPOOL1:4TCPIP30.TCPIP.DELTA     * Product service
 &APPLX  DIR   MYPOOL1:4TCPIP30.TCPIP.APPLYALT  * AUX and Inv files (ALT)
 &APPLZ  DIR   MYPOOL1:4TCPIP30.TCPIP.APPLYPROD * AUX and Inventory file
 &BAS1Z  DIR   MYPOOL1:4TCPIP30.TCPIP.OBJECT    * Base disk
 &BAS2Z  DIR   MYPOOL1:4TCPIP30.TCPIP.SOURCE    * Source code (Optional)
:EDCL.
:END.
```

This override will update the TCPIPSFS override area :DCL. section of the 4TCPIP30 $PPF file.

3 Save your changes when they are complete.

===> file = = *fm*

> where *fm* is the file mode where your override file resides (or A if you are creating one).

4 If your $PPF override file was created at file mode A, copy it to file mode D — the Software Inventory minidisk (MAINT 51D).

copyfile *overname* **$ppf** *fm* **= = d (olddate**

5 Compile your changes to create the usable PPF file, *overname* **PPF**.

vmfppf *overname* **tcpipsfs**

> where *overname* is the file name of your $PPF override file.

Now that the **overname PPF** file has been created, specify "*overname*" instead of "4TCPIP30" as the PPF name to be used for any VMSES/E commands that require a PPF name.

Appendix I. Copying TCP/IP for z/VM Client Code to the Y-Disk

To simplify access to TCP/IP client functions for your user community, you may find it desirable to copy all, or a subset of, TCP/IP for z/VM client code to the z/VM product code disk. Doing so will avoid the need for users to additionally link and access the TCPMAINT 592 minidisk.

As well, applications that use certain programming interfaces may require TCP/IP-specific information to be available for proper operation. For example, information defined in the TCPIP DATA file is referenced by:

- the OpenEdition® VM Sockets Library to correctly identify the TCP/IP virtual machine when the TCPIPID environment variable is not set. See the *z/VM: Language Environment 1.8 C Run-Time Library Reference* (SC24-6038) for more information.

- the VMTCPDT routine (which resides in the VMMTLIB TXTLIB that is associated with the VMLIB Callable Services Library (CSL). See the *z/VM: CMS Callable Services Reference* (SC24-6004) for more information about the VMTCPDT CSL routine.

- various functions provided as part of the CMS REXX Socket library. See the *z/VM: REXX/VM Reference* (SC24-6035) for more information.

To copy TCP/IP for z/VM client files to the Product Code minidisk (typically the MAINT 19E minidisk — the Y-disk), use the following procedure **after** you have installed TCP/IP for z/VM.

Warning - Cross-Component File Overlap Considerations

Before you copy *any* TCP/IP for z/VM client files to the Y-disk (or a similar "*common use*" minidisk), you should first determine whether any conflicts exist between the TCP/IP client files you choose to copy, and those present on the target (Y-disk) minidisk. If any file conflicts are found, these should be addressed and resolved with respect to your installation environment before you continue with the procedure that follows.

Notes:

1. You will need to repeat this procedure each time you apply service to TCP/IP for z/VM.

2. Use discretion when wildcards (*) are used for both the *fn* (file name) and *ft* (file type) parameters of the VMFCOPY commands shown in this section, since files that exist on the Y-disk can be replaced with similarly-named TCP/IP counterpart files. The overlay of certain files may be warranted in some cases, and may be undesirable for others.

An example of this latter case is cited here. Both TCP/IP for z/VM and the IBM Language Environment for MVS & VM, Version 1 Release 8 have several **H** files that are identically named, but differ in content. These files are:

```
FCNTL    H    IF      H    IN       H    INET    H
IOCTL    H    NETDB   H    RESOURCE H    SOCKET  H
STRINGS  H    TTYDEV  H    TYPES    H    UIO     H
```

An overlay of the IBM Language Environment **H** files (already present on the Y-disk) by their TCP/IP counterparts may create problems when applications that expect (and rely upon) the content of the IBM Language Environment files are developed or rebuilt.

3. Before copying TCP/IP for z/VM files to another minidisk, ensure adequate storage space is available to maintain the files you have selected.

1 Log on the **MAINT** (or equivalent) user ID.

2 Process TCP/IP for z/VM files used by or available to TCP/IP clients.

link tcpmaint 592 592 rr	**Note:** If the Y-disk is not defined as the 19E
access 592 e	minidisk in your environment, substitute the
access 19e f	appropriate device number for this minidisk.

vmfcopy *fn ft* **e = = f2 (olddate replace sprodid 4tcpip30%tcpip prodid 4tcpip30%tcpip**

> The VMFCOPY command will update the VMSES/E PARTCAT file on the Y-disk.
>
> Wildcards (*) can be substituted for *fn* (file name) and *ft* (file type), but should be used with discretion.

3 (Optional) Erase any TCP/IP for z/VM files that you do not want to remain on the Y-disk — for example, any MAP files that correspond to TCP/IP for z/VM modules re-built during service. Refer to the VMSES/E PARTCAT file on Y-disk to determine which files are associated with TCP/IP for z/VM.

Note: Additional information about various TCP/IP for z/VM client files, and their association with specific TCP/IP functions, is available on-line via the TCP/IP for z/VM home page on the World Wide Web. The URL for this home page is:

```
http://www.vm.ibm.com/related/tcpip/
```

vmferase file *filename filetype* **f**	See the *z/VM: VMSES/E Introduction and Reference* for more information about the VMFERASE command and options that may help you remove specific files.

4 Re-save the CMS saved system, to return the Y-disk to shared status. See the "Placing (Serviced) Components into Production" section of the *z/VM: Service Guide* for detailed information about how to save the CMS saved system.

Appendix J. z/VM Express Service Instructions

The service instructions that follow are for use by customers who have chosen to apply service to their z/VM Version 4 Release 3.0 system using z/VM Express Service support — most notably, the VMSES/E **SERVICE** and **PUT2PROD** commands. These commands automate the majority of the steps necessary to install Preventive or CORrective service, as well as those which place new service into production.

Note: z/VM Express Service instructions can also be found in the *z/VM Installation and Service Summary* pamphlet (provided with the *z/VM: Installation Guide*).

These instructions are to be used for installing either preventive service (the Recommended Service Upgrade, or **RSU**) *or* corrective (**COR**) service for TCP/IP for z/VM.

1 Before you proceed, make sure that you have a current backup of your system. If not, refer to the step titled "Store a Backup Copy of the z/VM System on Tape," in the "Post Load Installation Tasks" chapter of the *z/VM: Installation Guide*.

2 Log on to the **MAINT** user ID.

logon maint
`Enter`

3 Mount the RSU or COR service tape and attach the tape drive as 181.

attach tapeaddr * 181

If the RSU or COR tape encompasses multiple volumes, you will be requested to mount additional tapes as these are needed by the service exec.

Note: If you have received the RSU or COR electronically (or on CD-ROM), follow the appropriate instructions to retrieve and decompress the envelope file to the MAINT user ID A-disk. Decompression is currently done by using the DETERSE MODULE, which produces a decompressed file that has one of the following file name formats:

- RPTF*num* — for an RSU envelope
- VPTF*num* — for a COR service envelope.

The file type for either of these files (once decompressed) must be **SERVLINK**. You will need to enter the file name of your decompressed file as part of the SERVICE command, used in the next step.

4 Load the service

    a If you are servicing **TCP/IP for z/VM only**, issue *one* of the following commands (based on the service media in use):

service tcpip   for installing service from a tape.

OR

service tcpip rptf*num*   for installing electronic service (or from CD-ROM), where "rptf*num*" is the envelope file name, as previously discussed.

    This installs all TCP/IP for z/VM service that is present on the tape.

    b If you are servicing **all products** for which service has been provided, issue *one* of the following commands (again, based on the service media in use):

service   for installing service from a tape.

OR

service all rptf*num*   for installing electronic service (or from CD-ROM), where "rptf*num*" is the envelope file name, as previously discussed.

    This installs all service for all of the products that are present the tape.

5 Perform any testing you find necessary, or that is required for your environment, before you place this new service into production. See 7.2.2.6, "Test the New Service" on page 53 for more information about service testing recommendations.

6 Place the new service into production.

> ┌─ **Note - Shut Down of Service Machines** ─────────────
>
> This step will logoff *any* appropriate product server virtual machine(s) in your environment, in order to place this service into production.
>
> Therefore, you may want to perform this step as part of a planned system outage.

ipl cms
put2prod

This will copy all code — including service — from test build disks to the appropriate production build disks, for any component that has been serviced since the last time the PUT2PROD utility was executed.

7 Review the TCP2PROD message log (TCP2PROD $MSGLOG). If necessary, correct any problems before you proceed with the next step.

vmfview tcp2prod

8 Shutdown and re-IPL your system.

shutdown reipl
[Enter]

Notices

IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes to the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Trademarks and Service Marks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

FunctionPac
IBM
IBMLink
Language Environment
OpenEdition
RACF
RETAIN
VM/ESA
z/VM

Other company, product, and service names may be trademarks or service marks of others.

Reader's Comments

TCP/IP for z/VM

You may use this form to comment about this document, its organization, or subject matter. Please understand that your feedback is of importance to IBM, but IBM makes no promises to always provide a response to your feedback. If you prefer to provide feedback electronically, please e-mail your comments to: **vmtcpip@vnet.ibm.com**

For each of the topics below please indicate your satisfaction level by circling your choice from the rating scale. If a statement does not apply, please circle N.

```
┌─ RATING SCALE ────────────────────────────────────────────────────────────────┐
│                                                                                 │
│         very          ◄---------------------------►      very          not       │
│       satisfied                                       dissatisfied   applicable  │
│                                                                                 │
│         1          2          3          4                5             N        │
│                                                                                 │
└─────────────────────────────────────────────────────────────────────────────────┘
```

	Satisfaction					
Ease of product installation	1	2	3	4	5	N
Time required to install the product	1	2	3	4	5	N
Contents of program directory	1	2	3	4	5	N
Readability and organization of program directory tasks	1	2	3	4	5	N
Necessity of all installation tasks	1	2	3	4	5	N
Accuracy of the definition of the installation tasks	1	2	3	4	5	N
Technical level of the installation tasks	1	2	3	4	5	N
Installation verification procedure	1	2	3	4	5	N
Ease of customizing the product	1	2	3	4	5	N
Ease of migrating the product from a previous release	1	2	3	4	5	N
Ease of putting the system into production after installation	1	2	3	4	5	N
Ease of installing service	1	2	3	4	5	N

- Did you order this product as an independent product or as part of a package?

    □ Independent
    □ Package

    What type of package was ordered?

    □ CustomPac
        □ FunctionPac®
        □ SystemPac
    □ System Delivery Offering (SDO)
    □ Other - Please specify type: _____

123

- Is this the first time your organization has installed this product?

    □ Yes
    □ No

- Were the people who did the installation experienced with the installation of VM products using VMSES/E?

    □ Yes

        How many years of experience do they have? _____

    □ No

- How long did it take to install this product? _____

- If you have any comments to make about your ratings above, or any other aspect of the product installation, please list them below:

    _____

    _____

    _____

    _____

    _____

    _____

    _____

    _____

Please provide the following contact information:

_____
Name and Job Title

_____
Organization

_____

_____
Address

_____
Telephone

Thank you for your participation.

Please send the completed form to the following address, or give to your IBM representative who will forward it to the TCP/IP for z/VM Development group:

IBM Corporation
TCP/IP for VM Development
Dept. G79G
1701 North Street
Endicott, NY 13760

IBM®

Program Number: 5739-A03

Printed in U.S.A.