TOPS–10/TOPS–20
Autopatch
Procedures/Reference
Manual

AA-H729A-TK

January 1981

This manual describes the Automatic Patching Facility designed to patch distributed versions of Digital Equipment Corporation software products.

This is a new manual.

OPERATING SYSTEM: TOPS–10 Version 7.01
                                     TOPS–20 Version 4

SOFTWARE: PEP Version 1
                                          PEPB Version 1

digital equipment corporation • marlboro, massachusetts

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DECnet	IAS
DECUS	DECsystem–10	MASSBUS
Digital Logo	DECSYSTEM–20	PDT
PDP	DECwriter	RSTS
UNIBUS	DIBOL	RSX
VAX	EduSystem	VMS
		VT

Contents

Chapter 3   Reference

Tables

Figures

Preface

This manual describes Digital Equipment Corporation's automatic software patching facility called Autopatch.

Organization of the Manual

The Autopatch Procedures/Reference Manual consists of three chapters.

The first is an overview chapter that both system administrators and operators should read. The chapter explains the concept of automatic patching, lists the system requirements for utilizing the Autopatch tools, describes the contents of the Autopatch tape, and describes the special directories that you need to set up to prepare your system for automatic patching. There are eight figures to supplement the overview in Chapter 1.

The second chapter is addressed to the person at your site who is responsible for installing Autopatch and patching your software. It is devoted entirely to procedures: how to set up the directories described in Chapter 1 and install the files required for automatic patching; then, how to do the patching process itself. Once you become familiar with the step-by-step procedures in Chapter 2, the two flow charts on the divider between Chapters 1 and 2 should be useful as quick reference for you.

The third chapter also is addressed to both system administrators and operators. In addition to describing the AUTOPATCH command and the other commands that can be used in the patching process, it includes a section on error messages and recovery procedures, and a reference section.

Complementary Files

A feature of Autopatch is its use of two files from the patch tape that you will need to print out because they complement the instructions in the manual; that is, they add specific information and instruction not given in

the manual. These procedures are not difficult, but they do require close attention to the step-by-step instructions you receive from the manual and the files you print. Several other files on the patch tape can also be printed out to help you run checks on your operation of Autopatch.

Related Documents

TOPS–10 Monitor Installation Guide
TOPS–10 Operating System Commands
TOPS–10 Operator's Guide

TOPS–20 Commands Reference Manual
TOPS–20 Operator's Guide
TOPS–20 Software Installation Guide
TOPS–20 User Utilities Guide

Symbology

Conventions used in this manual

(Red print)	Anything you are expected to type on your terminal. Any printing the system does (prompting characters, recognition, messages) is in black.
(RET)	Indicates when you should press the RETURN key.
(ESC)	Indicates when you should press the (ESC) key.
@	In addition to being the TOPS–20 system prompt, this is used to indicate an indirect file.
.	The TOPS–10 system prompt.

Chapter 1
Introduction

1.1 The Autopatch Concept

Definitions and descriptions helpful to both system administrators and operators are included in this introductory chapter.

1.1.1 Autopatch Defined

Autopatch is a combination of software tools and operational procedures for easily incorporating recommended changes into Digital Equipment Corporation software products. (A product is autopatchable if it is a supported software product that has been declared autopatchable by Digital.) Autopatch is designed and engineered to be used at your site.

Autopatch is designed to enable you, the user, to avoid the time-consuming, complicated, and error-prone manual procedure for applying patches. Once Autopatch has been installed at your site, it allows you to bring to current patch level the latest versions of DIGITAL's software with little effort or operator intervention.

Patching a product with Autopatch consists of making the appropriate corrections to the product's component files, reexecuting the procedures that generated the executable (.EXE) files, using the patched files as input to the process, and then, when the product has been rebuilt, replacing the old versions of the .EXE files on the system area with the patched versions.

A single user interface, a new utility called the Patch Executive Program (PEP), maintains patch history and status for each autopatchable product and coordinates the activities required to update your software. You can, in fact, initiate the patching and rebuilding operation with a single command to PEP, and the operation can run, unattended, at your convenience.

Autopatch is distributed initially as a kit containing these complementary items:

- the Autopatch Procedures/Reference Manual

- the patch tape itself, a separate tape for products running on TOPS–10 and another for those running on TOPS–20. In addition to containing machine-readable patches, each patch tape contains two documentation files that are to be used with the manual. (Both are in save sets on the patch tape and are to be printed out at different stages to give you specific instructions for completing those stages.)

  See Figure 1–1; see also Section 1.1.4 of this chapter for a description of the patch tape.

Once the kit is in hand, you need to verify the system requirements for doing autopatching at your site and review the operator capabilities for installing and running Autopatch.

1.1.2 System Requirements

Autopatch runs on all supported configurations of DECsystem–10 and DECSYSTEM–20 hardware that have sufficient on-line mass storage. See Table 1–1.

Table 1–1: Autopatch Resource Requirements

DECsystem–10 Requirements

Operating System:	TOPS–10, version 7.01 or later
System Software:	GALAXY Batch, version 2 or later (Multiprogram Batch (MPB) not supported)
Hardware:	KS10, KI10, and KL10 CPU 256K memory (minimum) 1 RP06 or RM03 Disk 1 9-track magnetic tape drive (800 and/or 1600 BPI)

DECSYSTEM–20 Requirements

Operating System:	TOPS–20, version 4 or later
System Software:	GALAXY Batch, version 4 or later
Hardware:	KS10 or KL10 CPU 256K Memory (minimum) 1 RP06 or RM03 Disk 1 9-track magnetic tape drive (800 and/or 1600 BPI)

1.1.3 Operator Qualifications

It is our expectation that, if you have had varied directory-making experience and are familiar with the BACKUP, CREDIR and REACT programs on TOPS–10; the DUMPER program and BUILD command on TOPS–20; and running GALAXY batch jobs, you will have no difficulty initializing Autopatch at your site.

The procedures to set up the several disk directories that Autopatch requires for file storage and interchange in the patching process, and thus to initialize (install) Autopatch, are described in detail in Chapter 2, Procedures, Sections 2.1.1 and 2.2.1.

1.1.4 Patch Tape Structure

You now need to be aware of the structure and contents of the patch tape. It contains everything you need for your on-site patching, and is organized into several save sets. (See Figure 1–1, Autopatch Tape Structure.) A save set is defined as a group of files on tape, stored as the result of one SAVE command to either BACKUP (TOPS–10) or DUMPER (TOPS–20).

1.1.4.1 Current Patch Save Set — The current patch save set is the first one on the tape. It contains a set of files for every autopatchable software product requiring correction. The most important files in this set are:

- README.nnn, where nnn is the number of the current patch tape. This file contains summary information about the current patch save set, and is to be printed and used in conjunction with the manual.

- a patch directory file (file type .DPD), which contains a record of all the patches in this save set, and is to be processed by the Patch Executive Program.

- the patch files, a set of files for each product requiring patching, which contain the actual patches you will apply.

- patch retrieval control files (.Pnn and .Ann), one for each autopatchable product, which you can use to copy the patch files from the patch tape to disk.

1.1.4.2 Installation Save Set — The installation save set is the second save set on the patch tape. It contains all of the files (PEP, utility programs, control files, etc.) required to set up your system for autopatching. The most important file in this set is:

- INSTAL.DOC, which contains installation information *not contained in this manual*. It is to be printed out and its instructions followed in conjunction with the procedures described in the manual. (Chapter 2, Procedures, will tell you the proper time to obtain a listing of INSTAL.DOC.)

1.1.4.3 Cumulative Patch Save Sets — The cumulative patch save sets are a sequence of patch save sets from previous patch tapes. These accumulated patch save sets are provided to enable you to install a newly acquired software product at the current patch level without having to handle more than one tape.

See Figure 1–1 for the structure of the Autopatch tape.

Figure 1–1: Autopatch Tape Structure
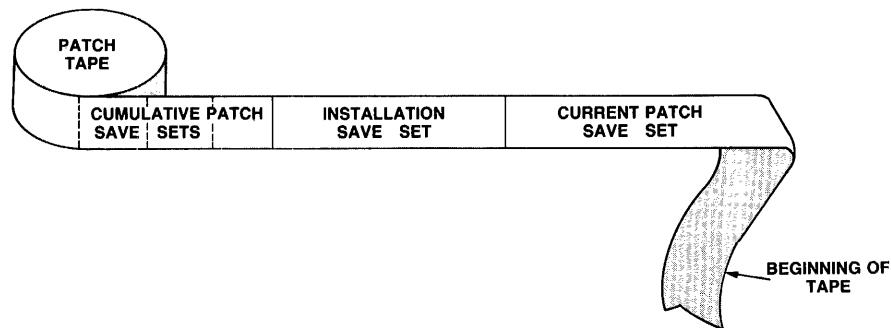
MR-S-954-81

1.1.5 Distribution of Patch Tapes

Additional patch tapes will be made available to you periodically, as patches are found necessary and as the number of autopatchable products grows. Each tape, as mentioned earlier, will be cumulative; that is, it will contain all the previous patches, so that you can update any autopatchable product at any time.

1.2 The Autopatch Environment

Before you can autopatch any software on your system, you must prepare an autopatching environment consisting of several disk directories. In these directories, you will need to install the files required for the auto-patching operation. Some of these files will be found on the patch tape; others are supplied on the distribution tape for each product. However, you will have to set up these directories only once, since they are to be retained on disk and used whenever you wish to apply distributed patches to update your software. The next sections deal with the disk directories and the files that make automatic patching possible.

1.2.1 Overview of the Directory Requirements

Described below are the four storage areas (directories) and their functions; the steps to set them up are shown in Chapter 2, Procedures, Section 2.1 for TOPS–10, Section 2.2 for TOPS–20.

1.2.1.1 Product Distribution Directory — Holds the input and control files (.REL, .MAC, .LNK, etc.) that you use to build the executable files (.EXE)

of the product. The files for this directory are distributed with each release of a software product. (You can generate a single directory to hold all the input files for all of the autopatchable products at your site, or a separate directory for each autopatchable product.) See Figure 1–2.

Figure 1–2:   Copying Files from the Product Distribution Tape
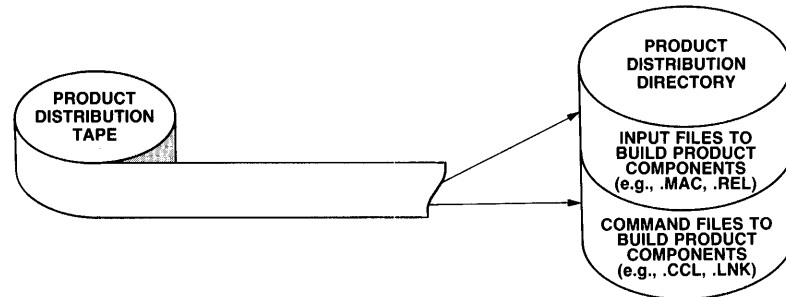
MR-S-955-81

1.2.1.2 Product Installation Directory — Contains all of the executable program (.EXE) files. Most sites already have a directory set up for this purpose. It is one of the directories belonging to SYS:.

1.2.1.3 Autopatching Directory — Receives the files copied from the patch tape. This directory also contains a batch control file (.CTL) for each autopatchable product. This directory serves as a working area where you patch and rebuild the product.

1.2.1.4 Product Backup Directory — Holds backup copies of the input as well as the executable files for the product. When you use PEP to install the patched version of a product, it saves the previous version in this area. (You can generate a single backup directory for all of the autopatchable products at your site, or a separate backup directory for each autopatchable product.)

NOTE

These directories can be built on any of your available disk packs. Although you will need less than 3000 pages (12000 blocks) initially, we recommend that you dedicate an entire disk for the autopatching, backup, and product distribution directories. Thus, all of the disk resources required for autopatching can be kept separate from the directories and files used every day. If you already have directories that can be utilized for autopatching, you may use those instead of creating new ones.

Earlier in this introductory chapter, we have indicated system requirements for autopatching and the qualifications for the operator, and have outlined both the patch tape structure and the functions of the special disk directories. In the next two sections we deal with some special considerations involved in preparing your system for Autopatch, and offer an overview of the activities controlled by the Patch Executive Program (PEP).

1.2.2 Special Considerations: Creating the Directories and Initializing Autopatch

To avoid problems with creating the directories, the order in which you should create them, and what you should call them, we make these recommendations:

1. Dedicate a separate disk pack for the distribution, backup, and auto-patching directories, so that you need to mount only a single disk pack to have everything you need. Our examples refer to this disk structure as DSKP: for TOPS–10 storage, or AP20: for TOPS–20 storage.

2. Of the four needed storage areas, one (which we refer to as the product installation directory) already exists on your system, probably as SYS:

3. Three other storage areas (directories) must be created as part of the Autopatch installation procedure. We show you in Chapter 2, Procedures, how to set up these directories (Section 2.1 for TOPS–10 and Section 2.2 for TOPS–20):

   • Autopatching directory

   • Product distribution directory

   • Product backup directory

NOTE

We urge that you use the naming conventions used in Chapter 2, Procedures, when building your directories, so that your names and those used by both Autopatch and the sample procedures shown in this manual will be compatible. Conforming to the naming conventions will allow you to patch your software products by simply executing the procedures as shown.

4. Product distribution directories must be initialized before you can run Autopatch. To initialize these directories, you will store in them the input files that build each of the products you want to be autopatchable at your site. (You will find a list of the required files for each autopatchable product in INSTAL.DOC, in the installation save set, the second set on the patch tape.)

5. The transition step between installing Autopatch and running it to apply patches is to select the products you want to be autopatchable at your site. You do this by running PEP and issuing a SELECT command for each of the desired products.

1.3 Overview: the Autopatch Process Described

This section describes in general terms what happens in the Autopatching process, that is, how Autopatch works once it has been installed at your site. You initiate these activities by using BACKUP in a TOPS–10 system

and DUMPER in a TOPS–20 system to obtain the necessary files from the patch tape. The activities are then controlled by commands to PEP, the Patch Executive Program.

As we mentioned earlier, AUTOPATCH, the preferable single command to PEP, will carry out the patching process; it handles the activities that RETRIEVE, SETUP and BUILD together handle.

Activity 1. You copy files from the current save set on the patch tape to the autopatching directory so that all the distributed patches for all the autopatchable products are made accessible to Autopatch. (If you are patching a software product for the first time, you will have to copy patches from the cumulative patch save sets also.) See Figure 1–3.

Figure 1–3: Copying the Current Patch Save Set to Disk
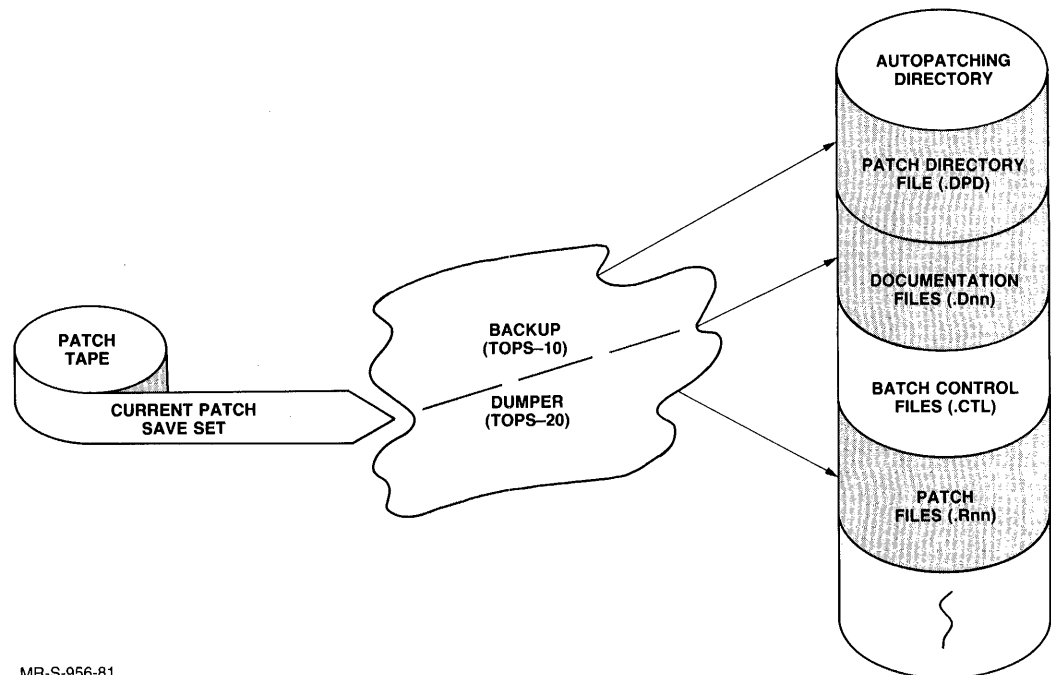
MR-S-956-81

Activity 2. You update the data base in your autopatching directory with information concerning all the new patches to those products that you have previously set up for autopatching. PEP does this activity for you when it processes the RETRIEVE command.

See Figure 1–4, which illustrates activities 2 and 3.

Activity 3. You set up the indirect command files needed to apply those patches that will update your software to the current revision level. PEP does this activity for you when it processes the SETUP command. See Figure 1–4.

Figure 1–4: Retrieving Patch Data and Setting Up the Command Files for Applying Patches
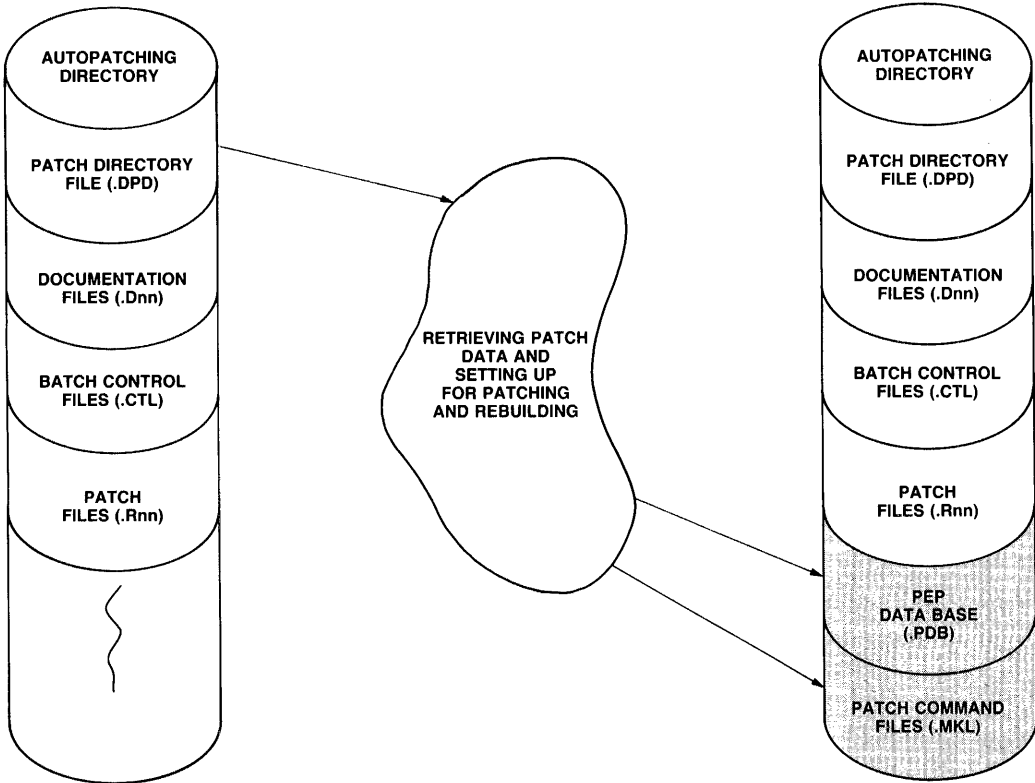
MR-S-957-81

Activity 4.   After you have generated the indirect command files, you are ready to run the batch control file that will apply the patches and rebuild the software components of the product.

This job runs unattended, under batch, and can be scheduled to run at your convenience. However, before you SUBMIT this job to the Galaxy Batch System, PEP must verify that all the required files are in place. PEP performs this activity when it processes the BUILD command.

See Figures 1–5 and 1–6, which illustrate the two phases of activity 4.

Figure 1–5: Batch Processing: Applying the Patches

AUTOPATCHING DIRECTORY

PATCH DIRECTORY FILE (.DPD)

DOCUMENTATION FILES (.Dnn)

BATCH CONTROL FILE (.CTL)

PATCH FILES (.Rnn)

PEP DATA BASE (.PDB)

PATCH COMMAND FILES (.MKL)

PRODUCT DISTRIBUTION DIRECTORY

INPUT FILES TO BUILD PRODUCT COMPONENTS (e.g., .MAC, .REL)

COMMAND FILES TO BUILD PRODUCT COMPONENTS (e.g., .CCL, .LNK)

BATCH PROCESSING

(APPLYING THE PATCHES)

AUTOPATCHING DIRECTORY

PATCH DIRECTORY FILE (.DPD)

DOCUMENTATION FILES (.Dnn)

BATCH CONTROL FILES (.CTL)

PATCH FILES (.Rnn)

PEP DATA BASE (.PDB)

PATCH COMMAND FILES (.MKL)

PATCHED INPUT FILES (.REL)

MR-S-958-81

Figure 1–6: Batch Processing: Rebuilding the Product

AUTOPATCHING DIRECTORY

PATCH DIRECTORY FILE (.DPD)

DOCUMENTATION FILES (.Dnn)

BATCH CONTROL FILE (.CTL)

PATCH FILES (.Rnn)

PEP DATA BASE (.PDB)

PATCH COMMAND FILES (.MKL)

PATCHED INPUT FILES (.REL)

PRODUCT DISTRIBUTION DIRECTORY

INPUT FILES TO BUILD PRODUCT COMPONENTS (e.g., .MAC, .REL)

COMMAND FILES TO BUILD PRODUCT COMPONENTS (e.g., .CCL, .LNK)

BATCH PROCESSING

(REBUILDING THE PRODUCT)

AUTOPATCHING DIRECTORY

PATCH DIRECTORY FILE (.DPD)

DOCMENTATION FILES (.Dnn)

BATCH CONTROL FILES (.CTL)

PATCH FILES (.Rnn)

PEP DATA BASE (.PDB)

PATCH COMMAND FILES (.MKL)

PATCHED INPUT FILES (.REL)

UPDATED PRODUCT FILES (.EXE)

MR-S-959-81

Activity 5.   When the batch job completes, the updated components of the product will be in the autopatching directory. You then copy the updated files to the appropriate directories. When you install the updated version of the software, you are performing two separate operations:

A. Backing up the currently installed versions of both the input and .EXE files of the patched product to the product backup directory.

B. Replacing the currently installed version of the input files in the product distribution directory on disk (superseding the old version with the new version), and replacing the currently installed version of the .EXE file in the product installation directory (SYS:) (superseding the old version with the new one).

See Figure 1–7, which illustrates the two phases of activity 5.

Figure 1–7:   Backing Up and Installing an Updated Software Product

A. Backing Up the Original Version
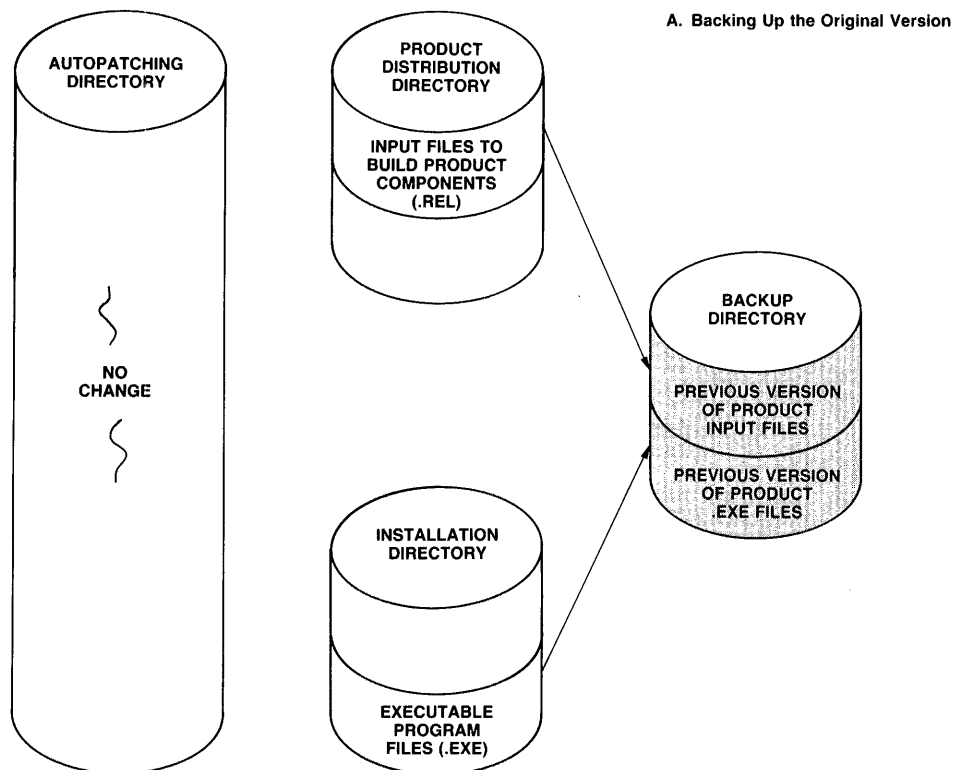
MR-S-960-81
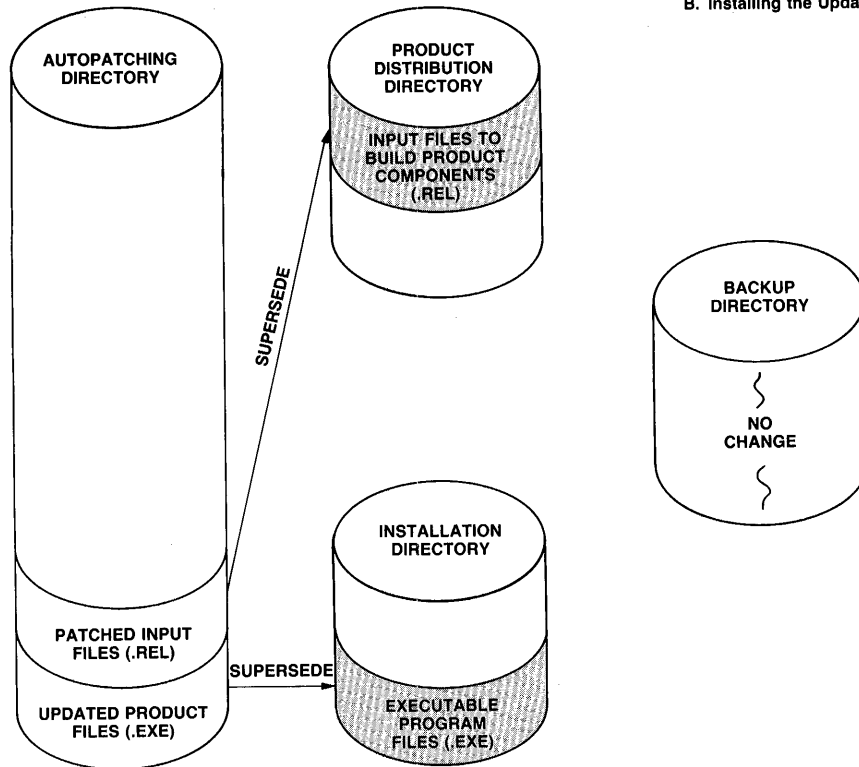
MR-S-961-81

Activity 6.   (Optional) Restoring your software to its original condition (before the product was patched and rebuilt) is the reverse of the install process. This option is provided to allow you to revert to the previous version of a software product if the rebuilt version does not meet your requirements.

See Figure 1–8.

Figure 1–8: Restoring an Installed Software Product

MR-S-962-81

PATCHING A PRODUCT

```
        ( START )
            |
            v
   +------------------+
   | MOUNT            |
   | THE              |
   | PATCH TAPE       |
   +------------------+
            |
            v
   +------------------+
   | COPY             |        Transfer README.nnn and patch
   | DOCUMENTATION    |        retrieval control files to the
   | AND          - - - -      autopatching directory on disk
   | CONTROL FILES    |
   | TO DISK          |
   +------------------+
            |
            v
   +------------------+
   | PRINT            |        Consult README.nnn for index of
   | DOCUMENTATION    |        files and instructions for
   | FILE TO      - - - -      obtaining them
   | USE WITH         |
   | MANUAL           |
   +------------------+
            |
            v
   +------------------+
   |    COPY          |        Transfer patch files from
   |    PATCH     - - - -      patch tape to autopatching
   |    FILES         |        directory
   +------------------+
            |
            v
   +------------------+        Give the AUTOPATCH command to PEP
   | PATCH AND    - - - -
   | REBUILD EACH     |
   | PRODUCT      - - - -      SUBMIT the batch control file that applies
   +------------------+        the patches and rebuilds the product
            |
            v
   +------------------+
   |    TEST          |        Try the update product: does it meet
   |    (optional) - - - -     your needs?
   +------------------+
            |
            v
   +------------------+        Back up your previous version
   | BACKUP       - - - -
   | AND              |
   | INSTALL      - - - -      Install the updated product
   +------------------+
            |
            v
        ( END )
```
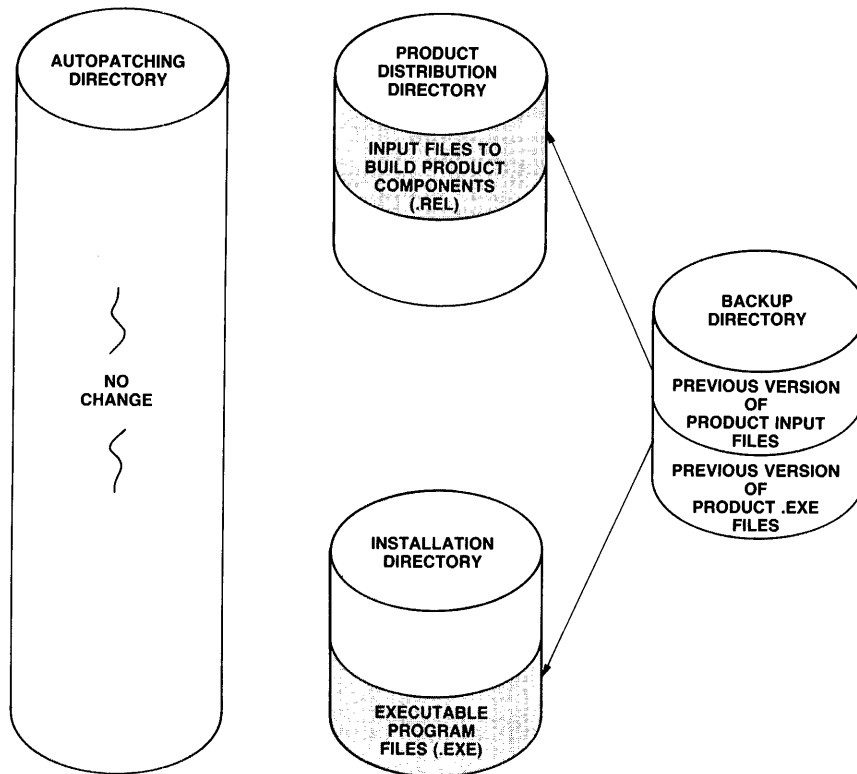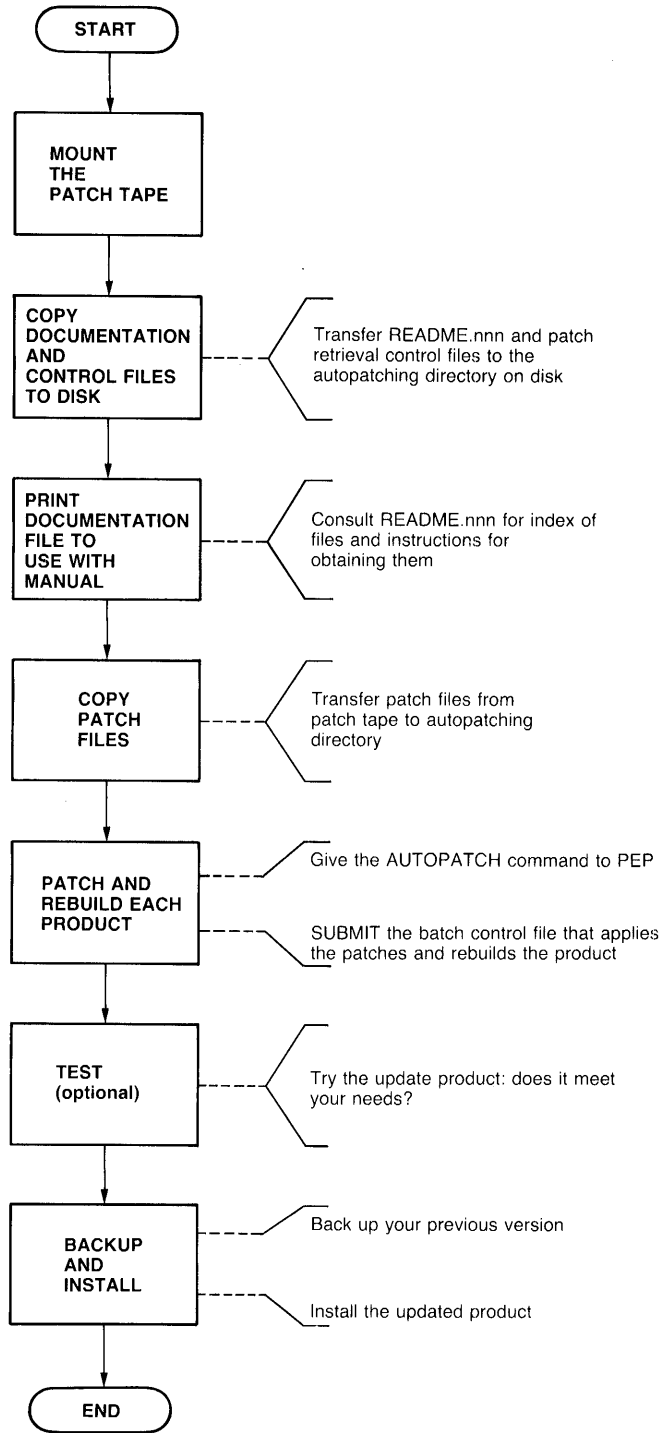
MR-S-964-81

Chapter 2
Procedures

2.1 TOPS–10 Procedures

2.1.1 Installing Autopatch on a TOPS–10 System

In the Introduction, we have given you an overview of the Autopatching concept and of the capabilities of this facility. To prepare your TOPS–10 system for autopatching, follow these step-by-step procedures. The ten steps will take you through the use of the SELECT command to PEP; at that point, Autopatch is installed, and you are ready to apply patches.

Step 1:
LOGIN — Log in under [1,2]

```
.LOGIN
JOB 43      RZ127A KL #1026/1042 TTY141
#1,2
Password:
[LGNJSP Other Jobs same PPN]
08:41    3-Dec-80         Wed

.
```

Step 2:
Mount the structure to be used for all the operations involved in autopatching. To be consistent with our examples, call this structure DSKP:. It should be a dedicated disk so that you can continue to update your system as new patch tapes are distributed.

```
.MOUNT DSKP:/NOWAIT
Request queued to mount units
```

Step 3:
If you already have a PPN established for the autopatching directory, skip this step. Otherwise, use REACT to create the accounting file entries for the autopatching directory. Our examples will reference PPN [7,6] as the

autopatching directory. This involves creating an entry for PPN [7,6] in ACCT.SYS. It also involves creating an entry for the autopatching structure DSKP: in AUXACC.SYS. Note that our example creates three structures in the default search list for this account. The first structure in the list *must* be the autopatching structure. For more information on REACT, see the RUNREA specification in the TOPS–10 Notebook Set and the *TOPS–10 Operator's Guide*, Part V. It is important that you set large permanent and working quotas for this structure, such as 20,000 blocks.

```
.R REACT

For help type "H<CR>"

*R
[793 out of 1100 entries are used]

*T 7,6
List codes?

*I 7,6
Name:AUTOPATCH
Code:PATCH
Privilege word:$
*C 7,6
Change: SPO ALL
Change: LOG ALL
Change:

*T 7,6
List codes?


7    6       AUTOPATCH    000000000000    777777777777
        511,,511 2,,5          017500001763
        8 SEP 768          0    511

*W

*^C

.R REACT

For help type "H<CR>"

*A
[422 out of 1100 entries are used]

*T 7,6

*I 7,6
Num STRs: 3
STR name: DSKP:
Resrvd: FCFS: 20000
Quota out: 20000
Status bits:
STR name: DSKB:
Resrvd: FCFS: 100
Quota out: 100
Status bits:
STR name: DSKC:
Resrvd: FCFS: 100
Quota out: 100
Status bits:

*T 7,6


7    6    DSKP  0    20000   20000   0
          DSKB  0    100     100     0
          DSKC  0    100     100     0
*W

*^C

.
```

Step 4:
Use CREDIR to create a product backup directory, calling it DSKP:[7,6,BACKUP]. This directory is important in case you choose to revert to the previous version after installing the patched software.

```
.R CREDIR
Create directory: DSKP:[7,6,BACKUP]
 Created DSKP0:[7,6],UFD/PROTECTION:775
 Created DSKP0:[7,6,BACKUP],SFD/PROTECTION:775
Create directory: ^Z

.
```

Step 5:
Mount the patch tape we have furnished you.

```
.MOUNT MTA: TAPE:/REELID:PATCH
Request queued
Waiting...2 ^Cs to EXIT

MTA Mounted, MTA101 Used
```

Step 6:
Transfer the file called INSTAL.DOC from the patch tape to the autopatching area, the directory named DSKP:[7,6]. This file is in the second save set (the installation save set) on the tape. Since this installation save set contains tools and control files needed for the autopatching operation, PRINT out INSTAL.DOC and read it to determine which of these files you need.

```
.R BACKUP
/TAPE TAPE:
/INTERCHANGE
/REWIND
/FILES
/SKIP 1
/RESTORE DSKP:INSTAL.DOC[7,6]
! INSTAL        DOC

"Done

/REWIND
/EXIT
.PRINT DSKP:INSTAL.DOC[7,6]
[LPT:INSTAL=/Seq:4602/Limit:57, 1 File]

.
```

Step 7:

As instructed in Section 2 of INSTAL.DOC (Installing Autopatch), transfer from the installation save set of the patch tape the files that are specified. These will include the executable, data, and help files for the Patch Executive Program (PEP). For each product you intend to autopatch, you will need to copy a patch and build control file (.CTL) and a BACKUP distribution control file (.CCL) from this save set to the autopatching directory. The example below shows the files that are necessary for APLSF–10.

```
.R BACKUP

/TAPE TAPE:
/INTERCHANGE
/FILES
/REWIND
/SKIP 1
/RESTORE SYS:=*.EXE,-
#HLP:=*.HLP,-
#DSKP:[7,6]=*.PDF,-
#DSKP:[7,6]=APL1S2.*
! APL1S2 CCL
  APL1S2 CTL
  PEP    EXE
  PEP    HLP
  PEP101 PDF
  PEPB   EXE

"Done
/EXIT

'
```

Step 8:

Build a product distribution directory for each product on your system that you want to patch. Call these directories DSKP:[7,6,product name]. For example, the directory for APLSF–10 would be called DSKP:[7,6,APL] to be consistent with the example.

NOTE

> Remember that the installation of Autopatch is a *one-time* procedure. For each product you intend to patch, however, you need to provide a product distribution directory by repeating Step 8; then move the needed files into it, as in Step 9.

```
.R CREDIR
Create directory: DSKP:[7,6,APL]
  Created DSKP0:[7,6,APL].SFD/PROTECTION:775
Create directory: ^Z

'
```

Step 9:

Again using INSTAL.DOC, transfer the files required for autopatching from the product's distribution tape to the appropriate product distribution area of DSKP:. The details concerning each autopatchable product and the list of appropriate files for each product that you have selected are described in INSTAL.DOC.

Step 10:

With all the elements for autopatching now in place on DSKP:, the next step is to run the PEP program, using the SELECT command to declare

each autopatchable product. Installation is now complete, and the actual autopatching process can begin. (See Section 2.1.2.)

```
.R PEP

[PEP version 1(44)]
What is the autopatching directory? DSKP:[7,6]
 PAT: defined as DSKP:[7,6]
[Initializing ...] [OK]
Now use the SELECT command to define the autopatchable products.

PEP>SELECT (the product) APLSF-10-V2

What is the distribution directory for APLSF-10-V2? DSKP:[7,6,APL]
 DIS: defined as DSKP:[7,6,APL]
What is the backup directory for APLSF-10-V2? DSKP:[7,6,BACKUP]
 BAK: defined as DSKP:[7,6,BACKUP]
What is the installation directory for APLSF-10-V2? SYS:
 INS: defined as SYS:[1,4]
[Checking components and files for APLSF-10-V2] [OK]
[APLSF-10-V2 SELECTed for autopatching]
[Checkpointing] [OK]

PEP>EXIT

.
```

CAUTION

If PEP generates warning messages %CID (Checksum is different) or %VID (Version is different) during this step, you have not set up your product distribution directory or installation directory with the distributed (field-image) version of the product files. Unless you correct this problem, you may not be able to autopatch your software successfully.

2.1.2 Automatic Patching and Rebuilding of a TOPS–10 Product

This section describes the procedures required to do automatic patching on a TOPS–10 configuration. Prior to executing these procedures, you must have completed the Autopatch installation described in Section 2.1.1.

The following procedure assumes that the autopatching directory is on a structure called DSKP:. Log in under [7,6]. If you have established another PPN as your autopatching directory, then log in under that PPN.

Step 1:
Mount the Patch Tape

Mount the patch tape supplied with the Autopatch kit. The procedures shown in our examples use logical name TAPE:.

```
MOUNT MTA:TAPE:/REELID:APATCH
Request queued
Waiting...2^Cs to EXIT
MTA Mounted, MTA101 Used

.
```

Step 2:
Copy the Documentation and Control Files from the Patch Tape to Disk

Transfer the documentation and patch retrieval control files from the current patch save set (see Section 1.1.4.1) to the autopatching directory on

disk. These files will include the latest copy of README as well as a patch retrieval control file for each autopatchable product. You can later use these patch retrieval control files to obtain the patch files from the tape. There are two types of patch retrieval control files. Those with file type .Pnn can be used to obtain patch files from the current save set for a specific product. Those with file type .Ann can be used to obtain the patch files from *all* the save sets for a specific product.

NOTE

> If you have a large amount of disk space available in the autopatching directory, you may elect to transfer the entire contents of the current patch save set to disk at this time. Although this could consume a large amount of disk space, it will simplify the tape-to-disk transfer operation. You can then delete the files you do not need. However, your operations then will not correspond to our example.

When BACKUP is ready to accept commands, it types a slash (/). Enter the following commands in response to the BACKUP prompts:

```
@R BACKUP
/TAPE TAPE:
/INTERCHANGE
/FILES
/REWIND
/RESTORE README.*,-
#*.P*
! APL1B2    P01
  APL1S2 P01
  README 001
/EXIT

@
```

Step 3:
Print the Documentation File

At this point, print out the disk file README.nnn, where nnn is the number of the current patch tape.

You need to read this documentation file because it contains the index of the various files you will use and the instructions for obtaining them from the patch tape.

```
@PRINT README.001
[LPT:README=/Seq:4603/Limit:54, 1 File]

.
```

Step 4:
Copy the Patch Files for the Products to be Autopatched from the Patch Tape to Disk

Follow the instructions in the README documentation file so that you are ready to do autopatching. After obtaining the patch files, DISMOUNT the patch tape.

Step 5:
Patch and Rebuild the Software Product(s)

To prepare a product for the patch and rebuild operation, run PEP and issue the AUTOPATCH command.

The AUTOPATCH command to PEP performs the following functions:

- updates the PEP data base with a record of the latest patches.

- generates the command files that the batch control file needs to apply the patches to the product you are autopatching.

- verifies that all of the required files are present and that they are at the required revision level.

NOTE

You can use the command AUTOPATCH * to prepare all your SELECTed products.

Following is a print-out of what you would see on your terminal if, for example, you were patching APLSF–10–V2:

```
.R PEP

[PEP version 1(44)]

What is the autopatching directory? DSKP:[7,6]
 PAT: defined as DSKP:[7,6]
[Initializing ...] (Reading data base ... ) [OK]

PEP>AUTOPATCH (product name) APLSF-10-V2

[Processing information from patch tape 1]
[Reading patch directory file] [OK]
[23 patches retrieved for APLSF-10-V2]

[Checkpointing] [OK]

[Setting up APLSF-10-V2 for patch and rebuild] [OK]
[23 patches will be applied to APLSF-10-V2]
[Checkpointing] [OK]

[Checking files for building APLSF-10-V2] [OK]

[Checkpointing] [OK]

[To patch and rebuild APLSF-10-V2, issue the monitor command ...]
 SUBMIT DSKP:APL1S2.CTL[7,6]/TIME:1:00:00/RESTART

PEP>EXIT

.
```

After PEP verifies that the product is set up for patching and rebuilding, SUBMIT its batch control file to the Galaxy Batch System. You may want to supply additional parameters to schedule the running of this job. Consult the *TOPS–10 Operating System Commands Manual* to determine those parameters.

```
.SUBMIT DSKP:APL1S2.CTL[7,6]/TIME:1:00:00/RESTART

[INP:APL1S2=/Seq:4604/TIME:1:00:00]
```

NOTE

When the batch job completes, you will have a log file. It is a good idea at this point to compare checksums in your log file with those in the log file (.Lnn) furnished with the patch tape. If they match, fine — but if they differ, put the two listings side by side to determine exactly where the differences exist.

If you get a message such as "version number different" or "checksums different," this will point to possible sources of error.

Following is the print-out of what you would see on your terminal as you check to determine whether the rebuilt product (in this case, APLSF–10–V2) is ready to be installed; use the INFORMATION command to PEP.

```
.R PEP

[PEP version 1(44)]

What is the autopatching directory?DSKP:[7,6]
 PAT: defined as DSKP:[7,6]
[Initializing ...] (Reading data base ... ) [OK]
[Checking results of batch job to rebuild APLSF-10-V2]
 [Completed successfully]
[Checkpointing] [OK]

PEP>INFORMATION (about product name) APLSF-10-V2

APLSF-10-V2 is ready to be INSTALLed
 Patches retrieved through patch tape 1

 Current installed files:
 DSKC:APLSF.EXE[1,4] sequential checksum 724333, version 2(435)
 DSKC:GENSF.EXE[1,4] sequential checksum 226667, version 2(12)
 DSKC:TENTAP.EXE[1,4] sequential checksum 430235, version 1(3)

 Updated files (ready to be installed):
 DSKP:APLSF.EXE[7,6] sequential checksum 750467, version 2(464)
 DSKP:GENSF.EXE[7,6] sequential checksum 226667, version 2(12)
 DSKP:TENTAP.EXE[7,6] sequential checksum 430235, version 1(3)

 Backup files:
 BAK:APLSF.EXE (File not found)
 BAK:GENSF.EXE (File not found)
 BAK:TENTAP.EXE (File not found)

 Logical names:
 PAT: (autopatching directory) defined as DSKP:[7,6]
 DIS: (distribution directory) defined as DSKP:[7,6,APL]
 BAK: (backup directory) defined as DSKP:[7,6,BACKUP]
 INS: (installation directory) defined as SYS:[1,4]
 ASL: (autopatch serach list) defined as DSKP:[7,6],DSKP:[7,6,APL],SYS:[1,4]
```

Step 6:
Testing the Rebuilt Software

Once the batch job completes successfully, the rebuilt software is available in the autopatching directory and is ready to be installed in the system area. Before installing it, however, you may want to try running it out of the autopatching directory to determine whether it performs to your satisfaction.

If the rebuilt software, after you have run some of your most important applications with it, does not meet your expectations, do *not* go on to Step 7, Installing the Rebuilt Software.

Step 7:
Installing the Rebuilt Software

As you install the rebuilt software, you will be copying files into the PPN defined to be the installation directory for this product. If you are using SYS: or another protected PPN as the installation directory, you will need to be logged in under [1,2].

<div align="center">

CAUTION

</div>

> If the installation operation is actually going to change files in the SYS: area, you must be careful to avoid interfering with the work others are doing if your system is running under timesharing.

Following is the print-out of what you would see on your terminal as you completed the installation of the rebuilt APLSF–10–V2 using the INSTALL command, and rechecked on the success of the installation by asking PEP a second time for INFORMATION:

```
PEP>INSTALL (product name) APLSF-10-V2

[Checking files for installing APLSF-10-V2] [OK]
[Backing up files for APLSF-10-V2]
 DSKP:APLG12.REL[7,6,APL] => DSKP:APLG12.REL[7,6,BACKUP] [OK]
 DSKP:APLS12.REL[7,6,APL] => DSKP:APLS12.REL[7,6,BACKUP] [OK]
 DSKP:APLW12.REL[7,6,APL] => DSKP:APLW12.REL[7,6,BACKUP] [OK]
 DSKC:APLSF.EXE[1,4] => DSKP:APLSF.EXE[7,6,BACKUP] [OK]
 DSKC:GENSF.EXE[1,4] => DSKP:GENSF.EXE[7,6,BACKUP] [OK]
 DSKC:TENTAP.EXE[1,4] => DSKP:TENTAP.EXE[7,6,BACKUP] [OK]
[Checkpointing] [OK]
[Installing new files for APLSF-10-V2]
 DSKP:APLG12.REL[7,6] => DSKP:APLG12.REL[7,6,APL] [OK]
 DSKP:APLS12.REL[7,6] => DSKP:APLS12.REL[7,6,APL] [OK]
 DSKP:APLW12.REL[7,6] => DSKP:APLW12.REL[7,6,APL] [OK]
 DSKP:APLSF.EXE[7,6] => DSKC:APLSF.EXE[1,4] [OK]
 DSKP:GENSF.EXE[7,6] => DSKC:GENSF.EXE[1,4] [OK]
 DSKP:TENTAP.EXE[7,6] => DSKC:TENTAP.EXE[1,4] [OK]
[Checkpointing] [OK]

PEP>INFORMATION (about product name) APLSF-10-V2

APLSF-10-V2 was successfully INSTALLed
 Patches retrieved through patch tape 1

 Current installed files:
 DSKC:APLSF.EXE[1,4] sequential checksum 750467, version 2A(464)
 DSKC:GENSF.EXE[1,4] sequential checksum 226667, version 2(12)
 DSKC:TENTAP.EXE[1,4] sequential checksum 430235, version 1(3)

 Updated files (ready to be installed):
 None

 Backup files:
 DSKP:APLSF.EXE[7,6,BACKUP] sequential checksum 724333, version 2(435)
 DSKP:GENSF.EXE[7,6,BACKUP] sequential checksum 226667, version 2(12)
 DSKP:TENTAP.EXE[7,6,BACKUP] sequential checksum 430235, version 1(3)

 Logical names:
 PAT: (autopatching directory) defined as DSKP:[7,6]
 DIS: (distribution directory) defined as DSKP:[7,6,APL]
 BAK: (backup directory) defined as DSKP:[7,6,BACKUP]
 INS: (installation directory) defined as SYS:[1,4]
 ASL: (autopatch search list) defined as DSKP:[7,6],DSKP:[7,6,APL],SYS:[1,4]

PEP>EXIT

*
```

A successful installation accomplishes three tasks:

- It establishes a backup copy of the current version of the product you are patching.

- It moves the rebuilt product into the installation directory.

- It moves the patched input files into the product's distribution directory.

NOTE

If the installation procedure fails, however, because of a system failure, or because of an error such as "disk quota exceeded," you will need to correct the error and repeat the INSTALL command.

2.1.3 Restoring an Installed TOPS–10 Product

If, for some reason, an installed software product does not perform to your expectations and the previous version is more in keeping with your particular requirements, you can revert to the previous version by using the RESTORE command to PEP:

```
.R PEP
[PEP version 1(44)]
What is the autopatching directory? DSKP:[7,6]
 PAT: defined as DSKP:[7,6]
[Initializing ...] (Reading data base ... ) [OK]

PEP>RESTORE (product name) APLSF-10-V2

[Checking files for restoring APLSF-10-V2] [OK]
[Checkpointing] [OK]
[Restoring files for APLSF-10-V2]
 DSKP:APLG12.REL[7,6,BACKUP] => DSKP:APLG12.REL[7,6,APL] [OK]
 DSKP:APLS12.REL[7,6,BACKUP] => DSKP:APLS12.REL[7,6,APL] [OK]
 DSKP:APLW12.REL[7,6,BACKUP] => DSKP:APLW12.REL[7,6,APL] [OK]
 DSKP:APLSF.EXE[7,6,BACKUP] => DSKC:APLSF.EXE[1,4] [OK]
 DSKP:GENSF.EXE[7,6,BACKUP] => DSKC:GENSF.EXE[1,4] [OK]
 DSKP:TENTAP.EXE[7,6,BACKUP] => DSKC:TENTAP.EXE[1,4] [OK]
[Checkpointing] [OK]

PEP>EXIT

.
```

See Section 3.1.1 for a complete description of the RESTORE command.

2.2 TOPS–20 Procedures

2.2.1 Installing Autopatch on a TOPS–20 System

In the Introduction, we have given you an overview of the Autopatching concept and of the capabilities of this facility. To prepare your TOPS–20 system for autopatching, follow the procedures below. The ten steps will take you through the use of the SELECT command to PEP; at that point, Autopatch is installed, and you are ready to apply patches.

Step 1:

LOGIN — You may log in to any account having WHEEL or OPERATOR access privileges; ENABLE these privileges.

```
@LOGIN (USER) WHEEL (PASSWORD)
 Job 57 on TTY21 10-Dec-80 10:16:49
@ENABLE (CAPABILITIES)

$
```

Step 2:

Prepare a disk structure to be used for all the operations involved in auto-patching. To be consistent with our examples, call this structure AP20:. It should be a dedicated disk so that you can continue to update your system as new patch tapes are distributed. Edit your PS:<SYSTEM>MOUNTR.CMD file to declare this a domestic structure. If you plan to use an existing domestic structure for autopatching, you can skip this step.

```
$CONNECT (TO DIRECTORY) SYSTEM:
$TV MOUNTR.CMD

*
INPUT FILE: MOUNTR.CMD.6
462 CHARS
*ZJ$$
*IDOMESTIC (STRUCTURE) AP20:
$$
* ;X$$
OUTPUT FILE: MOUNTR.CMD.7 !New generation!

$TYPE (FILE) MOUNTR.CMD
!MOUNTR COMMAND FILE
DOMESTIC (STRUCTURE) CBL79:
DOMESTIC (STRUCTURE) SUPT:
DOMESTIC (STRUCTURE) AP20:

$
```

Step 3:

Use the BUILD command to create a login directory on PS:. Call this new directory PS:<AUTOPATCH>. It is important that you set up this directory with LOGIN and WHEEL privileges. In this directory you create a BATCH.CMD file for running the patch and build batch jobs. BATCH.CMD should contain commands to MOUNT the autopatching structure and ACCESS the autopatching directory on that structure.

```
$CONNECT (TO DIRECTORY) <WHEEL>
$BUILD PS:<AUTOPATCH> PATCH
[New]
$$WHEEL
$$
$CONNECT (TO DIRECTORY) PS:<AUTOPATCH>
$CREATE BATCH.CMD
Input: BATCH.CMD.1
00100 MOUNT STRUCTURE AP20:
00200 ACCESS AP20:<AUTOPATCH>
00300 $
*EU

[BATCH.CMD.1]

$
```

At this point you will need to MOUNT the autopatching structure, AP20:, to BUILD an <AUTOPATCH> directory on it. AP20:<AUTOPATCH> will be the disk area referred to as your autopatching directory. Set the parameters for this directory to allow for 20 subdirectories and working and permanent quotas of 20,000 pages, and supply a user-group number for it and its subdirectories.

```
$MOUNT STRUCTURE AP20:
Structure AP20: mounted
$INFORMATION (ABOUT) STRUCTURE (NAME) AP20:
Status of structure AP20:
Mount count: 1, open file count: 0, units in structure: 1
Domestic
Users who have MOUNTed AP20: WHEEL
No users are ACCESSing AP20:
No users CONNECTed to AP20:
$BUILD AP20:<AUTOPATCH>
[New]
$$PERMANENT (DISK STORAGE PAGE LIMIT) 20000
$$WORKING (DISK STORAGE PAGE LIMIT) 20000
$$MAXIMUM-SUBDIRECTORIES (ALLOWED) 20
$$USER-GROUP (NUMBER) 10
$$SUBDIRECTORY-USER-GROUP (ALLOWED) 10
$$

$
```

Step 4:
Use BUILD to create your product backup subdirectory, calling it <AUTOPATCH.BACKUP>. This directory is important in case you wish to revert to the previous version after installing the patched software. You will want to set a working quota of 4,000 pages and a permanent quota of 3,000 pages. Assign a directory-group number that is the same as the user-group number of the autopatching directory.

```
$BUILD AP20:<AUTOPATCH.BACKUP>
[New]
$$WORKING (DISK STORAGE PAGE LIMIT) 4000
$$PERMANENT (DISK STORAGE PAGE LIMIT) 3000
$$DIRECTORY-GROUP (NUMBER) 10
$$

$
```

Step 5:
CONNECT to the autopatching directory and MOUNT the patch tape we have furnished you.

```
$CONNECT (TO DIRECTORY) AP20:<AUTOPATCH>
$MOUNT TAPE TAPE:
[Mount Request TAPE Queued, Request-ID 99]
[Tape set TAPE, volume TAPE mounted]

$
```

Step 6:
Transfer the file called INSTAL.DOC from the patch tape to the autopatching directory. This file is in the second save set (the installation save set) on the tape. Since this installation save set contains tools and control files needed for the autopatching operation, PRINT out INSTAL.DOC and read it to determine which of these files you need.

```
$DUMPER
DUMPER>TAPE TAPE:
DUMPER>REWIND
DUMPER>FILES
DUMPER>SKIP 1

DUMPER tape # 1  Volid TAPE, AUTOPATCH SET 002, Monday, 10-Nov-80 1003

DUMPER tape # 1  Volid TAPE, AUTOPATCH INSTALLATION SAVE SET, Monday, 10-Nov-80 1005
DUMPER>RESTORE DSK*:<*>INSTAL.DOC DSK:

DUMPER tape # 1  Volid TAPE, AUTOPATCH INSTALLATION SAVE SET, Monday, 10-Nov-80 1005
Loading file(s) into AP20:<AUTOPATCH>
AP20:<AUTOPATCH>INSTAL.DOC.2;P777700;A (TO) INSTAL.DOC.2      [OK]

End of saveset
DUMPER>REWIND
DUMPER>EXIT
$PRINT INSTAL.DOC
[Job INSTAL Queued, Request-ID 38, Limit 36]

$
```

Step 7:

As instructed in Section 2 of INSTAL.DOC (Installing Autopatch), transfer from the installation save set of the patch tape the files that are specified.

These will include the executable, data, and help files for the Patch Executive Program (PEP). For each product you intend to autopatch, you will need to copy a patch and build control file (.CTL) and a DUMPER distribution control file (.DDC) from this save set to the autopatching directory. The example below shows the files that are necessary for the 2060–MAXIMUM configuration of the TOPS–20 monitor.

```
$DUMPER
DUMPER>TAPE TAPE:
DUMPER>REWIND
DUMPER>FILES
DUMPER>SKIP 1

DUMPER tape # 1  Volid TAPE, AUTOPATCH SET 002, Monday, 10-Nov-80 1003

DUMPER tape # 1  Volid TAPE, AUTOPATCH INSTALLATION SAVE SET, Monday, 10-Nov-80 1005
DUMPER>RESTORE DSK*:<*>*.EXE SYS:,-
DSK*:<*>*.HLP HLP:,-
DSK*:<*>*.PDF DSK:,-
DSK*:<*>T20604.CTL DSK:,-
DSK*:<*>T20604-MAX.DDC DSK:

DUMPER tape # 1  Volid TAPE, AUTOPATCH INSTALLATION SAVE SET, Monday, 10-Nov-80 1005
Loading file(s) into PS:<SUBSYS>
AP20:<AUTOPATCH>PEP.EXE.2;P777752;A (TO) PS:<SUBSYS>PEP.EXE.2   [OK]
Loading file(s) into PS:<HELP>
AP20:<AUTOPATCH>PEP.HLP.2;P777752;A (TO) PS:<HLP>PEP.HLP.2      [OK]
Loading file(s) into AP20:<AUTOPATCH>
AP20:<AUTOPATCH>PEP201.PDF.1;P777700;A (TO) PEP201.PDF.1       [OK]
Loading file(s) into PS:<SUBSYS>
AP20:<AUTOPATCH>PEPB.EXE.2;P777752;A (TO) PS:<SUBSYS>PEPB.EXE.2[OK]
Loading file(s) into AP20:<AUTOPATCH>
AP20:<AUTOPATCH>T20604.CTL.1;P777700;A (TO) T20604.CTL.1       [OK]
AP20:<AUTOPATCH>T20604-MAX.DDC.1;P777700;A (TO) T20604-MAX.DDC.1[OK]

End of saveset
DUMPER>REWIND
DUMPER>EXIT

$
```

Step 8:
Build a product distribution directory for each product on your system that you intend to autopatch. Call these directories <AUTOPATCH.product name>. For example, the directory for the TOPS–20 monitor would be called <AUTOPATCH.TOPS–20>. For each directory, set up working and permanent quotas of 3000 pages, and assign a directory-group number that is the same as the user-group number you assigned to the autopatching directory.

NOTE

Remember that the installation of Autopatch is a *one-time* procedure. For each product you intend to patch, however, you need to provide a product distribution directory by repeating Step 8; then move the needed files into it, as in Step 9.

```
$BUILD AP20:<AUTOPATCH.TOPS-20>
[New]
$$WORKING (DISK STORAGE PAGE LIMIT) 3000
$$PERMANENT (DISK STORAGE PAGE LIMIT) 3000
$$DIRECTORY-GROUP (NUMBER) 10
$$

$
```

Step 9:
Again using INSTAL.DOC, transfer the files required for autopatching from the product's distribution tape to the appropriate product distribution area of AP20:. The details concerning each autopatchable product and the list of appropriate files for each product that you have selected are described in INSTAL.DOC.

Step 10:
With all the elements for autopatching now in place on AP20:, the next step is to run the PEP program, using the SELECT command to declare each autopatchable product. Installation is now complete, and the actual auto-patching process can begin. (See Section 2.2.2, the section following this one.)

```
$PEP
[PEP version 1(44)]
What is the autopatching directory? AP20:<AUTOPATCH>
 PAT: defined as AP20:<AUTOPATCH>
[Initializing ...] [OK]

Now use the SELECT command to define the autopatchable products.

PEP>SELECT TOPS-20-2060-V4
What is the distribution directory for TOPS-20-2060-V4? AP20:<AUTOPATCH.TOPS-20>
 DIS: defined as AP20:<AUTOPATCH.TOPS-20>
What is the backup directory for TOPS-20-2060-V4? AP20:<AUTOPATCH.BACKUP>
 BAK: defined as AP20:<AUTOPATCH.BACKUP>
What is the installation directory for TOPS-20-2060-V4? PS:<SYSTEM>
 INS: defined as PS:<SYSTEM>
[Checking components and files for TOPS-20-2060-V4] [OK]
[TOPS-20-2060-V4 SELECTed for autopatching]
[Checkpointing] [OK]

PEP>EXIT

$
```

CAUTION

If PEP generates warning messages %CID (Checksum is different) or %VID (Version is different) during this step, you have not set up your product distribution directory or installation directory with the distributed (field-image) version of the product files. Unless you correct this problem, you may not be able to autopatch your software successfully.

2.2.2 Automatic Patching and Rebuilding of a TOPS–20 Product

This section describes the procedures required to do automatic patching on a TOPS–20 configuration. Prior to executing these procedures, you must have completed the Autopatch installation described in Section 2.2.1.

TOPS–20 Login Procedures — Autopatch requires that the autopatching directory be the connected directory. If you built the subdirectories as described in Section 2.2.1, you will also have to obtain group ownership rights for this directory.

The following procedure assumes that the autopatching directory is on a mountable structure called AP20:. After logging in, you mount the structure, obtain the group ownership rights, and connect to the autopatching directory.

For example:

```
@LOGIN (USER) AUTOPATCH (PASSWORD)
 Job 7 on TTY42 12-Dec-80 18:28:27
@MOUNT STRUCTURE (NAME) AP20:
Structure AP20: mounted
@ACCESS (TO DIRECTORY) AP20:<AUTOPATCH>
@CONNECT (TO DIRECTORY) AP20:<AUTOPATCH>

@
```

Step 1:
Mount the Patch Tape

Mount the patch tape supplied with the Autopatch kit. The procedures shown in our examples use logical name TAPE:.

```
@MOUNT TAPE (NAME) TAPE:
[Mount Request TAPE Queued, Request-ID 18]
[Tape set TAPE, volume TAPE mounted]
[TAPE: defined as MT0:]

@
```

Step 2:
Copy the Documentation and Control Files from the Patch Tape to Disk

Transfer the documentation and patch retrieval control files from the current patch save set (see Section 1.1.4.1) to the autopatching directory on disk. These files will include the latest copy of README as well as a patch retrieval control file for each autopatchable product. You can later use these patch retrieval control files to obtain the patch files from the tape. There are two types of patch retrieval control files. Those with file type .Pnn can be used to obtain patch files from the current save set for a specific product. Those with file type .Ann can be used to obtain the patch files from *all* the save sets for a specific product.

If you have a large amount of disk space available in the autopatching directory, you may elect to transfer the entire contents of the current patch save set to disk at this time. Although this could consume a large amount of disk space, it will simplify the tape-to-disk transfer operation. You can then delete the files you do not need. However, your operations then will not correspond to our example.

When DUMPER is ready to accept commands, it types DUMPER>. Enter the following commands in response to the DUMPER prompts:

```
@DUMPER
DUMPER>TAPE TAPE:
DUMPER>REWIND
DUMPER>FILES
DUMPER>RESTORE DSK*:<*>README.* DSK:,-
DSK*:<*>*.P* DSK:,-
DSK*:<*>*.A* DSK:

DUMPER tape # 1  Volid TAPE, AUTOPATCH SET 002, Monday, 10-Nov-80 1003
Loading file(s) into AP20:<AUTOPATCH>
AP20:<AUTOPATCH>APL2B2.A02.1;P777700;A (TO) APL2B2.A02.1    [OK]
AP20:<AUTOPATCH>APL2B2.P02.1;P777700;A (TO) APL2B2.P02.1    [OK]
AP20:<AUTOPATCH>APL2S2.A02.1;P777700;A (TO) APL2S2.A02.1    [OK]
AP20:<AUTOPATCH>APL2S2.P02.1;P777700;A (TO) APL2S2.P02.1    [OK]
AP20:<AUTOPATCH>README.002.1;P777700;A (TO) README.002.1    [OK]
AP20:<AUTOPATCH>T20204.A02.1;P777700;A (TO) T20204.A02.1    [OK]
AP20:<AUTOPATCH>T20204.P02.1;P777700;A (TO) T20204.P02.1    [OK]
AP20:<AUTOPATCH>T20404.A02.1;P777700;A (TO)'T20404.A02.1    [OK]
AP20:<AUTOPATCH>T20404.P02.1;P777700;A (TO) T20404.P02.1    [OK]
AP20:<AUTOPATCH>T20604.A02.1;P777700;A (TO) T20604.A02.1    [OK]
AP20:<AUTOPATCH>T20604.P02.1;P777700;A (TO) T20604.P02.1    [OK]
AP20:<AUTOPATCH>T20AL4.A02.1;P777700;A (TO) T20AL4.A02.1    [OK]
AP20:<AUTOPATCH>T20AL4.P02.1;P777700;A (TO) T20AL4.P02.1    [OK]
AP20:<AUTOPATCH>T20AS4.A02.1;P777700;A (TO) T20AS4.A02.1    [OK]
AP20:<AUTOPATCH>T20AS4.P02.1;P777700;A (TO) T20AS4.P02.1    [OK]

End of saveset
DUMPER>EXIT

@
```

Step 3:
Print the Documentation File

At this point, print out the disk file README.nnn, where nnn is the number of the current patch tape.

You need to read this documentation file because it contains the index of the various files you will use and the instructions for obtaining the various patch files from the patch tape.

```
@PRINT (FILES) README.002
[Job README Queued, Request-ID 20, Limit 27]

@
```

Step 4:
Copy the Patch Files for the Products to be Autopatched from the Patch Tape to Disk

After obtaining the patch files, DISMOUNT the patch tape.

```
@DISMOUNT TAPE TAPE:
[Tape dismounted, logical name TAPE: deleted]

@
```

Step 5:
Patch and Rebuild the Software Product(s)

To prepare a product for the patch and rebuild operation, run PEP and issue the AUTOPATCH command.

The AUTOPATCH command to PEP performs the following functions:

- updates the PEP data base with a record of the latest patches.

- generates the command files that the batch control file needs to apply the patches to the product you are autopatching.

- verifies that all of the required files are present and that they are at the required revision level.

NOTE

You can use the command AUTOPATCH * to prepare all your SELECTed products.

Following is a print-out of what you would see on your terminal if, for example, you were patching TOPS–20–2060–V4:

```
@PEP
[PEP version 1(44)]
What is the autopatching directory? AP20:<AUTOPATCH>
 PAT: defined as AP20:<AUTOPATCH>
[Initializing ...] (Reading data base ...) [OK]

PEP>AUTOPATCH (product name) TOPS-20-2060-V4

[Processing information from patch tape 1]
[Reading patch directory file] [OK]
[32 patches retrieved for TOPS-20-2060-V4]

[Processing information from patch tape 2]
[Reading patch directory file] [OK]
[41 patches retrieved for TOPS-20-2060-V4]

[Checkpointing] [OK]

[Setting up TOPS-20-2060-V4 for patch and rebuild [OK]
[73 patches will be applied to TOPS-20-2060-V4]
[Checkpointing] [OK]

[Checking files for building TOPS-20-2060-V4] [OK]
[To patch and rebuild TOPS-20-2060-V4, issue the monitor command ...]
 SUBMIT AP20:<AUTOPATCH>T20604.CTL.1/TIME:1:00:00/RESTART
[Checkpointing] [OK]

PEP>EXIT

@
```

After PEP verifies that the product is set up for patching and rebuilding, SUBMIT its batch control file to the Galaxy Batch System. You may want to supply additional parameters to schedule the running of this job. Consult the *TOPS–20 Commands Reference Manual* to determine those parameters.

```
@SUBMIT (BATCH JOB) T20604.CTL.1/TIME:1:00:00/RESTART

[Job T20604 Queued, Request-ID 23, Limit 1:00:00]
```

NOTE

> When the batch job completes, you will have a log file. It is a
> good idea at this point to compare checksums in your log file
> with those in the log file (file type .Lnn) furnished with the
> patch tape. If they match, fine — but if they differ, put the
> two listings side by side to determine exactly where the dif-
> ferences exist.
>
> If you get a message such as " version number different" or
> "checksums different," this will point to possible sources of
> error.

Following is a print-out of what you would see on your terminal as you
check to determine whether the rebuilt product (in this case,
TOPS–20–2060–V4) is ready to be installed; use the INFORMATION com-
mand to PEP.

```
@ENABLE (CAPABILITIES)
$PEP
[PEP version 1(44)]
[PAT: (autopatching directory) defined as AP20:<AUTOPATCH>]
[Initializing ...] (Reading data base ... ) [OK]
[Checking results of batch job to rebuild TOPS-20-2060-V4)
 [Completed successfully]
[Checkpointing] [OK]

PEP>INFORMATION (about product name) TOPS-20-2060-V4

TOPS-20-2060-V4 is ready to be INSTALLed
 Patches retrieved through patch tape 2

 Current installed files:
 PS:<SYSTEM>EXEC.EXE.1 sequential checksum 765667, version 4(560)
 PS:<SYSTEM>BUGSTRINGS.TXT.1 sequential checksum 511220
 PS:<SYSTEM>MONITR.EXE.1 sequential checksum 647550, version 4(3247)

 Updated files (ready to be installed):
 AP20:<AUTOPATCH>EXEC.EXE.1 sequential checksum 737352, version 4(567)
 AP20:<AUTOPATCH>BUGSTRINGS.TXT.1 sequential checksum 216046
 AP20:<AUTOPATCH>MONITR.EXE.1 sequential checksum 60334, version 4(3370)

 Backup files:
 BAK:EXEC.EXE (File not found)
 BAK:BUGSTRINGS.TXT (File not found)
 BAK:MONITR.EXE (File not found)

 Logical names:
 PAT: (autopatching directory) defined as AP20:<AUTOPATCH>
 DIS: (distribution directory) defined as AP20:<AUTOPATCH.TOPS-20>
 BAK: (backup directory) defined as AP20:<AUTOPATCH.BACKUP>
 INS: (installation directory) defined as PS:<SYSTEM>
 ASL: (autopatch search list) defined as PAT:,DIS:,SYS:

PEP>
```

Step 6:
Testing the Rebuilt Software

Once the batch job completes successfully, the rebuilt software is available
in the autopatching directory and is ready to be installed in the system
area. Before installing it, however, you may want to try running it out of
the autopatching directory to determine whether it performs to your
satisfaction.

If the rebuilt software, after you have run some of your most important applications with it, does not meet your expectations, do *not* go on to Step 7, Installing the Rebuilt Software.

Step 7:
Installing the Rebuilt Software

As you install the rebuilt software, you will be copying files into the directory defined to be the installation directory for this product. If you are using SYS: or another protected directory as your installation directory, you will need to have your WHEEL or OPERATOR privileges enabled.

CAUTION

If the installation operation is actually going to change files in the SYS: area, you must be careful to avoid interfering with the work others are doing if your system is running under timesharing.

Following is a print-out of what you would see on your terminal as you completed the installation of the rebuilt TOPS–20–2060–V4 with the command INSTALL, and rechecked on the success of the installation by asking PEP a second time for INFORMATION:

```
PEP>INSTALL (product name) TOPS-20-2060-V4

[Checking files for installing TOPS-20-2060-V4] [OK]
[Backing up files for TOPS-20-2060-V4]
 AP20:<AUTOPATCH.TOPS-20>T20GM4.REL.1 => AP20:<AUTOPATCH.BACKUP>T20GM4.REL.1 [OK]
 AP20:<AUTOPATCH.TOPS-20>T20EX4.REL.1 => AP20:<AUTOPATCH.BACKUP>T20EX4.REL.1 [OK]
 PS:<SYSTEM>EXEC.EXE.3 => AP20:<AUTOPATCH.BACKUP>EXEC.EXE.1 [OK]
 PS:<SYSTEM>BUGSTRINGS.TXT.1 => AP20:<AUTOPATCH.BACKUP>BUGSTRINGS.TXT.1 [OK]
 PS:<SYSTEM>MONITR.EXE.1 => AP20:<AUTOPATCH.BACKUP>MONITR.EXE.1 [OK]
[Checkpointing] [OK]
[Installing new files for TOPS-20-2060-V4]
 AP20:<AUTOPATCH>T20GM4.REL.40 => AP20:<AUTOPATCH.TOPS-20>T20GM4.REL.2 [OK]
 AP20:<AUTOPATCH>T20EX4.REL.9 => AP20:<AUTOPATCH.TOPS-20>T20EX4.REL.2 [OK]
 AP20:<AUTOPATCH>EXEC.EXE.1 => PS:<SYSTEM>EXEC.EXE.2 [OK]
 AP20:<AUTOPATCH>BUGSTRINGS.TXT.1 => PS:<SYSTEM>BUGSTRINGS.TXT.2 [OK]
 AP20:<AUTOPATCH>MONITR.EXE.1 => PS:<SYSTEM>MONITR.EXE.2 [OK]
[Checkpointing] [OK]

PEP>INFORMATION (about product name) TOPS-20-2060-V4

TOPS-20-2060-V4 was successfully INSTALLed
 Patches retrieved through patch tape 2

 Current installed files:
 PS:<SYSTEM>EXEC.EXE.2 sequential checksum 737352, version 4(567)
 PS:<SYSTEM>BUGSTRINGS.TXT.2 sequential checksum 216046
 PS:<SYSTEM>MONITR.EXE.2 sequential checksum 60334, version 4(3370)

 Updated files (ready to be installed):
 None

 Backup files:
 AP20:<AUTOPATCH.BACKUP>EXEC.EXE.1 sequential checksum 765667, version 4(560)
 AP20:<AUTOPATCH.BACKUP>BUGSTRINGS.TXT.1 sequential checksum 511220
 AP20:<AUTOPATCH.BACKUP>MONITR.EXE.1 sequential checksum 647550, version 4(3247)

 Logical names:
 PAT: (autopatching directory) defined as AP20:<AUTOPATCH>
 DIS: (distribution directory) defined as AP20:<AUTOPATCH.TOPS-20>
 BAK: (backup directory) defined as AP20:<AUTOPATCH.BACKUP>
 INS: (installation directory) defined as PS<SYSTEM>
 ASL: (autopatch search list) defined as PAT:,DIS:,SYS:

PEP>EXIT
$
```

A successful installation accomplishes three tasks:

- It establishes a backup copy of the current version of the product you are autopatching.

- It moves the rebuilt product into the installation directory.

- It moves the patched input files into the product's distribution directory.

NOTE

If the installation procedure fails, however, because of a system failure, or because of an error such as "disk quota exceeded," you will need to correct the error and repeat the INSTALL command.

2.2.3 Restoring an Installed TOPS–20 Product

If, for some reason, an installed software product does not perform to your expectations and the previous version is more in keeping with your particular requirements, you can revert to the previous version by using the RESTORE command to PEP.

```
@ENABLE (CAPABILITIES)
$PEP
[PEP version 1(44)]
What is the autopatching directory? AP20:<AUTOPATCH>
 PAT: defined as AP20:<AUTOPATCH>
[Initializing ...] (Reading data base ... ) [OK]

PEP>RESTORE (product name) TOPS-20-2060-V4

[Checking files for restoring TOPS-20-2060-V4] [OK]
[Checkpointing] [OK]
[Restoring files for TOPS-20-2060-V4]
 AP20:<AUTOPATCH.BACKUP>T20GM4.REL.1 => AP20:<AUTOPATCH.TOPS-20>T20GM4.REL.3 [OK]
 AP20:<AUTOPATCH.BACKUP>T20EX4.REL.1 => AP20:<AUTOPATCH.TOPS-20>T20EX4.REL.3 [OK]
 AP20:<AUTOPATCH.BACKUP>EXEC.EXE.1 => PS:<SYSTEM>EXEC.EXE.3 [OK]
 AP20:<AUTOPATCH.BACKUP>BUGSTRINGS.TXT.3 => PS:<SYSTEM>BUGSTRINGS.TXT.3 [OK]
 AP20:<AUTOPATCH.BACKUP>MONITR.EXE.1 => PS:<SYSTEM>MONITR.EXE.3 [OK]
[Checkpointing]  [OK]

PEP>EXIT

$
```

See Section 3.1.1 for a complete description of the RESTORE command.

Contents

Chapter 3
Reference

3.1 The Patch Executive Program

PEP is a multi-function control program to carry out the Autopatch operation on TOPS–10 and TOPS–20. It provides a single interface to manage the details of automatic software patching. Among its advantages are these:

- It reduces the error possibilities inherent in manual patching.

- It requires only a single command to initiate the patching operation.

- It provides a convenient mechanism for installing updated software.

- It provides, as well, a simple means of reverting to the previous software version.

These are the highlights of PEP's responsibilities in the automatic patching operation:

- It maintains a data base of patch and software status information.

- It generates the indirect files required by the batch job that applies patches and rebuilds the software components of a product.

- It performs consistency checks on Autopatch-related files and verifies that the files required by the batch job are in place before allowing the batch job to run.

Run PEP as you would any system program. On a TOPS–10 system type

.R PEP⏎

in response to the system prompt.

On a TOPS–20 system type

`@PEP`(RET)

in response to the system prompt.

If the logical name PAT: is already defined for your job, PEP assumes that the definition specifies the autopatching directory. If the logical name PAT: is not defined, PEP prompts with the question

`What is the autopatching directory?`

Respond by typing the structure and directory you are using for the autopatching directory. PEP will then use this definition to define the logical name PAT:.

NOTE

> If your job has the logical name PAT: defined, and the definition does not identify the autopatching directory, you will have to delete that definition before you can run PEP successfully.

PEP then performs an initialization operation during which it processes the product description file (file type .PDF) in the autopatching directory. If this file cannot be accessed, PEP will terminate with a CBM error. (See Section 3.1.2 for a description of this error and the recovery procedures.) At this time, PEP also reads the data base file it maintains.

Following a successful initialization, PEP is ready to process commands. If you have no products SELECTed for autopatching, PEP will prompt you to declare your autopatchable products. You will not be able to enter the full set of PEP commands until you have SELECTed at least one product.

3.1.1 PEP Commands

You can type a question mark after the PEP prompt to get the list of valid PEP commands. Also, you can get the list of valid arguments for a command by typing a question mark after the command name. For example, if you type a question mark after the PEP prompt, PEP prints the following:

```
PEP>? one of the following:
   AUTOPATCH      BUILD      DESELECT    EXIT        HELP
   INFORMATION    INSTALL    RESTORE     RETRIEVE    SELECT
   SETUP          SUMMARY
```

When typing commands to PEP, you do not need to give the full command name. You can abbreviate the command name, or you can use the recognition feature of PEP. When abbreviating the PEP command, type only enough characters to make the specific command unique. Usually, the first three letters are sufficient to distinguish one command from another. To give a command using the recognition feature of PEP, type a portion of the

command, and press the ESCape key. Depending on where in the command you press the ESCape key, PEP responds with one of the following:

1. The remainder of the command name and any guidewords in parentheses.

2. The guideword in parentheses.

3. The default value for the next argument.

The guidewords tell you the type of argument PEP is expecting you to type next.

If you press the ESCape key before you enter enough characters to make the command unique, PEP rings the terminal bell. At this point, type a question mark to see the choices you have. Then furnish the characters needed to make the command complete.

You can also use recognition input when typing arguments to the PEP commands. If you make a typing error, PEP responds with a question mark. If you type a CTRL/H at this point, PEP retypes the correct portion of the command. You must then retype the portion of the command that was incorrect.

You can mix abbreviated and recognition input in the same command line. This method requires less typing, so you are less likely to make a mistake. Use abbreviated input for the parts of the command you know, and use recognition for the parts of the command you are uncertain of.

AUTOPATCH

Function

The AUTOPATCH command performs three functions:

• updates the PEP data base by processing all of the patch directory files (RETRIEVE processing)

• generates the indirect command files needed for the batch job to do the patch and build operation (SETUP processing)

• verifies that all of the required files are in place and that they are at the required revision level (BUILD processing)

In other words, the AUTOPATCH command performs the same operations as the sequence of the RETRIEVE, SETUP, and BUILD commands. The AUTOPATCH command is the recommended command for applying the patches and rebuilding the product.

Format

AUTOPATCH (product name) product⟨RET⟩

where product is the name of any product that you have SELECTed to be autopatchable at your site

AUTOPATCH (Cont.)

or

AUTOPATCH * (RET)

where * indicates that you want to patch all of the products you have SELECTed at your site

BUILD

Function

The BUILD command initiates the patch-and-rebuild operation for a product by verifying that all files required by the batch job are in place and are at the correct revision level. Before BUILD processing can begin, the product must have been SETUP. When the BUILD command completes successfully, you can then SUBMIT the batch control file for the product, which will apply the patches and rebuild the product components.

Format

BUILD (product name) product(RET)

where product is the name of any product that you have SELECTed to be autopatchable at your site

The following example shows the use of the BUILD command to verify that everything is in order to patch and rebuild APLSF–10.

```
PEP>BUILD (Product name) APLSF-10-V2

[Checking files for building APLSF-10-V2] [OK]
[To Patch and rebuild APLSF-10-V2, issue the monitor command ...]
  SUBMIT DSKP:APL1S2.CTL[7,6]/TIME:1:00:00/RESTART
[Checkpointing] [OK]

PEP>
```

DESELECT

Function

The DESELECT command removes from your site's autopatchable product list the name of a product that you had previously SELECTed to patch. DESELECTing a product will also delete all patch history information for that product from the PEP data base. Therefore, you should use this command with caution. Once DESELECTed, the product cannot be autopatched again until you set up the product distribution and installation directories with the distributed (field-image) version of the product files and use PEP to SELECT the product.

Format

DESELECT (product name) product⟨RET⟩

where product is the name of any product that you have SELECTed to be autopatchable at your site

The following example shows the use of the DESELECT command to remove APLSF–10 from your site's list of autopatchable products. Because this action also deletes from the PEP data base the patch history for this product, PEP asks for confirmation before completing the command.

```
PEP>DESELECT (product name) APLSF-10-V2

%DDH DESELECTing will destroy the patch history of APLSF-10-V2
Do you want to continue? YES
[APLSF-10-V2 removed from the list of SELECTed products]
[Checkpointing] [OK]

PEP>
```

EXIT

Function

The EXIT command allows you to exit from the PEP program and to return to monitor level.

Format

EXIT (from program)⟨RET⟩

Example

```
PEP>EXIT⟨RET⟩
```

HELP

Function

The HELP command prints a paragraph of text about the subject you specify as the keyword.

Format

HELP (with subject name) keyword⟨RET⟩

where keyword is usually any valid PEP command; it can, however, be a term other than a command.

To get a list of the subjects for which you can get an explanation or description, use a ? as the keyword.

INFORMATION

Function

The INFORMATION command displays the current status of an autopatch-able product as well as the status of the files and disk directories associated with that product.

The command lists the following:

- current installed files, with their locations, checksums, edit levels, and version numbers

- updated files

- backup files

- defined logical names

Format

INFORMATION (about product name) product(RET)

where product is the name of any autopatchable product that you have SELECTed to be autopatchable at your site

INSTALL

Function

The INSTALL command places the updated (patched) version of a product into the installation directory (usually defined to be SYS:). It also copies the patched input files for the product into the distribution directory. This command, before superseding the previous version with the patched version, creates backup copies of both the input and the executable files for the product and places these files in the backup directory you have created. This backing-up action allows you to revert later to the old version if you so choose. (See the RESTORE command.)

Before you can INSTALL a product, the batch job that applies the patches and rebuilds the product components must have completed successfully.

Once you type the INSTALL command, the INSTALL processing must be allowed to complete; otherwise files associated with the product you are patching would be left in an indeterminate state.

If an error occurs, or there is a system failure, and the INSTALL processing did not complete, correct the error and reissue the INSTALL command.

Format

INSTALL (product name) product(RET)

where product is the name of any product that you have SELECTed to be autopatchable at your site

RESTORE

Function

The RESTORE command allows you to return to the previous version of the software product that you have patched and installed. That is, you can undo the last INSTALL procedure if for any reason the patched software is unsatisfactory to you. This command restores the product input files found in the distribution directory and the executable files found in the installation directory to their state prior to execution of the most recent INSTALL command.

NOTE

It is possible to RESTORE only the previous version of the software. Multiple backup copies are not maintained.

The RESTORE processing must be allowed to complete; otherwise files associated with the version of the product you are reverting to would be left in an indeterminate state.

If an error occurs, or there is a system failure, and the RESTORE processing did not complete, correct the error and reissue the RESTORE command.

Format

RESTORE (product name) product⟨RET⟩

where product is the name of any product that you have SELECTed to be autopatchable at your site

RETRIEVE

Function

The RETRIEVE command transfers information about the patches for the SELECTed product(s) from the patch directory files (file type .DPD) to the PEP data base in the autopatching directory. When the command completes, it signals that the patches for the products you have SELECTed have been retrieved and are ready for processing.

Format

RETRIEVE (patch entries for all SELECTed products)⟨RET⟩

NOTES

The RETRIEVE command can obtain data only for those products that have been SELECTed.

Every time you receive a new patch tape, you must do RETRIEVE processing to enter the data about the new patches into the PEP data base.

RETRIEVE (Cont.)

Example 1

This example shows the use of the RETRIEVE command. In this case, APLSF–10–V2 was the only SELECTed product and the only patch directory file in the autopatching directory was from patch tape 1.

```
PEP>RETRIEVE (patch entries for all SELECTed products)

[Processing information from patch tape 1]
[Reading patch directory file] [OK]
[23 patches retrieved for APLSF-10-V2]

[Checkpointing] [OK]

PEP>
```

Example 2

This example also shows the use of the RETRIEVE command. However, this site has declared two products, APL–20–V2 and TOPS–20–2060–V4, to be autopatchable. Since the autopatching directory contains the patch directory files from patch tape 1 and patch tape 2, the RETRIEVE command causes PEP to obtain patch information for both these products from both patch tapes.

```
PEP>RETRIEVE (patch entries for all SELECTed products)

[Processing information from patch tape 1]
[Reading patch directory file] [OK]
[20 patches retrieved for APL-20-V2]
[32 patches retrieved for TOPS-20-2060-V4]

[Processing information from patch tape 2]
[Reading patch directory file] [OK]
[7 patches retrieved fro APL-20-V2]
[41 patches retrieved for TOPS-20-2060-V4]
[Checkpointing] [OK]

PEP>
```

Example 3

This example shows that once patch data has been RETRIEVEd from a patch directory file, PEP will not need to process the file again.

```
PEP>RETRIEVE (patch entries for all SELECTed products)

[Processing information from patch tape 1]
[Patches for all SELECTed products already processed from this
tape]

[Checkpointing] [OK]

PEP>
```

SELECT

Function

The SELECT command allows you to declare the products that you want to be autopatchable at your site. It also allows you to define the disk directories associated with each product.

Before PEP adds to its list of autopatchable products the product you have SELECTed, it verifies that all of the associated component input files and control files are available for the products, that they are in the correct directories, and that each file has the expected version number and checksum.

Format

SELECT (product name) product⑧⑤

where product is the name of the product that you are declaring to be autopatchable at your site

NOTES

1. Once you SELECT a product, it is permanently SELECTed, so that you do not need to SELECT it again.

2. If you change your mind about patching a product, refer to the DESELECT command for information on removing the product.

3. A product must be SELECTed before you can reference it in any other PEP command.

Example 1 (TOPS–10)

The following example shows the use of the SELECT command to declare that APLSF–10 is autopatchable. This example also demonstrates that, if necessary, you may set up your disk directories according to your own local conventions rather than using the directories suggested in Chapter 2.

```
PEP>SELECT (product name) APLSF-10-V2
What is the distribution directory for APLSF-10-V2?
DSKB:[10,64,APL]
 DIS: defined as DSKB:[10,64,APL]
What is the backup directory for APLSF-10-V2? OLD:
 BAK: defined as OLD:[1,3]
What is the installation directory for APLSF-10-V2? SYS:
 INS: defined as DSKB:[10,5277],SSL:[1,4]

PEP>
```

SELECT (Cont.)

Example 2   (TOPS–20)

The following example shows the use of the SELECT command to declare that APLSF–20 is autopatchable. This example also demonstrates that, if necessary, you may set up your disk directories according to your own local conventions rather than using the directories suggested in Chapter 2.

```
PEP>SELECT (Product name) APLSF-20-V2
What is the distribution directory for APLSF-20-V2? PS:<APL>
 DIS: defined as PS:<APL>
What is the backup directory for APLSF-20-V2?
PS:<AUTOPATCH.BACKUP>
 BAK: defined as PS:<AUTOPATCH.BACKUP>
What is the installation directory for APLSF-20-V2? SYS:
 INS: defined as SYS:

PEP>
```

SETUP

Function

The SETUP command creates the indirect command files needed to apply the patches to a product. PEP will generate commands to apply those patches necessary to upgrade the product from its current state to the latest revision level.

SETUP processing must be complete before the batch job that applies the patches can run.

Format

SETUP (product name) product(RET)

where product is the name of any product that you have SELECTed to be autopatchable at your site

This example shows the use of the SETUP command to prepare the necessary command files required to apply the patches to APLSF–10.

```
PEP>SETUP (Product name) APLSF-10-V2

[Setting up APLSF-10-V2 for patch and rebuild]
[23 Patches will be applied to APLSF-10-V2]
[Checkpointing] [OK]

PEP>
```

SUMMARY

Function

The SUMMARY command displays on your terminal a list of the patches available for a given product.

Each patch will have a status associated with it. The patch status will be one of the following:

retrieved   A record of the patch has been transferred from the patch directory file to the PEP data base and the patch is ready for processing.

selected   The patch is selected to be applied to the product during the next patch and rebuild operation. During SETUP processing, command files are generated to apply all "selected" patches.

applied   The patch has been applied to all the product input and executable files in the autopatching directory. The batch job that does the patch and rebuild operation has completed successfully.

installed   The patch has been applied to all the product input files in the product distribution directory and to all the product executable files in the product installation directory. "Applied" patches become "installed" when you INSTALL the product. All patches with an "installed" status will be removed when you RESTORE a product.

permanent The patch was applied previous to the last installation of the product. If a patch has a permanent status, it will not be removed when you RESTORE the product.

Format

SUMMARY (of the patches to product name) product⟨RET⟩

where product is the name of any product that you have SELECTed to be autopatchable at your site

3.1.2 PEP Messages and Recovery Procedures

In this section you will find an alphabetized listing of three-letter error message codes, along with the error message strings and the suggestions for recovering from these errors.

PEP issues two types of error messages. Warning messages are preceded by the % symbol. When these errors occur, processing will still continue. Fatal errors are preceded by the ? symbol. When a fatal error occurs, the processing of the command is terminated. Some fatal errors will also terminate the PEP program.

AEM   Address Entry Missing

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

BJF   Batch Job Failed

This error indicates that the batch job that applies the patches and rebuilds the software components of the product did not complete successfully. As a result, the product cannot be installed.

In order to determine why the batch job failed, you will have to look at the .LOG file that was generated. A sample .LOG file is included on the patch tape. These .LOG files show the files that were used or created during the batch jobs. By comparing the corresponding checksums for each file shown in the two .LOG files, you will be able to determine the differences. After you have corrected the problem, you must run PEP and give the AUTOPATCH command before you SUBMIT the batch job again.

CBL   Cannot Build List

Under normal conditions, this error should not occur. It indicates that an internal data structure could not be successfully generated.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

CBM   Cannot Build Master List

This error occurs during PEP initialization. It indicates that the product description file (file type .PDF) could not be read, and, as a result, PEP could not build some of the data structures it requires.

This error will usually be preceded by a COF error message that specifies why the file cannot be read. Usual causes of this error are failure to install the product description file in the autopatching directory or failure to provide PEP with the correct directory specification for the autopatching directory. To correct this situation, verify that the .PDF file is in the autopatching directory, that you have read access to the file, and that the logical name PAT: defines the autopatching directory. Normally, you do not have to define PAT: yourself. PEP prompts you for the information and then defines the logical name. However, if you have entered an incorrect directory, you will have to redefine PAT: yourself.

CBP   Cannot Build Product

This error indicates that the BUILD command cannot be completed successfully. Therefore, the operation is terminated.

This error will usually be preceded by other error messages indicating specifically what is wrong. This error usually occurs when some of the files required to patch and build the product cannot be accessed. In this case, the preceding COF errors identify the specific files. The problems with each file will have to be corrected and then the BUILD command can be reentered.

CCD   Cannot Checkpoint Data Base

This error indicates that PEP cannot update the data base file in the auto-patching directory. Under normal situations, this error should not occur.

This error could indicate that there is no available disk storage in the autopatching directory. You may be able to free up some space by deleting unnecessary files. If the structure is full, you may have to delete files from additional directories. Under TOPS–20, deleted files will have to be EXPUNGEd. Once disk space is available, run PEP, and enter the last command again.

CCF   Cannot Close File

This error indicates that the specified file could not be closed.

This error message will give the reason why the close failed. This condition will have to be corrected, and then the last PEP command can be reentered. This error might also indicate a system problem.

CCJ   Cannot Complete Batch Job

This error will appear in the .LOG file of the batch job that applies patches and rebuilds the software components of a product. It will usually be preceded by additional error messages specifying the problems.

After you have corrected the problem, you must run PEP and give the AUTOPATCH command before you SUBMIT the batch job again.

CCR   Cannot Complete RETRIEVE

This error indicates that the RETRIEVE command cannot be completed successfully. It is produced when an internal consistency check fails; therefore the operation is terminated.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

CCS   Cannot Complete SETUP

This error indicates that the SETUP command cannot be completed successfully. Therefore, the operation is terminated.

This error will usually be preceded by other error messages indicating specifically what is wrong. These problems will have to be corrected and then you can reenter the SETUP command.

CDC   Checkpointed Data Base Corrupt

This error occurs when PEP finds illegal data in the internal data base file PEPCKP.PDB. Normally, this error should not occur.

If the data base file has been damaged, there may be no easy way to correct the situation. Simply getting an old copy of the file off your backup recovery tapes is not recommended, since the old copy may not reflect the current state of your system software and patches. However, you can always recover by deleting the damaged copy of the file and then SELECTing all the autopatchable products again. You will also have to set up your product distribution and product installation directories with the distributed (field-image) versions of the autopatchable software. If all the patch files are still

CDC (Cont.)

in the autopatching directory, you can use the AUTOPATCH command to update your software to the current patch level. If all the patch files are not in the autopatching directory, you will have to restore them from the patch tape before giving the AUTOPATCH command to PEP.

CDF   Cannot Delete File

This error indicates that the specified file could not be deleted.

The error message typed at your terminal will give the reason why the deleted file failed. For example,

%CDF Cannot delete file PAT:APL1S2.BCF (Protection violation)

This condition will have to be corrected, and then the last PEP command can be reentered. This error might also indicate a system problem.

CDL   Cannot Define Logical Name

This error could occur while you are specifying a disk directory for PEP. The error indicates that the specified directory could not be used to define a logical name.

The error message will give the reason why the definition is incorrect. This will usually be due to a syntax error in your response. PEP then reprompts you, so that you may enter the definition correctly.

CEC   Cannot Enter Code

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

CGC   Cannot Get Checksum

This error indicates that PEP was unable to compute the checksum of the file specified in the error message. The error is only a warning, and all processing will continue.

CGV   Cannot Get Version

This error indicates that PEP was unable to compute the version of the file specified in the error message. The error is only a warning, and all processing will continue.

CID   Checksum Is Different

This error is produced while PEP is verifying the sequential checksums of the files that will be used during the autopatch operation. If a discrepancy is found, a message such as

%CID Checksum of SYS:MAKLIB.EXE is different than expected
    expected 251204 but found 353076

is typed on your terminal. This error is only a warning and all processing continues.

However, since the checksum of the file you are using is different from the expected checksum, the operation may not complete as expected. You should try to determine why the checksums are different, before you continue. Most likely, they will be different because you are using an outdated copy of the file, or the file has been locally modified. If the file has in fact been modified, you should determine how these changes will affect the autopatch operation.

CIP   Cannot INSTALL Product

This error indicates that the INSTALL command cannot be completed successfully. Therefore, the operation is terminated.

This error will usually be preceded by other error messages indicating specifically what is wrong. This error usually occurs when some of the files being backed up or replaced cannot be accessed. In this case, the preceding COF errors identify the specific files. The problems with each file will have to be corrected, and then the INSTALL command can be reentered.

COF   Cannot Open File

This error indicates that a specific file cannot be accessed. The message that is typed will indicate whether the file needs read or write access and why the attempt to access it failed. For example,

%COF Cannot open file DIS:APL1S2.REL for input (File not found)

The common causes of this error are "file not found" or "protection violation." To correct these situations you will have to copy the missing file to the right directory or change the protection code. Sometimes you will get a protection violation when you try to create a file in SYS: or another protected directory. In this case, you will need to have OPERATOR or WHEEL privileges enabled before you can successfully complete the operation.

CPF   Cannot Process Batch Status File

Under normal conditions, this error should not occur. It indicates that the status file that was created during the batch job cannot be processed by PEP.

As a result of this error, the PEP data base cannot be updated. Therefore, you should run PEP and give the AUTOPATCH command for this product. Then SUBMIT the batch control file to patch and rebuild the product again.

CPI   Cannot Provide Information

This error indicates that the INFORMATION command cannot successfully be completed and is therefore terminated.

Preceding error messages will give the specific problems.

CRC   Cannot Read Checkpointed Data Base

This error occurs during PEP initialization and indicates that the internal data base file cannot be read. This error will usually be preceded by other error messages identifying the specific problem.

CRD   Cannot Read Patch Directory

This error indicates that the specified patch directory file (file type .DPD) cannot be read. Under normal conditions, this error should not occur.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

CRE   Cannot Read Product Description Entry

This error indicates that the specified product description file (file type .PDF) cannot be read. Under normal conditions, this error should not occur.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

CRF   Cannot Read File

This error indicates a data error while trying to read the specified file.

Since this error occurs during the actual transfer of data, you may be able to reenter the command and have the data transferred without error. This error could be symptomatic of global system problems.

CRP   Cannot RESTORE Product

This error indicates that the RESTORE command cannot be completed successfully. Therefore, the operation is terminated.

This error will usually be preceded by other error messages indicating specifically what is wrong. For example, if the product was previously RESTOREd, it cannot be RESTOREd again. This error usually occurs when some of the files being copied cannot be accessed. In this case, the preceding COF errors identify the specific files. The problems with each file will have to be corrected, and then the RESTORE command can be reentered. You will not be able to RESTORE a product while there is an active batch job to patch and rebuild the product. This will prevent you from changing any of the files that the batch job may be using.

CSP   Cannot SELECT Product

This error indicates that the SELECT command cannot be completed successfully. Therefore, the operation is terminated.

This error will usually be preceded by other error messages indicating specifically what is wrong. This error usually occurs when some of the files in the product distribution directory or product installation directory are missing or cannot be accessed. In this case, the preceding COF errors identify the specific files. The problems with each file will have to be corrected, and then the SELECT command can be reentered.

CWF   Cannot Write File

This error indicates a data error while trying to write the specified file.

Since the error occurs during the actual transfer of data, you may be able to reenter the command and have the data transferred without error. This error could be symptomatic of global system problems.

DAD   Directory is Already Defined

This error occurs while you are defining a disk directory for PEP. This error indicates that you had previously defined that directory for another use.

PEP then reprompts you, so that you may enter a new definition.

DDH   DESELECTing Will Destroy History

This error warns that DESELECTing the product will delete all status and patch history information from the PEP data base.

If you complete this command, the product will no longer be autopatchable. In order to make the product autopatchable again, you will have to set up the product distribution directory and product installation directory with the distributed versions of the software components and then run PEP and SELECT the product again.

DEF   Delete Table Entry Failed

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

FAM   Files Appear to be Missing

This error is produced when PEP cannot access a file that was created during the patch and build patch job. Normally, this error should not occur.

This error will be preceded by a COF error for each file that cannot be accessed. The COF error message will specify the file and the associated problem. The best recovery procedure is to run PEP and enter the command to AUTOPATCH that product. Then SUBMIT the batch job again.

FCW   Format of Checkpointed Data Base is Wrong

This error indicates that the data in the data base file is not in a format that PEP can process. This error should not occur under normal conditions.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

FNB   Files Not Backed Up

This error indicates that during the backup cycle of INSTALL processing, not all files got backed up. There will be a preceding error message specifying each file that did not get backed up and the problem.

See error INC for possible recovery procedures.

FNR   Files Not Replaced

This error indicates that during the replace cycle of INSTALL processing, not all files got replaced with their updated versions. There will be preceding error message specifying each file that did not get replaced and the problem.

See error INC for possible recovery procedures.

IDL   Illegal Data in Line

Under normal conditions, this error should not occur. It indicates that a PEP data file (e.g., patch directory file or product description file) contains an incorrectly formatted entry.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

IDT   Illegal Datatype

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

IFR   Initialization Failed for Reading

This error indicates that PEP was unable to perform the initialization required to read a specified file. In most cases, it means the file cannot be accessed.

The usual cause of this error is that the specified file is not in the disk directory that PEP expects it to be in. After verifying that the file is in the correct directory and that you have read access to it, run PEP and try the operation again.

INC   Installation Not Completed

This error indicates that INSTALL processing started, but it was terminated before it could complete successfully. This error will usually be preceded by other error messages specifying why the operation was terminated.

This error usually occurs when disk quotas are exceeded, or disk storage is exhausted during the file backup or replacement operation. If this is the problem, you will have to provide additional disk storage before you can continue. Often you do this by EXPUNGing deleted files from the directory that is full. Once you obtain the additional disk storage, run PEP and reenter the INSTALL command for the product. PEP will then complete the operation.

JNF   Job Never Finished

This warning message indicates that, as far as PEP can determine, the batch job that was SUBMITed to patch and rebuild the product has not completed. It must complete before the updated software can be INSTALLed.

If the batch job is still running or in the queue waiting to run, you will have to let it complete before you can proceed. However, if the batch job is no longer in the queue, you will have to run PEP and AUTOPATCH the product again. Then reSUBMIT the batch job.

MBI   Must Be INSTALLed

This error indicates that the last INSTALL operation for the product did not complete successfully. As a result, the correct versions of the product files may not be in the appropriate directories, and won't be until the INSTALL operation successfully completes.

Before PEP will process any more commands for that product, you must correct the problem that previously terminated the INSTALL and then run PEP and reenter the INSTALL command for the product. PEP will then complete the INSTALL operation.

MBR   Must Be RESTOREd

This error indicates that the last RESTORE operation for the product did not complete successfully. As a result, the correct versions of the product files may not be in the appropriate directories, and won't be until the RESTORE operation successfully completes.

Before PEP will process any more commands for that product, you must correct the problem that previously terminated the RESTORE and then run PEP and reenter the RESTORE command for the product. PEP will then complete the RESTORE operation.

NDF   No Patch Directory Found

This error indicates that no patch directory files were found in the auto-patching directory. Therefore, no patches will be applied to any of the auto-patchable products.

This error occurs when the patch directory files from the patch tape are not transferred from the patch tape to the autopatching directory. Patch directory files have a file type of .DPD.

NFD   Null Files Descriptor

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

NPR   No Patches RETRIEVEd

This warning indicates that PEP has no patch information for the specified product.

If patches have been distributed for that product, make certain that the patch directory files (file type .DPD) have been copied from the patch tape to the autopatching directory. There is one patch directory file for each patch save set on the tape. Once you have copied these files to the auto-patching directory, run PEP and issue the RETRIEVE command. All new patch information for all autopatchable products will then be read from all patch directory files and entered in the PEP data base.

NPS   No Product Specified

This error indicates that an incomplete command has been typed. The command requires you to specify the name of a product.

Retype the command specifying a product name. To see a list of valid product names, type the command followed by a question mark. For example,

```
AUTOPATCH ?
```

NTS   No HELP Topic Specified

This error indicates that an incomplete HELP command has been typed. The command requires you to specify the name of a product.

Retype the command specifying a keyword. To see a list of valid keywords, type the command followed by a question mark. For example,

```
HELP ?
```

PAS   Product Already SELECTed

This error is only a warning and indicates that the product has previously been SELECTed to be an autopatchable product, and it does not have to be SELECTed again.

If you want to go through SELECT processing again (to redefine the disk directories associated with the product), you will first have to DESELECT the product.

PNB   Product Not Built

This error indicates that you have attempted an operation that first requires the product to be successfully built.

Before you can complete the current operation, you will have to use the BUILD command to PEP to rebuild the product. The product will be rebuilt when the batch job successfully completes.

PNI   Product Not INSTALLed

This error indicates that you have attempted an operation that first requires that product to be successfully INSTALLed.

Before you can complete the current operation, you will have to run PEP and INSTALL the product.

PNS   Product Not SETUP

This error indicates that you have attempted an operation that first requires the product to be successfully SETUP. This error may also appear in the batch .LOG file indicating that the product was not SETUP properly before the batch control file was SUBMITted.

Before you can complete the current operation, you will have to run PEP and enter the SETUP command to prepare the product to be patched and rebuilt. Once you complete SETUP processing, you will need to give the BUILD command to PEP before the batch job can be SUBMITted.

PPM   Prerequisite Patches Missing

This error occurs during RETRIEVE processing when you attempt to retrieve patch information for a product from the current patch directory file without having processed some of the previous patch directory files containing patch information for that product. Before PEP will enter in its data base any information about the latest set of patches for a product, a record of all previously distributed patches must already be in the data base. PEP obtains this information from the patch directory files (file type .DPD) during RETRIEVE processing. As long as all the patch directory files are in the autopatching directory, PEP will be able to obtain the information it needs. However, if some of the previous directories have been deleted, this error could occur.

To correct this situation, copy the missing patch directory files from the patch tape to the autopatching directory. Then run the PEP program and retype the RETRIEVE command. (In general, you can delete a patch directory file from the disk once it has been processed by the RETRIEVE command. However, if you later SELECT a new product to be autopatchable, you will need all previous patch directory files containing patch information for that product.)

RNC   RESTORE Not Completed

This error indicates that RESTORE processing started, but it was terminated before it could complete successfully. This error will usually be preceded by other error messages specifying why the operation was terminated.

This error usually occurs when disk quotas are exceeded, or disk storage is exhausted during the file-copying operaton. If this is the problem, you will have to provide additional disk storage before you can continue. Often you can do this by EXPUNGing deleted files from the directory that is full. Once you obtain the additional disk storage, run PEP and reenter the RESTORE command for the product. PEP will then complete the operation.

SIB   SETUP Will Interface with Batch Job

This warning message indicates that the SETUP operation you are about to perform could interfere with the patch and rebuild batch job that you have already SETUP for the product. Once you SETUP a product and initiate BUILD processing, no additional processing can be done for that product until its batch job completes. That way, none of the files used during the batch job will be inadvertently modified.

If you have not already SUBMITted the batch job, you may proceed with this SETUP without any problems. However, if you have SUBMITted the batch job, you should cancel it or let it complete before you attempt to continue with the current SETUP.

SSF   Symbol Table Search Failed

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

TEF   Table Entry Failed

Under normal conditions, this error should not occur. It indicates that an internal data structure could not be successfully generated.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

UDT   Unknown Datatypes

Under normal conditions, this error should not occur. It indicates that an internal consistency check has failed.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

UHT   Unrecognized HELP Topic

This error indicates that an incorrect HELP command has been typed. The keyword that you specified was not valid.

Retype the command specifying a valid keyword. To see a list of valid keywords, type the command followed by a question mark. For example,

```
HELP ?
```

UPL   Unrecognized Product Library

This error is provided as part of a consistency check during the processing of patch directory files. It should seldom occur.

In most cases you will not be able to correct this problem. It should be reported to your Software Specialist.

UPN   Unrecognized Product Name

This error indicates that an incorrect command has been typed.

Retype the command specifying a valid product name. To see a list of valid product names, type the command followed by a question mark. For example,

```
AUTOPATCH ?
```

URC   Unrecognized Command

This error indicates that the command just typed is not valid.

Retype the command. To see a list of valid commands, type a question mark in response to the PEP prompt. For example,

```
PEP>?
```

Remember, many commands (AUTOPATCH, BUILD, INSTALL, RESTORE, RETRIEVE, and SETUP) will not be recognized until at least one product has been SELECTed.

VID   Version Is Different

This error is produced while PEP is verifying the version numbers of the .EXE files that will be used during the autopatch operation. If a discrepancy is found, a message such as

%VID Version number of SYS:MAKLIB.EXE is different than expected
    expected 2A(67) but found 2A(21)

is typed on your terminal. This error is only a warning and all processing continues.

However, since the version of the file you are using is different from the expected version, the operation may not complete as expected. You should try to determine why the versions are different, before you continue. Most likely, they will be different because you are using an outdated copy of the file, or the file has been locally modified. If the file has in fact been modified, you should determine how these changes will affect the autopatch operation.

3.2 System Programs Used with Autopatch

3.2.1 BACKUP and DUMPER

The Autopatch tapes are distributed in BACKUP interchange format for TOPS–10 and in DUMPER format for TOPS–20. You must run these utilities to retrieve all items from the patch tapes and place them in the proper directories so that they are accessible to Autopatch.

On the patch tape are included control files that enable you to retrieve easily only the needed files for the products you are patching. For the BACKUP utility these are indirect command files. For the DUMPER utility, they are batch control files. The README documentation file will identify these files.

3.2.2 Batch

Each autopatchable product is furnished its own Galaxy batch control file for patching and rebuilding that software product. The file has been standardized so that you do not need to do any editing or further customizing to make it run on a supported configuration.

The batch control file performs four sequential functions:

- It defines logical names for the directories associated with the product.

- It applies patches to the product input files.

- It rebuilds the software components of the product from the patched input files.

- It creates a status file for PEP which signals the success or failure of the batch job.

The batch job produces a log file which shows exactly what happened while the batch job was running. It keeps a record of system errors encountered in the process, and therefore is a debugging aid. The patch tape contains a corresponding log file (.Lnn) that you can print out and compare with your batch log file. (The patch log file is included on the patch tape just for information; it is *not* required as part of the Autopatch operation.)

3.2.3 MAKLIB

The MAKLIB utility is run as part of the batch job. It applies the patch files to the product libraries as instructed by the indirect command files.

3.2.4 PEPB

PEPB is the Patch Executive Program for Batch. It is run as part of the batch job that patches and rebuilds a software product. It is functionally different from PEP. PEPB performs the initialization required for the batch job and verifies the results.

NOTE

Use PEP, not PEPB, to interactively prepare a product for autopatching.

3.3 Disk Directories and Logical Names Used with Autopatch

Each of the disk directories used by Autopatch has a logical name associated with it.

PAT: is the logical name defined for the autopatching directory
DIS: is the logical name defined for the product distribution directory
BAK: is the logical name defined for the backup directory
INS: is the logical name defined for the installation directory
ASL: is the logical name defined for the Autopatch search list, which is further defined as PAT:,DIS:,SYS:.

The Patch Executive Program (PEP) handles the details of defining these logical names, so that you do not have to define them explicitly yourself. The INFORMATION command to PEP will show what logical name definitions have been established for a particular product.

3.4 Files Used with Autopatch

3.4.1 Files Supplied on the Distribution Tapes of Autopatchable Products

REL Libraries (file type .REL)
> These are input files that are used to build the executable files of the product. Patches are applied to these REL libraries. You copy these files to the product distribution directory when you prepare your system for autopatching.

Link Command Files (file type .LNK)
> These are command files used by Link to build the executable files of the product from the REL libraries for the product. You copy these files to the product distribution directory when you prepare your system for autopatching.

3.4.2 Files Supplied on the Patch Tape Used to Install Autopatch

INSTAL.DOC
> This file is provided in the Installation Save Set of the patch tape and gives additional information needed to prepare your system for autopatching.

DUMPER Distribution Control Files (file type .DDC) (TOPS–20 only)
> These files are provided in the Installation Save Set of the patch tape. You can use these files to help you set up the product distribution directory for an autopatchable product. Each file runs under batch and transfers from the product distribution tape to the product distribution directory those files needed for autopatching. INSTAL.DOC describes the use of these files.

BACKUP Distribution Control Files (file type .CCL) (TOPS–10 only)
> These files are provided in the Installation Save Set of the patch tape. You can use these files to help you set up the product distribution directory for an autopatchable product. Each file, an indirect command file for BACKUP, transfers from the product distribution tape to the product distribution directory those files needed for autopatching. INSTAL.DOC describes the use of these files.

3.4.3 Files Supplied on the Patch Tape Used for Patching

README.nnn

One of these files is included with each patch save set on the tape. The file type .nnn specifies the number of the patch set. The number of the current patch set will be the same as the patch tape number. Each README file contains important information about the files in that patch set.

TOPS–10 Patch Retrieval Files (file types .Ann and .Pnn)

These files are included in the current patch save set for each auto-patchable product. They are indirect command files for BACKUP. You can use the .Pnn file to transfer from the current patch save set to the autopatching directory all those files required to apply the current set of patches to the product. Or, you can use the .Ann file to transfer from ALL the patch save sets to the autopatching directory those files required to apply all patches to the product. The README documentation file will describe the use of these files.

TOPS–20 Patch Retrieval Files (file types .Ann and .Pnn)

These files are included in the current patch save set for each auto-patchable product. They are batch control files that run DUMPER. You can use the .Pnn file to transfer from the current patch save set to the autopatching directory all those files required to apply the current set of patches to the product. Or, you can use the .Ann file to transfer from ALL the patch save sets to the autopatching directory those files required to apply all patches to the product. The README documentation file will describe the use of these files.

Patch Directory File (file type .DPD)

Each patch save set has a patch directory file that describes all the patches contained in the save set. You must transfer this file from the tape to your autopatching directory. PEP will process the entries in this file when it executes the RETRIEVE command.

Patch and Build Batch Control Files (file type .CTL)

Each autopatchable product has its own patch and build patch control file. When SUBMITted to the Galaxy Batch System, this control file will patch and rebuild all the components of the product. Each .CTL file has been standardized so that you do not need to do any editing or further customizing to make it run on a supported configuration. These files are provided in the Installation Save Set of the patch tape. You copy these files to the autopatching directory when you prepare your system for autopatching.

Product LOG Files (file type .Lnn)

A LOG file is provided for each autopatchable product. It shows a successful patch and rebuild operation. These LOG files are provided only as a reference tool so that you can compare them with your LOG files.

Edit Description Files (file type .Dnn)

The edit description file on each patch tape is a documentation text file that can be printed out if you wish to see it. (Unlike README and INSTAL.DOC, which have to be printed out and used in conjunction with the manual, the use of the edit description file is *not* mandatory.)

This file provides the user with symptom, diagnosis, and cure information, basically the same kind of information you would find in a reply to an SPR.

3.4.4 PEP Data Files

These files are used by the Patch Executive Program. Unless otherwise indicated, these files are created and maintained by PEP. You should not alter or delete them.

Product Description File (file type .PDF)

This file describes each autopatchable product. It is distributed in the Installation Save Set of the patch tape, and you copy it to the auto-patching directory when you prepare your system for autopatching.

PEP Data Base (file type .PDB)

PEP creates and maintains this data base file containing the patch history and status of each product that you have SELECTed for autopatching.

Patch Command Files (file type .MKL)

PEP generates one of these command files for each autopatchable product. Each file contains MAKLIB commands for applying patches to the product REL libraries.

Batch Communication Files (file type .BCF)

PEPB generates one of these files at the completion of the patch and rebuild batch job. These files contain data for PEP about the status of the batch job.

Index

NOTE

In this index we refer to some of the complementary files on tape, chiefly INSTAL.DOC and README.

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments sub-
mitted on this form at the company's discretion. If you require a written reply
and are eligible to receive one under Software Performance Report (SPR) ser-
vice, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make
suggestions for improvement.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Did you find errors in this manual? If so, specify the error and the page number.

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Please indicate the type of reader that you most nearly represent.

☐ Assembly language programmer
☐ Higher-level language programmer
☐ Occasional programmer (experienced)
☐ User with little programming experience
☐ Student programmer
☐ Other (please specify) _____

Name _____   Date _____

Organization _____   Telephone _____

Street _____

City _____   State _____ Zip Code _____
                                                          or Country
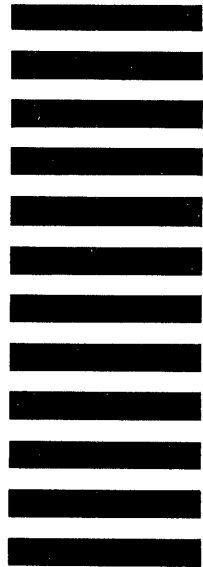
digital

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS
200 FOREST STREET MR1-2/E37
MARLBOROUGH, MASSACHUSETTS 01752