IBM

World Trade Systems Centers

OFFICE SYSTEMS INTERCONNECTION: GUIDE TO NETWORK MANAGEMENT FOR DPCX/DOSF SYSTEMS

GG24-1578-0

Office Systems Interconnection: Guide to Network Management for DPCX/DOSF Systems

Document Number GG24-1578-0

May 18th, 1983

William Taylor
IBM United Kingdom

Project Advisor: Gordon Hay
Raleigh International Systems Centre
PO Box 12195, Dept 987/622
Research Triangle Park
NC 27709, USA

The information contained in this document has not been submitted to any formal IBM test and is distributed on an 'As Is' basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk. The samples described in this material are presented for illustrative purposes only and are not intended to be implemented as described.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

First Edition (May 1983)

Publications are not stocked at the address below; requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to:

Raleigh International Systems Centre
IBM Corporation (985/B622-3)
PO Box 12195
Research Triangle Park, NC 27709, USA

IBM may use or distribute any of the information you supply without incurring any obligation to you. You may, of course, continue to use the information you supply.

ABSTRACT

There are many products and techniques which could be used to help manage a network of DPCX/DOSF systems; choosing between them, however, and combining a selection of them into a coherent system, may be a complex task.

The purpose of this guide is to help the network manager select the products and techniques he needs, and special attention is paid to three areas in which DPCX/DOSF systems have unique requirements:

- Software Management
- Configuration Management
- Problem Management

This guide is essentially a practical document, not a theoretical one, and offers detailed guidance on how to make best use of the tools. It places particular emphasis on centralised network management techniques, and includes tested procedures for using them, with working examples.

ABOUT THIS BOOK

This manual suggests procedures for managing a small to medium sized network of 8100s running DPCX and text applications. It lays particular emphasis on techniques which enable personnel using the central site host computer effectively to maintain, control and diagnose problems on systems which are physically remote.

This book assumes a working knowledge of DPCX with DOSF and familiarity with SNA networks. It does not teach basic skills, but seeks to draw together a number of diverse techniques, and, where they overlap, to assist the user in selecting between them.

WHO SHOULD READ THIS BOOK

The intended audience for this publication is data processing personnel at the central computer site, particularly:

- Systems programmers involved in maintaining distributed software.
- Network administrators or others involved with the configuration of the 8100 systems.
- Help Desk or other user support staff with involvement in 8100 problem determination.

This book does not set out to formulate procedures for 8100 Control Operators or DOSF/DISOSS operators.

HOW THIS BOOK IS ORGANISED

The manual is organised into an introduction and four major parts:

- The introduction defines more precisely the environment for which the guidance in this manual is intended, and lists those things which are considered pre-requisite to being able to use it.
- Part 1 considers in detail a methodology for the maintenance of software in distributed DPCX/DOSF systems.
- Part 2 investigates the data which should be recorded about the configuration of DPCX/DOSF systems and precautions which should be taken to maintain their integrity.
- Part 3 discusses aspects of problem determination unique to DPCX and DOSF systems, and how such problem determination may be carried out from the central site.

- Part 4 consists of appendices which contain background information on some topics, and sample aids related to the procedures in the main body of the book.

RELATED PUBLICATIONS

This manual collates network management techniques which are separately documented in a number of different publications. Each manual referred to in this book is listed in the "Bibliography" on page 187, with its forms number and a brief description.

CONTENTS

1.0 INTRODUCTION

This document has been designed to assist the user who is installing or who has recently installed a network of 8100s for text applications. It seeks to identify the products and procedures which are necessary to operate effectively in such an environment.

The typical installation which this document is intended to aid is described below. As networks become larger, or add other applications, additional network management tools may become necessary. Where possible, pointers are provided to identify such new requirements arising from increasing network complexity, and possible ways of meeting the need, but these are not investigated in detail.

1.1 THE TYPICAL NETWORK ENVIRONMENT

The following assumptions are made about the network:

1. The 8100s are all running a similar text application, using DPCX with DOSF and DISOSS for document filing and distribution.

2. There are approximately ten 8100s in the network.

3. All 8100s are connected to a host System/370, 4300, 303x or 308x, running OS/VS2 MVS, OS/VS1 or DOS/VSE control programs.[1]

4. The SNA network is based on ACF/VTAM and ACF/NCP, and may use the multi-system networking facility.

5. The SNA network is already in place or being put in place.

6. The following CNM products are installed:

   - Network Communications Control Facility (NCCF), 5735-XX6, release 2 or later, with the Terminal Access Feature (TAF).

   - Network Problem Determination Application (NPDA) Version 2, 5668-983.

   - Host Command Facility (HCF) Version 2, 5668-985.

   - Host Prep, 5735-XR3, release 4.1 or later.

7. Any required DB/DC subsystem (such as CICS or IMS) is generated with the DPCX support needed.

8. The host job entry subsystem is configured to support DPCX RJE.

9. One 8100 is available for testing and development (it is hereafter called the test 8100), but it may also be used for production work.

---

[1] The procedures in this document have been tested with MVS, but should work except where otherwise noted for OS/VS1 and DOS/VSE. The JCL given is that for MVS.

10. The test 8100 is not necessarily at the host site, but it is accessible to personnel responsible for installing and maintaining DPCX, and there is access to host CNM products and batch job submission facilities (for example, using DPCX DSC and RJE respectively).

11. The test 8100 has an 8809 tape unit attached, but no other 8100 need have a tape.

12. The skills of the 8100 control operators on the production machines are limited to text functions, and following documented procedures or verbal instructions.

13. A centre of competence exists to assist 8100 control operators with day-to-day problems with DPCX and DOSF. This centre of competence is elsewhere referred to as the Help Desk. It may be as small as a single person, perhaps the network administrator also responsible for planning and configuring DPCX/DOSF systems.

14. ASSIST installation, and hence system defined categories, has been used in the initial installation of each DPCX system.

PART 1: SOFTWARE MANAGEMENT

This part of the manual should be read by system programmers or network administrators responsible for maintaining software on 8100 DPCX/DOSF systems.

2.0 INTRODUCTION TO SOFTWARE MANAGEMENT

The management of software in a network of distributed systems must address the following areas:

- Change control: the orderly testing and application of modifications to IBM and user programs, with capability for regression if necessary.
- Distribution: the mechanics of applying software changes to a number of geographically separated systems.
- Recording: the maintenance of accurate records of the level of the software at each system; what modifications have been applied, and when.

Whilst the adoption of proper change control procedures is strongly advocated, there are no considerations which are specific to the environment of a network of DPCX/DOSF systems. In consequence, such procedures are not discussed here, but if you wish to know more, ask your IBM representative for an appropriate installation management manual. The Information/Management feature of the Information/System program product, 5735-OZS, described in "Overview of the Information/Management Program Product" on page 90, contains functions to assist in change control. The processes of distribution and recording are discussed below.

For 8100 DPCX/DOSF systems it is necessary to distinguish between two classes of software, each of which will require different installation and maintenance procedures.

First, there is System Software: this includes the DPCX operating system itself, (and, prior to DPCX Release 3, its feature #6001), and DOSF. These programs are distinguished by being delivered on diskette 2D for installation by the procedures detailed in the DPCX DOSF Installation manual.

Programs in the second category are hereafter referred to as Application Software. They include DISOSS, DIF and any user-written DPCX code, and are distinguished by being written to run as DPCX Function Programs. They are distributed on tape for installation via the host System/370, or on 128 byte sectored diskette 1 for installation via the DPCX SYSBDISK System Service.

The different techniques to be used with these two categories of software are described in two further chapters in this part of the manual, under "Network Management of System Software" on page 7 and "Network Management of Application Software" on page 59.

3.0 NETWORK MANAGEMENT OF SYSTEM SOFTWARE

The techniques used for the software management of the system programs vary according to the type of maintenance being handled. The following are discussed separately:

- Initial installation of the DPCX and DOSF products throughout the network and subsequent update installations to new releases. See "Base Code Initial and Update Installation."
- Application of PTFs throughout the network. See "PTF Management for System Software" on page 19.

3.1 BASE CODE INITIAL AND UPDATE INSTALLATION

The guidance in this section does not seek to teach you about DPCX and DOSF planning and installation. These topics are covered in the DPCX Planning manual, the DOSF Planning manual and the DPCX DOSF Installation manual. What is documented here are specific techniques to minimise the skills and involvement required in the installation process at the production 8100 sites. The intention is that when the operators first sign on to the 8100 the DPCX/DOSF system should be fully operational.

The processes of initial and update installation are discussed separately in "DPCX/DOSF Initial Installation" on page 8 and "DPCX/DOSF Update Installation" on page 12.

- Besides being required on a new 8100, an initial installation may be desirable if you want to make substantial changes to the layout of disk space when, for instance, an extra disk has been added. It creates an entirely new DPCX/DOSF system, destroying entirely anything that was there before. If you decide to overwrite an existing system in this way, then the guidance in "PART 2: CONFIGURATION MANAGEMENT" on page 77 may be of use to you in making the transition.
- An update installation will upgrade software to the next release, but will leave unaltered all user documents and data, and most configuration information.

3.1.1 INTRODUCTION TO NIM

DPCX release 3 introduces a capability for installing DPCX and DOSF base code from the host. This function requires the Distributed Systems Executive program product at the host system. It also needs at least a minimal installation of DPCX release 3 or later at any 8100 which is to receive code across the network.

NIM works by distributing images of the installation diskettes to the 8100s to be installed. These images are saved on disk, and NIM supports transmission of special command lists to an 8100 to cause installation to be performed from disk images of the diskettes.

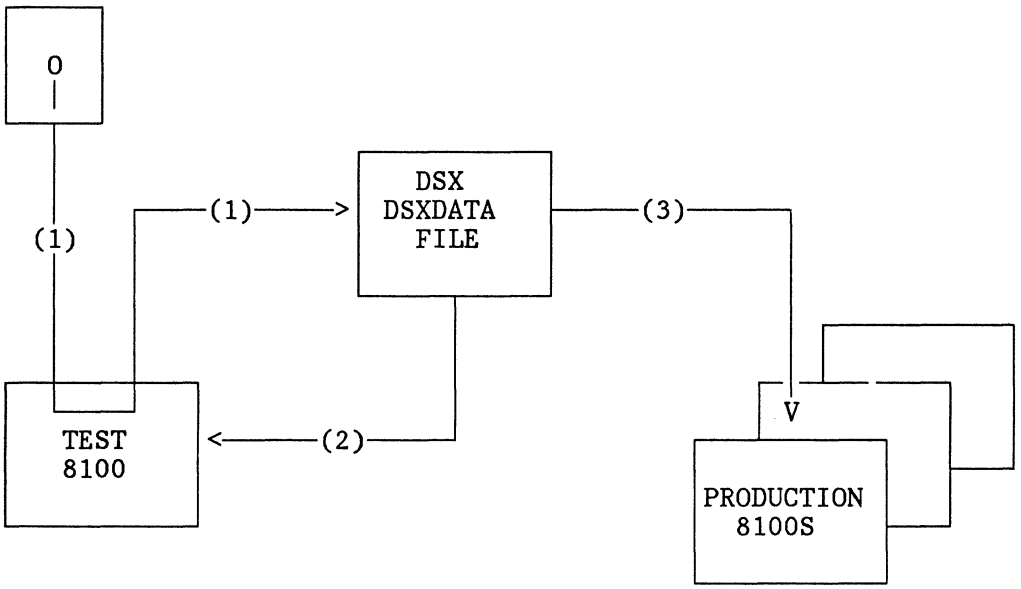


Figure 1. Installing DPCX and DOSF with NIM: this is a schematic of the process of installing software with the Network Installation Management capability of DPCX release 3.

NIM would be used in a network of 8100s as illustrated in Figure 1 on page 8. The numbered steps are as follows:

1. DSX is used to retrieve images of the installation diskettes from the test 8100 to the DSX holding file. The data may be kept here to install future 8100s. Note that in this step the test 8100 merely serves as a diskette reader: the contents of the diskettes are not stored at the test 8100, but are simply transmitted to DSX.

2. The diskette images may be transmitted first to the test 8100 system and installed here to verify the NIM installation process and the new release.

3. The diskette images are transmitted to production 8100 systems and installed.

As well as transmitting DPCX and DOSF program code, DSX can also retrieve to the host a set of SYSCONFG and SYSIMOD configuration data, customised with the SYSRCNFG System Service. DSX can send these records to an 8100 and instruct NIM to copy them into the system configuration dataset.

NIM also has a capability to allow a DPCX configuration IPL using the host as the configuration device, so that configuration of a newly installed system may be completed from the host.

3.1.2 DPCX/DOSF INITIAL INSTALLATION

Three techniques are available for an initial installation of DPCX and DOSF:

- Installation from diskettes using ASSIST
- 'Manual' installation from diskettes, not using ASSIST
- Installation from the host, using the Network Installation Management (NIM) capability of DPCX release 3.

Of these three techniques, installation using ASSIST is strongly recommended unless very special requirements necessitate the 'manual' installation. From DPCX release 3 onwards considerable flexibility in re-allocation of bit maps is possible on an ASSIST installed system, so making use of ASSIST practicable in all but very exceptional cases. A 'manual' installation will require more work to configure an operational system. Use of NIM is not suggested because it does not perform an ASSIST installation and so does not configure the bitmaps in a suitable way for a production DOSF system, and it cannot be used without first installing the DPCX release 3 diskettes on the 8100. It is considerably easier to use the ASSIST installation rather than either of the alternatives.

The ASSIST installation is very well documented in the DPCX DOSF Installation manual and in the DOSF Installation Card for ASSIST. Its use will result in an almost fully configured system with no more skill required at the 8100 site than the ability to follow the instructions in the ASSIST installation card. The SYSIMOD host parameters and the SYSHOST parameters, however, will not be configured. The paragraphs which follow describe a technique by which this can be accomplished from the host, and in consequence allow all remaining customisation to be performed without DPCX skills at the 8100 site. It uses part of the NIM function of DPCX release 3.

As pre-requisites to using this technique, you need the following:

- DPCX release 3 or later; presumably just installed with ASSIST along with DOSF on a new 8100.
- All terminals to have been powered on, and all loops and links to have been connected during that installation.
- The DPCX LU with local address 02 defined in the NCP.
- Host Command Facility (HCF) installed at the host.
- Instructions for the control operator to reply to the IPL prompts on the 8100 Basic Operator Panel (BOP).

Having installed with ASSIST[2], follow the steps set out below:

1. Have the control operator perform a configuration IPL designating the host as the logon device thus:

---

[2] Ensure that, during the ASSIST installation process, PTF UC01830 is installed. This PTF is on PTF Diskette 28 or above, and without it, the subsequent configuration IPL to the host may fail with a code 4934 displayed on the BOP. This will occur in the following circumstances: if the ASSIST installation process finds more than one SDLC adaptor without powered-on devices attached, it cannot guess which of them is intended to be used as the host link. Therefore, no host link will be configured, and without PTF UC01830 the 4934 error will subsequently occur. If ASSIST is able to configure a host link, the configuration IPL to the host will work without the PTF.

- Set the IPL mode select switch to 'manual', and the security key-lock (if fitted) to 'enable'.

- Press the IPL button.

- When the BOP displays 0200, key in 13dd and press the Enter Data button. 13dd designates the IPL disk address, and so the reply to this prompt would normally be 1380 or 1384 depending on processor model.

- When the BOP displays 0201, key in 9A00 and press the Enter Function button.

- When the BOP displays 43AA, key in 0001 and press the Enter Function button.

- When the BOP displays 43AB, key in D7hh and press the Enter Function button. D7hh designates the host adaptor address; for instance key in D710 if the host adaptor is the first port on the first 8101.

- When the BOP displays 43CD, key in the host link parameters. These are fully described in the DPCX DOSF Installation manual. Press the Enter Function key.

- IPL should process to completion, when the BOP display will be blank[2].

2. For the configuration IPL, host processing bypasses validation of the host SSCP and batch applications against the SYSHOST tables. The LU with local address 01 is available for type 1 batch (NIM functions only) and that with local address 02 for HCF.

   At a host terminal, sign on to HCF. Acquire the DPCX LU with local address 02.

3. The DPCX logon menu will appear. Sign on as the control operator, 01 with password of OPID01.

4. Because this was a configuration IPL, you will be forced to run SYSCONFG. Use option 1 to ensure that the host adaptor is correctly configured. Use option 3 to process the configuration.

5. SYSCONFG concludes by IPLing the 8100, and your HCF session will be lost. This IPL remembers the information specified previously and will activate the host again in a similar fashion, but now all type 1 batch functions are available to the LU with local address 01.

   From HCF, acquire the LU with local address 02 as before.

6. When the DPCX logon menu appears, sign on as the control operator.

7. Using SYSIMOD, ensure that the host configuration information is set correctly.

8. Define the SYSHOST parameters and the PROC to be executed at system reset. Use one of the two procedures that follow. The first may be simpler if only one 8100 is involved; the second may be preferable if several installations are to be performed and DSX is available. (Take note of the SPE required for DSX, mentioned in "DPCX/DOSF Update Installation" on page 12.)

a. Using HCF only:

   - Use the DEFINE HOST command with ID, BATCHID and LUPGMID operands to establish the SYSHOST parameters.
   - Using SYSEDIT, define a procedure named IPLPROC containing an ENABLE HOST command.
   - Issue the command DEFINE STARTUP=IPLPROC to identify the procedure to DPCX.

b. Using HCF and DSX:

   - Issue the command ENABLE HI=2 to allow a second host-initiated session.
   - Run a DSX job such as that illustrated in Figure 3 on page 13 to transmit PROCs 'DEFHOST' and 'IPLPROC' to DPCX. Sample JCL procedures are shown in Figure 2 on page 12.
   - Issue the command DEFINE STARTUP=IPLPROC to identify the system reset procedure to DPCX.
   - Enter the command DEFHOST to run that PROC.

     Note: If at this point the DEFHOST command is not recognised, use SYSEDIT to access the PROCs and file them again. The DEFHOST command should now work.

9. Enter the command DISABLE HOST,HI

10. Use SYSIMOD option 8 to IPL DPCX.

11. After this IPL, the DPCX terminals should be active as well as the fully configured host link. HCF may now be used to communicate with the LUs designated by the installation, to perform further customisation.

If any problems occur with the host link during this procedure, DPCX will automatically re-enable the host link if it fails while the configuration IPL is in effect. If, for any reason, you need to re-IPL DPCX before SYSCONFG has processed, have the control operator perform the configuration IPL again. If DPCX needs to be IPLed before you have established the host parameters, have the control operator commence the IPL as for the configuration IPL, but in place of '9A00' enter '9A80'. This will re-establish the special host environment which existed after SYSCONFG IPLed DPCX.

If, after the host parameters have been set and DPCX been IPLed by SYSIMOD option 8, the 8100 does not come active to VTAM, it may be because the NCP has discontacted it after the IPL. Once the network resources have been activated, instruct the control operator to perform a primary IPL at the 8100; this time the 8100 should come active to VTAM.

After the installation, DSX may be used to transmit DOSF sample (or other) PROCs to DPCX. These would have been previously retrieved to the host (with DSX) from the test system. Alternatively, these may be loaded directly from diskette. The spelling dictionaries will have to be loaded from diskette at the 8100 site as shown on the DOSF Installation Card for ASSIST.

```
//*
//* THIS PROC IS USED FOR DSX BATCH FUNCTIONS
//*
//DSX       PROC
//PREP      EXEC  PGM=DSXPREP
//STEPLIB   DD    DSN=DSX2.DSXLOAD,DISP=SHR
//DSXCMF    DD    DSN=DSX2.DSXCMF,DISP=SHR
//DSXLIB1   DD    DSN=DSX2.DSXLIB1,DISP=SHR
//DSXLIB2   DD    DSN=DSX2.DSXLIB2,DISP=SHR
//DSXDATA   DD    DSN=DSX2.DSXDATA,DISP=SHR
//DSXTCF    DD    DSN=DSX2.DSXTCF,DISP=SHR
//SYSPRINT  DD    SYSOUT=A
//DSXPRINT  DD    SYSOUT=A
//         PEND



//*
//* THIS PROC IS USED FOR DSX TRANSMISSIONS
//*
//DSXXMIT   PROC
//XMIT      EXEC  PGM=DSXTM000
//STEPLIB   DD    DSN=DSX2.DSXLOAD,DISP=SHR
//DSXCMF    DD    DSN=DSX2.DSXCMF,DISP=SHR
//DSXLIB1   DD    DSN=DSX2.DSXLIB1,DISP=SHR
//DSXLIB2   DD    DSN=DSX2.DSXLIB2,DISP=SHR
//DSXDATA   DD    DSN=DSX2.DSXDATA,DISP=SHR
//DSXTCF    DD    DSN=DSX2.DSXTCF,DISP=SHR
//DSXCWK    DD    DSN=DSX2.DSXCWK,DISP=SHR
//SYSPRINT  DD    SYSOUT=A
//DSXPRINT  DD    SYSOUT=A
//DSXSTAT   DD    DSN=DSX2.DSXSTAT,DISP=SHR
//         PEND
```

Figure 2. OS/VS JCL for executing DSX: sample JCL procedures showing the DSX datasets required to use the Network Installation Management capability of DPCX Release 3.

3.1.3 DPCX/DOSF UPDATE INSTALLATION

The same three techniques are available for an update installation as for an initial installation:

- Installation from diskettes using ASSIST.
- 'Manual' installation from diskettes, not using ASSIST.
- Installation from the host, using the Network Installation Management (NIM) capability of DPCX release 3.[3]

[3] Note that the requirements and restrictions described here relate to the use of NIM to update-install a DPCX/DOSF R3 system on an older DPCX/DOSF R3 system. This is probably not a common requirement; the real intention of NIM is to ease the installation of future DPCX/DOSF releases on a R3 base. It may be that the restrictions, described here

```
//DSX      JOB
//         EXEC  DSX,PARM='FUNCTION=CMFMAINT'
//SYSIN    DD    *
****** AFTER CONFIGURATION IPL, WE MUST USE LOCADDR 01
 CHANGE MASTER CLUS=TEXTAB,BATCH=INTXAB01
 ADD CLIST ID=((DEFHOST,1,0)),CLUS=TEXTAB
 ADD CLIST ID=((IPLPROC,1,0)),CLUS=TEXTAB
 END
/*
//         EXEC  DSX,PARM='FUNCTION=LIBMAINT'
//SYSIN    DD    *
 REPLACE CLIST ID=(DEFHOST,1,0),CMD='*PROC',                    X
               CMD='DEFINE HOST,ID=11',                         X
               CMD='DEFINE HOST,BATCHID=SIRF',                  X
               CMD='DEFINE HOST,BATCHID=DSX',                   X
               CMD='DEFINE HOST,LUPGMID=(10,958)',              X
                       . . . . . . . . . .
                       . . . . . . . . . .
               CMD='DEFINE HOST,LUPGMID=(52,932)',              X
               CMD='DEFINE HOST,LUPGMID=(55,932)',CMD='*EXIT'
 REPLACE CLIST ID=(IPLPROC,1,0),CMD='*PROC',                    X
               CMD='ENABLE HOST,LU=(1,64),HI=3',CMD='*EXIT'
 END
/*
//         EXEC  DSX,PARM='FUNCTION=DEFSESS'
//SYSIN    DD    *
 DEFINE SESSION CLUS=TEXTAB,TIME=0900,DISP=(NEW,DELETE)
 REPLACE CLIST ID=((DEFHOST,1,0)),NEWNAME=SYSPROC.DEFHOST
 REPLACE CLIST ID=((IPLPROC,1,0)),NEWNAME=SYSPROC.IPLPROC
 END
/*
//XMIT     EXEC  DSXXMIT
//         EXEC  DSX,PARM='FUNCTION=CMFMAINT'
//SYSIN    DD    *
****** RESTORE BATCH LU FOR NORMAL OPERATIONS
 CHANGE MASTER CLUS=TEXTAB,BATCH=TEXTAB11
 END
/*
//
```

Figure 3. DSX control statements to send PROCs to a DPCX system: these statements will cause DSX to transmit the DEFHOST and IPLPROC PROCs to the 8100 known as TEXTAB.

A 'manual' update installation should not be performed on an ASSIST installed system; it cannot be done without first setting off the ASSIST bit with SYSIMOD. Conversely an ASSIST update installation cannot be performed on a system not installed with ASSIST. Hence if installing from diskettes, use ASSIST on an ASSIST system, but the 'manual' update installation on a non-ASSIST system.

The alternative is to use NIM to perform the update centrally from the host. Note that this procedure, which is described below, will handle only

---

for R3 installation, will be different for installation of a future release; it is therefore very important to check the announcement and availability documentation for the release you are installing.

new releases of DPCX and DOSF, not new releases of IDTF or other 8775 microcode features, nor a new version of the spelling dictionaries. It has the following prerequisites:

- DPCX release 3 or later installed on a test (or 'source') 8100
- DPCX release 3 or later installed on production (or 'target') 8100s
- The diskettes for the new release of DPCX and/or DOSF available at the test 8100 site
- Distributed Systems Executive Version 1 (5748-XXG) installed at the host system. In order to function with NIM, DSX Version 1 requires an SPE (Small Programming Enhancement) which is contained in PTF PTF90061[4].
- The DPCX LUs with local addresses 01 and 02 defined in the NCP
- Host Command Facility (HCF) installed at the host
- Instructions for the control operator to reply to the IPL prompts on the 8100 Basic Operator Panel (BOP)

Note: The NIM update install process overwrites any 8775 microcode features (including IDTF) which may be present in system dataset 1. They cannot be re-installed except by loading them from diskette at the 8100s which have been update installed.

If you use IDTF and wish to use NIM for update installations, you may wish to use SASRAC (the Stand-Alone Save, Restore And diskette Copy program, as the DASD dump/restore program is known in DPCX release 3), to make a copy of an IDTF diskette when it is installed. The copy would then be available for re-installation after an update install.

Note: The NIM update install process requires that ASSIST be set off at the target 8100s. This implies, of course, that future Update installations cannot be done with ASSIST, but seems to have no other significant disadvantages.

It is assumed that the test (or 'source') system is that known to DSX as TEXTAA; the examples show how to update target system TEXTAB. Several DPCX clusters could be identified to DSX as a group and update installed together. Sample DSX procedures as referenced in the examples of DSX control statements are shown in Figure 2 on page 12. Follow these steps:

1. Run the DSX control statements shown in Figure 4 on page 15, to retrieve images of the DPCX and DOSF program diskettes to the host. The control operator at the test system will receive messages, and a 96FE prompt appears on the BOP display, asking for diskettes to be mounted. The diskettes are requested by feature code (9502 or 9500). If you use a national language feature, mount that diskette in place of diskette 3 of 3 for DPCX (9502) or 2 of 2 for DOSF (9500). If an incorrect diskette is mounted twice in response to a prompt the function will be terminated.

   The transmission of the diskette images may take several hours, depending on line speed, the load on the host system and so on. Dur-

---

[4] No support for the DPCX NIM functions has been announced in DSX Version 2 Release 1 (5668-986).

```
//DSX      JOB
//         EXEC  DSX,PARM='FUNCTION=DEFSESS'
//SYSIN    DD    *
 DEFINE SESSION CLUS=TEXTAA,TIME=0900,DISP=(NEW,DELETE)
 RETRIEVE DATASET ID=(SYSINST.R03M00.9502),FCTLNAME=DPCX,DISP=OLD
 RETRIEVE DATASET ID=(SYSINST.R03M00.9500),FCTLNAME=DOSF,DISP=OLD
 END
/*
//XMIT     EXEC  DSXXMIT
//
```

Figure 4. DSX control statements to retrieve installation data: these statements will cause DSX to retrieve installation diskette images from the DPCX system known as TEXTAA.

ing the transmission the BOP displays a count as a progress indicator, and normal DPCX/DOSF operations may continue.

2. On the target systems, 16 bitmaps have to be allocated to category 118 subcategory 1. This may be difficult on a one-disk system, but the bitmaps are required only for the duration of the update install. If the impact to operation is acceptable, these bitmaps may be borrowed from the system spool:

   - Run SYSSPREC to ensure that all spool data is valid.
   - Print and delete all spool files.
   - All but the first bitmap in category 122, subcategory 1 should now be empty. Using SYSCAT option 1, re-assign the last 16 to category 118, subcategory 1. (An ASSIST system has over 20 bitmaps in category 122, subcategory 1)

   This can be done even if system assigned categories are in effect.

   On systems with two or more disks these bit maps probably can be found in category 1 subcategory 1; you may wish to allocate them permanently to category 118 when you initially install the system.

3. On the target systems, ASSIST must be set off using SYSIMOD option 1 suboption 8. The NIM update install cannot be performed if ASSIST is on. After setting ASSIST off, you will not be able to use ASSIST to update install your system (so you must choose between ASSIST and NIM), but there will be no impact to its operation.

4. Run the DSX control statements shown in Figure 5 on page 16 to send the installation diskette images to the target 8100(s).

```
//DSX      JOB
//         EXEC  DSX,PARM='FUNCTION=DEFSESS'
//SYSIN    DD    *
 DEFINE SESSION CLUS=TEXTAB,TIME=0900,DISP=(NEW,DELETE)
 REPLACE DATASET ID=(SYSINST.R03M00.9502),FCTLNAME=DPCX,          X
               ORIGIN=TEXTAA
 REPLACE DATASET ID=(SYSINST.R03M00.9500),FCTLNAME=DOSF,          X
               ORIGIN=TEXTAA
 END
/*
//XMIT     EXEC  DSXXMIT
//
```

Figure 5. DSX control statements to send installation data: these statements will cause DSX to send installation diskette images to the holding area at the DPCX system known as TEXTAB.

The installation diskettes are placed in a holding area on disk, contained in category 118.

The transmission of the diskette images may take several hours. During the transmission the BOP displays a count as a progress indicator. Normal system operation can continue.

5. Perform the update install, by running the DSX control statements shown in Figure 6 on page 17 which sends an IPL command list to the target 8100(s). Of the commands, 9A05 causes DPCX to be installed from the holding area, and 9A08 has the same effect for DOSF. 9B00 causes immediate execution of the commands, irrespective of what other tasks are running. 9B08 may be used in place of 9B00 and defers execution of the commands until DPCX is shutdown and all operators signed off.

6. The DPCX system(s) will perform a configuration IPL; that is, prompt 43AB will be displayed at the BOP.

7. Have the control operator perform the following:

   - When the BOP displays 43AB, key in D7hh and press the enter function button. D7hh designates the host adaptor address; for instance key in D710 if the host adaptor is the first port on the first 8101.

   - When the BOP displays 43CD, key in the host link parameters. These are fully described in the DPCX DOSF Installation manual. Press the enter function key.

   - IPL should process to completion, when the BOP display will be blank.

8. For the configuration IPL, host processing bypasses validation of the host SSCP and batch applications against the SYSHOST tables. The LU with local address 01 is available for type 1 batch (NIM functions only) and that with local address 02 for HCF.

   At a host terminal, sign on to HCF. Acquire the DPCX LU with local address 02.

```
//DSX       JOB
//          EXEC  DSX,PARM='FUNCTION=CMFMAINT'
//SYSIN     DD    *
 ADD CLIST ID=((SYSCSEQ,1,0)),CLUS=TEXTAB
 END
/*
//          EXEC  DSX,PARM='FUNCTION=LIBMAINT'
//SYSIN     DD    *
 REPLACE CLIST ID=(SYSCSEQ,1,0),CMD='9A05,9A08,9B00'
 END
/*
//          EXEC  DSX,PARM='FUNCTION=DEFSESS'
//SYSIN     DD    *
 DEFINE SESSION CLUS=TEXTAB,TIME=0900,DISP=(NEW,DELETE)
 REPLACE CLIST ID=((SYSCSEQ,1,0))
 END
/*
//XMIT      EXEC  DSXXMIT
//
```

Figure 6. DSX control statements to send IPL commands to a DPCX system: these statements will cause DSX to transmit the IPL commands to perform an update install, to the 8100 known as TEXTAB.

9. The DPCX logon menu will appear. Sign on as the control operator, 01 with password of OPID01.

10. Because this was a configuration IPL, you will be forced to run SYSCONFG. Use option 2 to restore the configuration.

11. This process concludes by IPLing the 8100, and your HCF session will be lost. This IPL remembers the information specified previously and will activate the host again in a similar fashion, but now all type 1 batch functions are available to the LU with local address 01.

    Acquire the LU with local address 02 as before.

12. When the DPCX logon menu appears, sign on as the control operator.

13. Use SYSIMOD option 2 to restore SYSIMOD information. Occasionally some data may be invalidated by the update installation. If so, use SYSIMOD option 1 to respecify.

14. Verify that the SYSHOST parameters have not been invalidated by the update installation. Respecify them if necessary.

15. Enter the command DISABLE HOST.

16. Use SYSIMOD option 8 to IPL DPCX.

17. After this IPL, the DPCX terminals should be active as well as the fully configured host link. HCF may now be used to communicate with the LUs designated by the installation, to perform further customisation.

If any problems occur with the host link during this procedure, DPCX will automatically re-enable the host link if it fails while the configuration

IPL is in effect. If, after the final IPL, the 8100 does not come active to VTAM, it may be because the NCP has discontacted it after the IPL. Once the network resources have been activated, instruct the control operator to perform a primary IPL at the 8100; this time the 8100 should come active to VTAM.

After the update installation, DSX may be used to transmit new versions of the DOSF sample (or other) PROCs to DPCX. These would have been previously retrieved to the host (with DSX) from the test system.

If bitmaps were borrowed from (for example) the system spool to allocate to the holding area, they should be returned using SYSCAT option 4 to delete category 118 and option 1 to assign them to category 122.

3.1.4 APPLICATION OF PTFS

After an initial or update installation, applicable PTFs should be installed. See "Installing PTFs after an Initial or Update Install" on page 55 for details of how to do this from the host.

If DPCX and DOSF are installed from diskettes, you may wish to install the current fix package at the same time, from the PTF diskette, and using ASSIST if yours is an ASSIST installed system. In this case, you may need to do further work to remove any PTFs known to be in error, or to install additional corrective service PTFs. In either case, the methods in "Suggested Technique for Handling PTFs" on page 19 are applicable.

3.2 PTF MANAGEMENT FOR SYSTEM SOFTWARE

There are two techniques available for management of PTFs to DPCX system software. They are:

1. Manual transmission of PTFs on diskette and application with the SYSPTF System Service.

2. Host Prep SYSINFOREF (Subsystem Information Retrieval Facility, 5735-XR3).

Of the two possibilities, for any network with more than three or four 8100/DPCX systems geographically distributed, use of SYSINFOREF is likely to be more productive. It is the primary tool discussed here. Information on the initialisation of SYSINFOREF and important notes on its operation are to be found in "SYSINFOREF Initialisation and Use" on page 125, and you should read this before attempting to use the product.

3.2.1 SUGGESTED TECHNIQUE FOR HANDLING PTFS

DPCX and SYSINFOREF provide between them a number of possible ways to achieve the end result of applying PTFs across the network. "DPCX and SYSINFOREF PTF Handling Capabilities" on page 133 contains a summary of the different possible routes, and some comparison of the DPCX and the SYSINFOREF functions. The course recommended here employs only SYSINFOREF, and was chosen because it will remain practicable as the network grows in size. Underlying the strategy is the assumption that one 8100 is designated as a test machine, on which the PTFs will be applied first, and not until any problems have been ironed out will they be sent to other, production DPCX systems. The steps are as follows; they are summarised schematically in Figure 7 on page 20.

1. Get the PTFs into the SYSINFOREF datasets at the host.

   a. For preventive service PTFs, distributed on diskette:

      1) The test 8100 is used as a diskette reader.

      2) The PTFs are retrieved to the host.

   b. Corrective service PTFs are entered directly at the host.

2. Send the PTFs to the test 8100 and install them there. When you are satisfied with the operation of the test machine, proceed to the next step.

3. Send the PTFs to the production 8100s and install them there.

Although it is possible to perform installation of PTFs directly from diskette for the test system, it is preferable to retrieve them to the host and send them back to the test 8100 so that any problems that might arise in the PTF network distribution and installation process are discovered at this stage rather than when working with the production systems.

The following terminology is employed when referring to PTFs in SYSINFOREF or DPCX datasets:

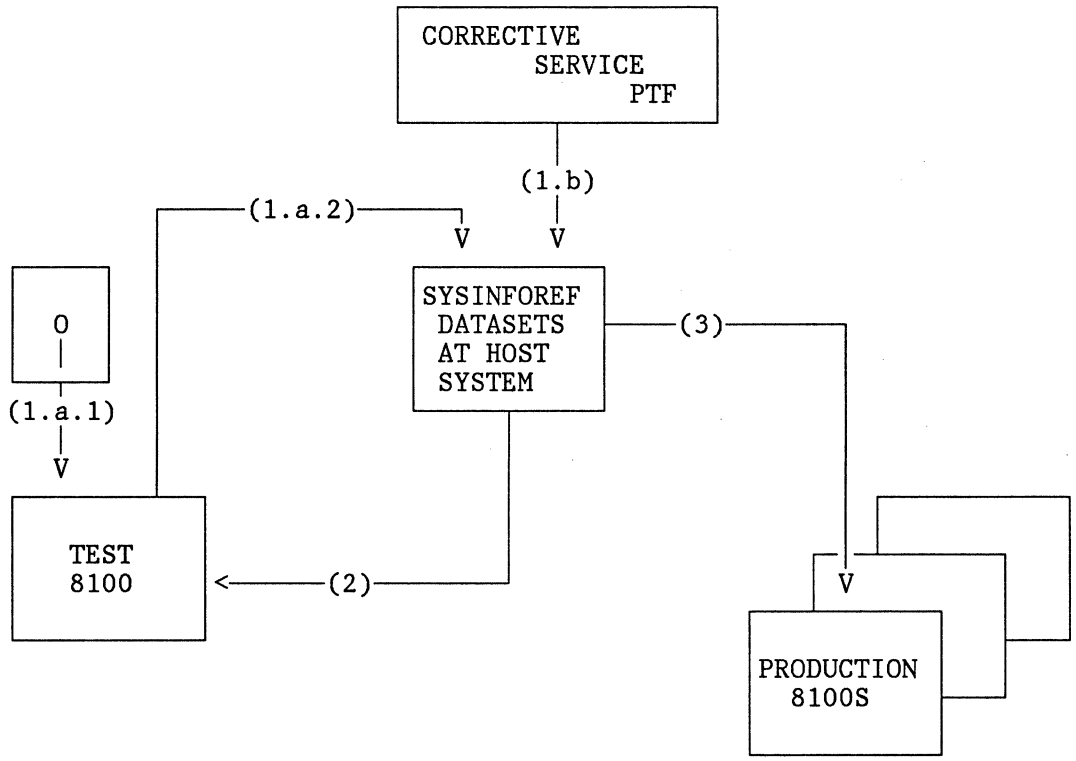


Figure 7. Network application of PTFs: this is a schematic overview of installing PTFs across the network.

- DPCX PTF dataset: this is DPCX system dataset 12, and PTFs are stored, maintained and tracked here until the next DPCX/DOSF update installation.
- DPCX system dataset 8: for DPCX releases prior to 3.0, this dataset is used to hold an image of a PTF diskette.
- SYSINFOREF temporary PTF library, host temporary library or TEMPLIB: is the logical library within a SYSINFOREF VSAM dataset where PTFs retrieved from an 8100 are initially placed.
- SYSINFOREF PTF library or host PTF library: is the logical library within a SYSINFOREF VSAM dataset where PTFs are maintained at the host, and from which they may be sent to 8100s.

3.2.2 INSTALLING PREVENTIVE SERVICE TO SYSTEM SOFTWARE

This section suggests a procedure for applying PTF diskettes to DPCX and DOSF throughout the network. It aims to retain a central library of all PTFs at the host while the software levels to which they apply are current anywhere in the network. It assumes that SYSINFOREF has been initialised as in "Initialising SYSINFOREF" on page 127. Examples of control statements are included. Use the same JCL as that in Figure 61 on page 127.

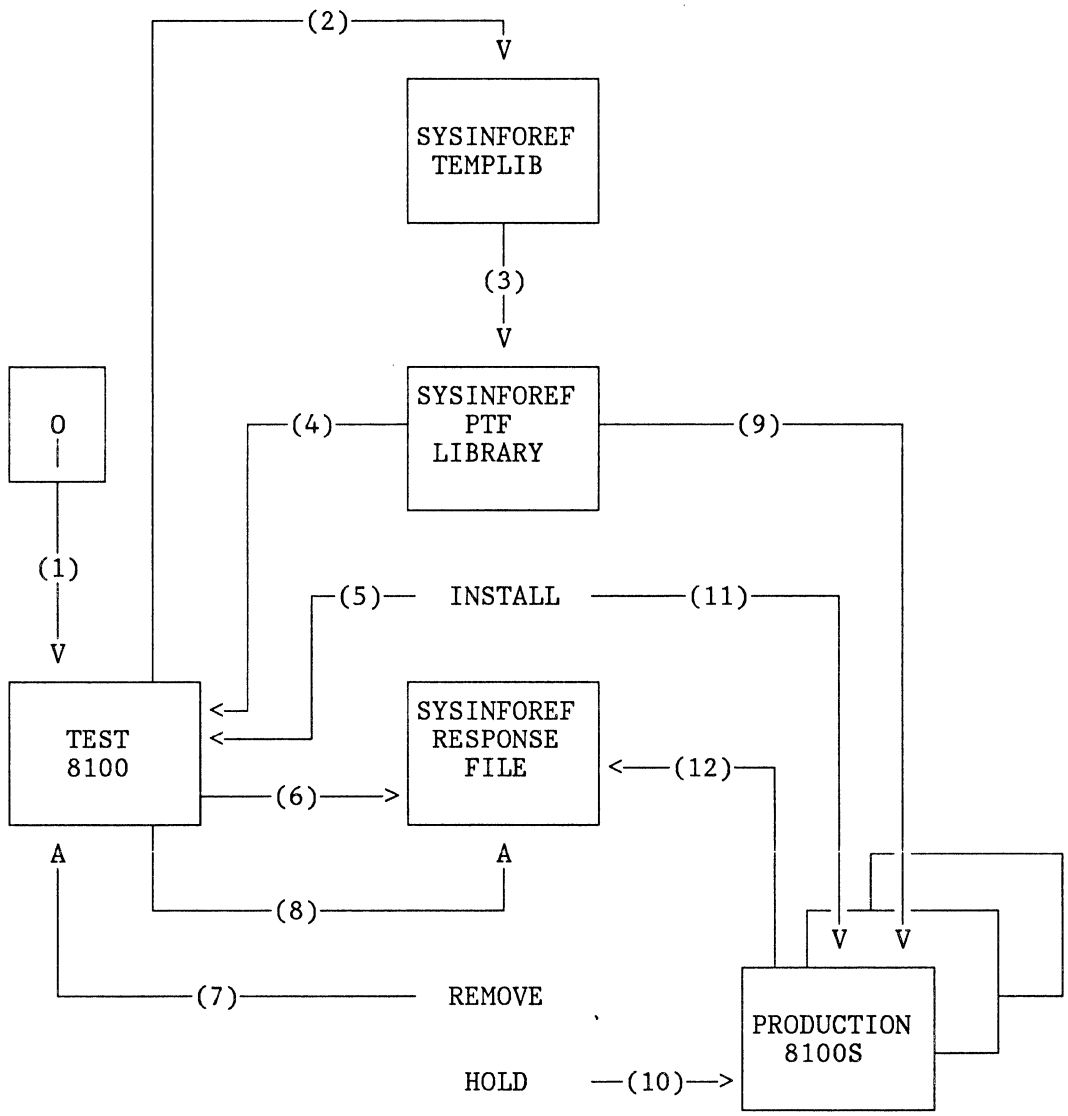


Figure 8. Network application of preventive service: this is a schematic of the processes involved in installing a diskette of PTFs across the network.

The steps are as follows. They are illustrated in Figure 8 on page 21.

1. For DPCX prior to Release 3, IPL the PTF diskette to load its contents to system dataset 8 at the test DPCX system.

2. Use SYSINFOREF to retrieve the PTFs on the diskette to the host temporary PTF library from the test DPCX system.

3. For all software levels in use in the network, copy applicable PTFs from the host temporary PTF library to the host PTF library.

4. Send the PTFs to the test DPCX system.

5. Install the PTFs at the test DPCX system.

6. Determine the status of the test DPCX system: retrieve the PTF installation messages and test and list PTF status.

IPL and test the operation of the test DPCX system until you are satisfied it is working normally.

7. A PTF might have to be removed if it impacts operation.

8. Retrieve the PTF removal messages from the test DPCX system if a PTF had to be removed.

9. Send the PTFs to production DPCX systems.

10. If any PTFs had to be removed from the test system, place these PTFs in HOLD status at production systems.

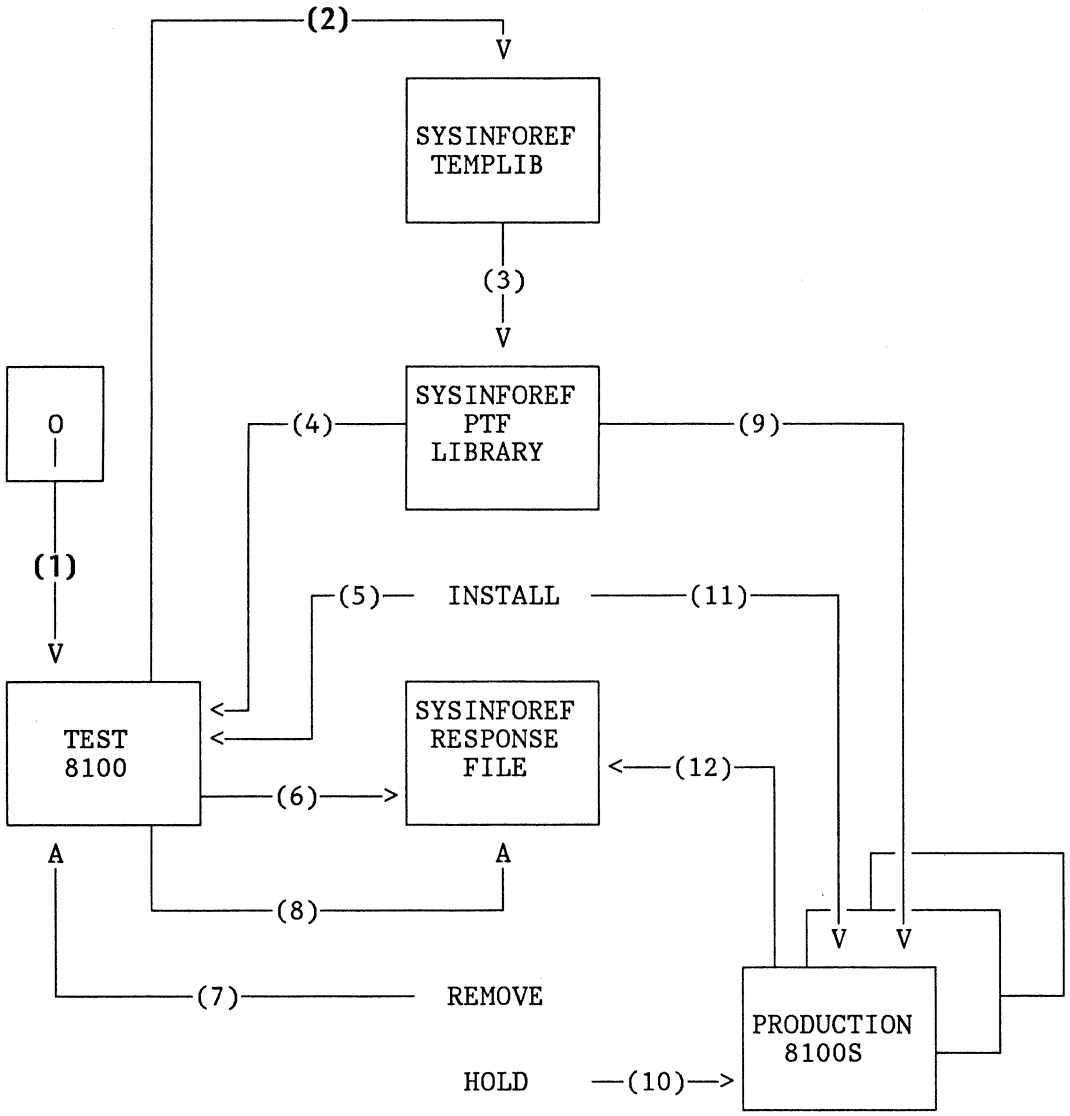
11. Install the PTFs at the production systems.

12. Determine the status of the production DPCX system: retrieve the PTF installation messages and test and list PTF status.

The examples assume that there are some DPCX systems in the network with DPCX Release 2.1, feature 6001E and DOSF Release 2.0, and some with DPCX Release 2.2, feature 6001F and DOSF Release 2.1. Furthermore, the production systems have been defined to the SYSINFOREF machine list as being in class 1, and the test 8100 as being in a separate class, 900. (See "SYSINFOREF Classes" on page 126 for a discussion of SYSINFOREF classes.) PTF diskette 23 is to be installed.

For convenience of illustration, the sample control statements show the SYSINFOREF maintenance, generation, communication and response phases being executed consecutively within a single jobstream. As described in "SYSINFOREF Phases and Timing of Sessions" on page 128, it may well be desirable to separate the execution of the different phases.

THIS PAGE LEFT INTENTIONALLY BLANK

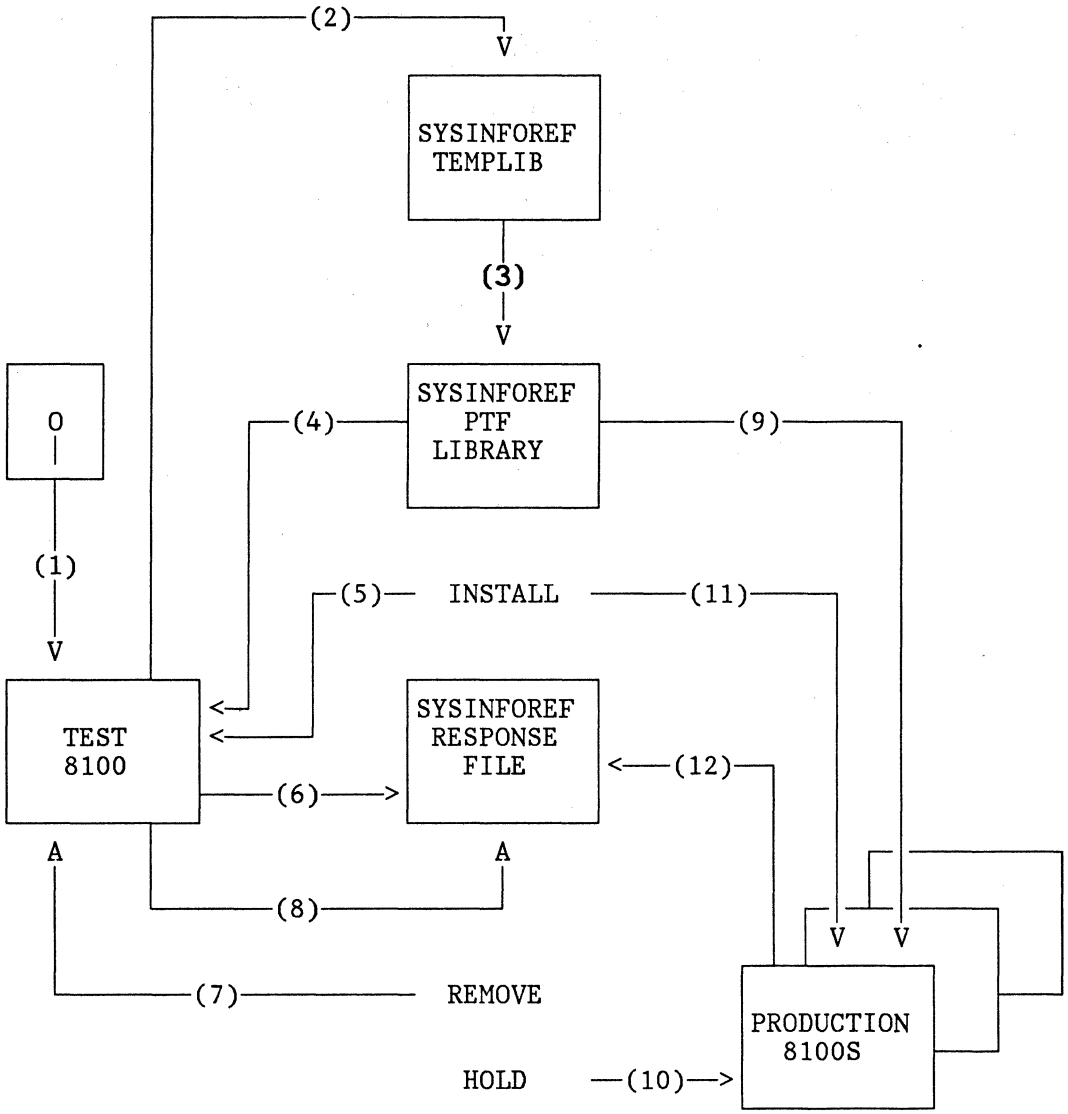

(2)
SYSINFOREF
TEMPLIB
(3)
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
(8)
(7)
REMOVE
HOLD
(10)
PRODUCTION
8100S

Steps 1 and 2: Retrieve the entire diskette of PTFs to the host: On DPCX systems prior to Release 3, the diskette must first be IPLed as described in the DPCX DOSF Installation manual. For Release 3 or later systems, the PTF diskette should be loaded and ready in the reader on the 8100 before the SYSINFOREF job is initiated. Sample control statements follow in Figure 9.

Note: Do not follow the instructions on the DOSF Installation Card for Assist - that process does more than is wanted here.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     RETRIEVE     PTF                 DISKETTE             ;
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)       SYSID(TESTZZ10)      ;
```

Figure 9. Retrieving a PTF diskette with SYSINFOREF: these control statements will retrieve all the PTFs on a diskette to the host.


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

Step 3: Copy PTFs into the host PTF library: When PTFs are retrieved to the host, they are placed on the PTF temporary library (TEMPLIB). It is necessary to use the maintenance phase of SYSINFOREF to copy them into the host PTF library. Copy all PTFs for software levels and features that are present on DPCX systems in the network, as in Figure 10.

Only those PTFs that are new on this diskette, that is, which are not already on the host PTF library will be copied. The LIST PTF statement causes header information to be listed for each PTF in the host PTF library. This statement is not needed in Host Prep R5, since the ADD PTF statements will cause the required information to be reported. Separate copies of PTFs exist in the library for each level or feature to which they apply. Note the identification of the diskette for the PTFs just loaded; you will need to code this in the control statements which send the PTFs to DPCX systems.

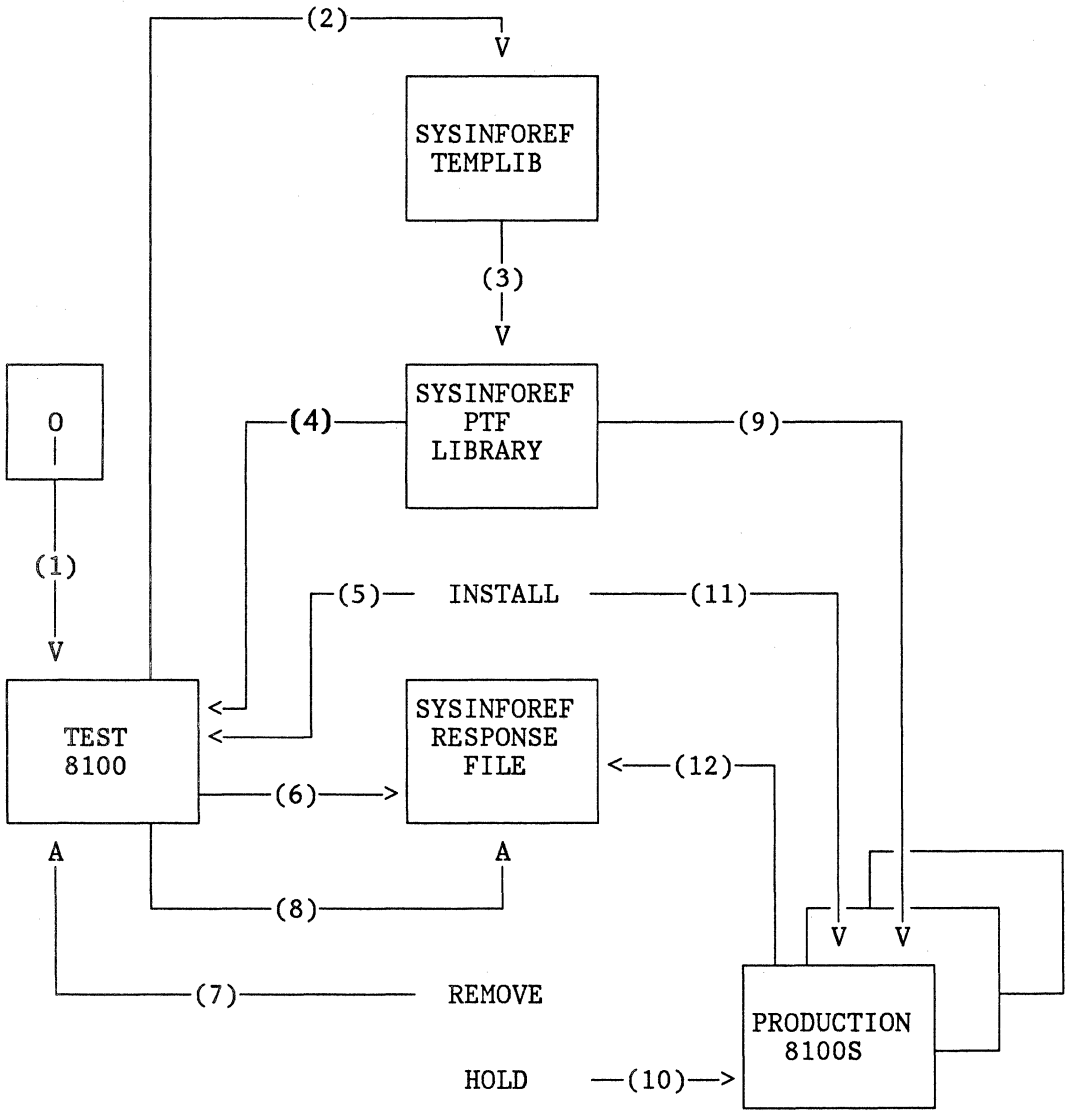
```
.&     ADD          PTF                  LEVEL(R02M01)
.&                  SYSID(TESTZZ10)                    ;
.&     ADD          PTF                  FEATURE(6001E)
.&                  SYSID(TESTZZ10)                    ;
.&     ADD          PTF                  FEATURE(DOSF20)
.&                  SYSID(TESTZZ10)                    ;
.&     ADD          PTF                  LEVEL(R02M02)
.&                  SYSID(TESTZZ10)                    ;
.&     ADD          PTF                  FEATURE(6001F)
.&                  SYSID(TESTZZ10)                    ;
.&     ADD          PTF                  FEATURE(DOSF21)
.&                  SYSID(TESTZZ10)                    ;
.&     LIST         PTF                                ;
```

Figure 10. Adding retrieved PTFs to the host PTF library: these control statements will add PTFs retrieved from DPCX to the host PTF library. PTFs are copied from the host temporary PTF library, for each software level and feature requested.

At this point, the host PTF temporary library (TEMPLIB) may be deleted. The control statement in Figure 11 will accomplish this. If you have allocated a BHDWK2 dataset to contain the TEMPLIB you may take the faster option of deleting and redefining it with AMS, as described in the Host Prep SYSINFOREF Guide and Reference for Release 5.0 of Host Prep.

```
.&     DELETE       FILE                 TEMPLIB          . ;
```

Figure 11. Deleting the host TEMPLIB: this control statement will cause deletion of the entire contents of the host PTF temporary library, that is, all PTFs retrieved from diskette.


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
0
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

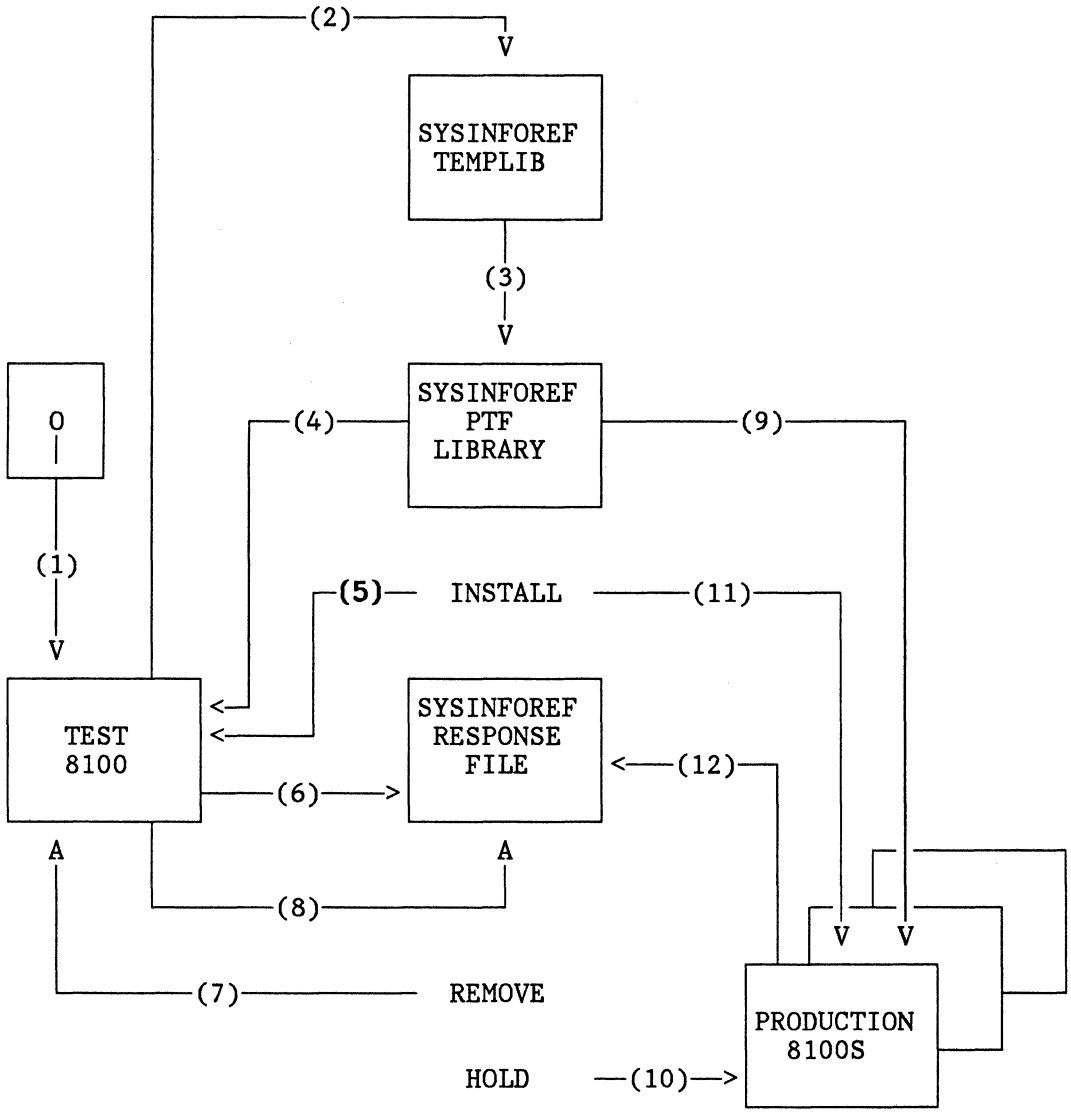
Step 4: Send the PTFs to the test DPCX system: Note that SEND PTF requests are generated by level or feature. Assuming that the test system is at the later software level, generation is required for level R02M02, feature 6001F and feature DOSF21[5]. The control statements shown in Figure 12 will do this. (If generation is performed for SYSID(TESTZZ10) only PTFs corresponding to level R02M02 are sent - features are not included[5].)

The diskette identification was determined from the LIST PTF statement in Figure 10 on page 27. Only the new PTFs which were retrieved from that diskette are sent. Specify ACTION(INSTALL) in order to be able to request installation of only the PTFs which have just been transmitted.

```
.&     GENERATE     LEVEL(R02M02)                              ;
.&     RETRIEVE     PTF                  MSG                   ;
.&     SEND         PTF
.&                  DISKETTE(CD20023)    ACTION(INSTALL)       ;
.&     END                                                     ;
.&     GENERATE     FEATURE(6001F)                             ;
.&     SEND         PTF
.&                  DISKETTE(CD20023)    ACTION(INSTALL)       ;
.&     END                                                     ;
.&     GENERATE     FEATURE(DOSF21)                            ;
.&     SEND         PTF
.&                  DISKETTE(CD20023)    ACTION(INSTALL)       ;
.&     RETRIEVE     PTF                  MSG                   ;
.&     END                                                     ;
.&     STARTCOM     SYSID(TESTZZ10)                            ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)       ;
```

Figure 12. Sending PTFs to the test system: these control statements will transmit to the test 8100 PTFs which were loaded from the diskette.

[5] For SYSINFOREF with Host Prep Release 5.0, it is possible to generate by SYSID and pick up the feature PTFs.


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

Step 5: Install the PTFs on the test DPCX system: Specifying the ACTION option causes only those PTFs flagged for installation to be installed. These will be the ones previously transmitted.

The PTF install process will not tolerate any concurrent activity in the DPCX system, including a host session with SYSINFOREF. SYSINFOREF therefor sets a flag in the DPCX system requesting installation; this can occur either at DPCX shutdown or at IPL time. If shutdown is chosen, the control operator must properly close down active tasks and request shutdown with option 4 of the SYSTERM System Service, otherwise PTF install takes place at IPL time, with attendant delay in DPCX initialisation. The control statements illustrated in Figure 13 will ask for installation at shutdown.

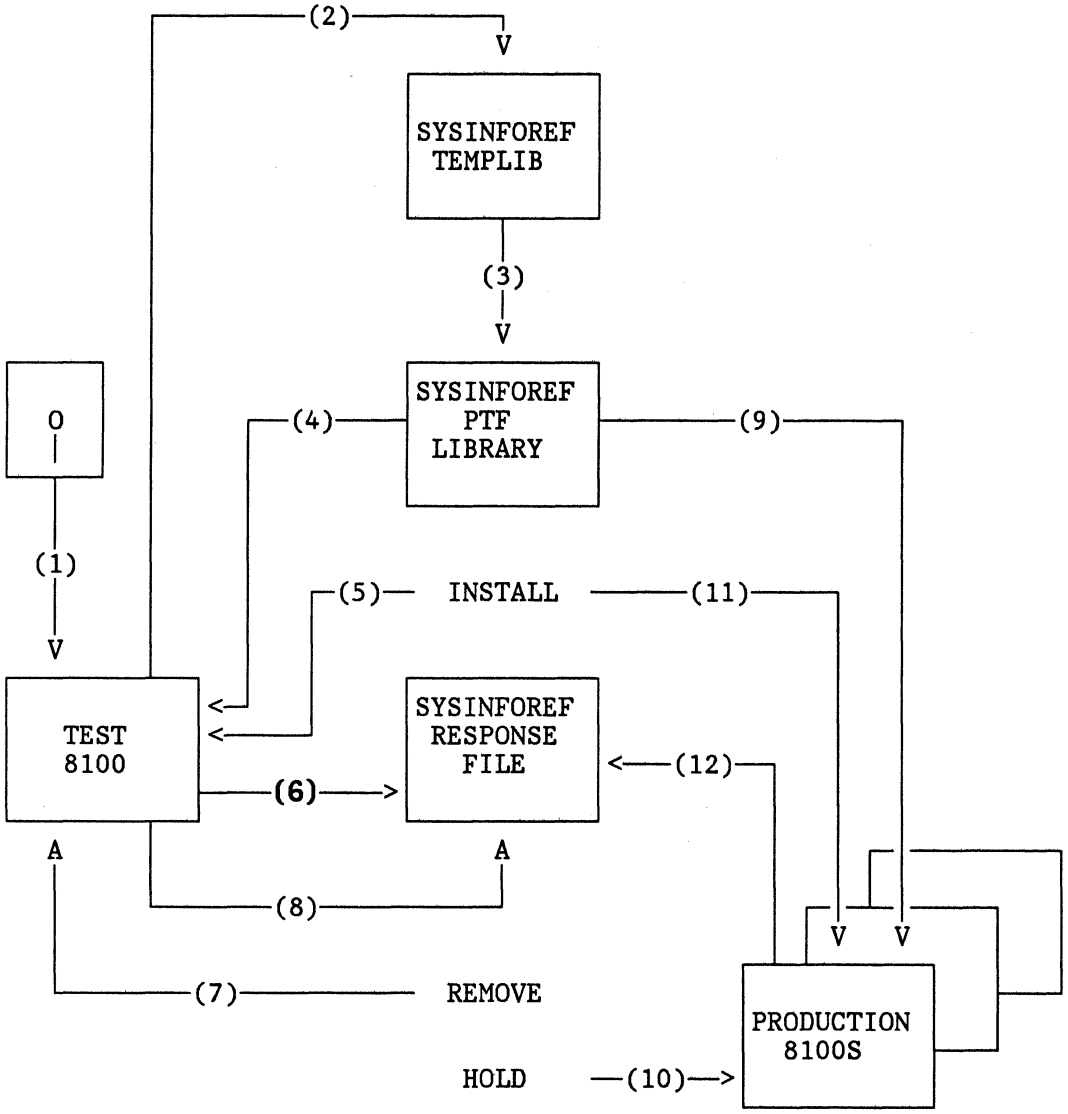
SYSINFOREF with Host Prep Release 5 or later can request an immediate IPL of DPCX; control statements to use this facility are shown in Figure 14. Take care that they are executed when no-one is using DPCX! Make sure that the communication phase of SYSINFOREF executes over-night.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     INSTALL      SHUTD                ACTION              ;
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)     ;
```

Figure 13. Installing PTFs at DPCX shutdown at the test system: these control statements will transmit to the test 8100 a request for PTF installation when all other tasks terminate.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     INSTALL      IPL                  ACTION              ;
.&     END          IPL                  IMMEDIATE           ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)     ;
```

Figure 14. Installing PTFs using immediate IPL at the test system: these control statements will transmit a request for PTF installation at IPL to the test 8100 and then cause the IPL (SYSINFOREF Release 5.0).


(2)
SYSINFOREF
TEMPLIB
(3)
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
(8)
(7)
REMOVE
HOLD
(10)
PRODUCTION
8100S

Step 6: Check the status of the PTF installation: Which ever method is adopted to effect the installation, control statements such as those in Figure 15 should be executed subsequently to determine the results of the installation.

- The PTF installation messages are a blow-by-blow account of what actually took place.

- The RETRIEVE PTF VERIFY causes the status of each PTF to be tested by a comparison of the verify and replace data in the PTF with the actual target data in the DPCX system.

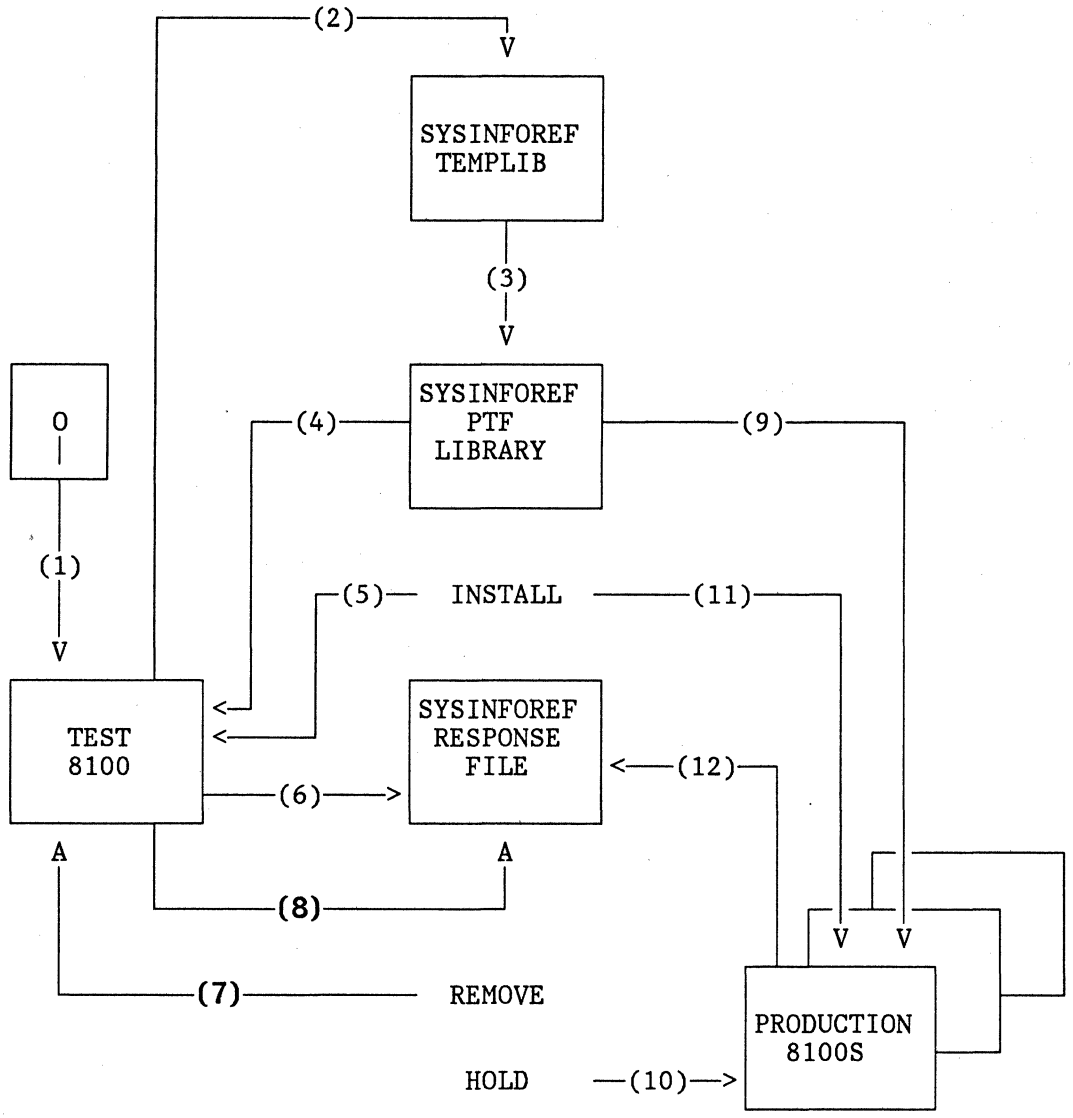
The results of this should be checked against those expected as appended to the Memo to Users distributed with the PTF diskette; this lists PTFs whose status is anticipated to be other than installed. Hopefully, the only other PTFs not appearing as installed will be those in a privilege/hold status, usually for national language features. This may be ascertained from the information produced by the following:

- The RETRIEVE PTF STATUS statement lists the status of each PTF as recorded in the PTF data set: whether or not installed, where it came from, whether held and what requisites it has.

This latter listing should be retained for future reference, as it records all the PTFs at the DPCX system.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     RETRIEVE     PTF                  MSG                 ;
.&     RETRIEVE     PTF                  VERIFY              ;
.&     RETRIEVE     PTF                  STATUS              ;
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)     ;
```

Figure 15. Determining the results of test system PTF installation: these control statements retrieve the messages generated by PTF installation from the test DPCX system, and test and list the status of PTFs there, and print the results from the response file.


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

Steps 7 and 8: Test the system and perform any necessary regression: The test 8100 should be used sufficiently to conclude that there is no impact on operation. If a PTF is determined to have caused a problem, on the advice of the IBM Support Centre it may be removed by:

- executing control statements such as those in Figure 16 which will remove PTF UC54321
- then invoking PTF installation as in Figure 13 on page 31 or Figure 14 on page 31
- and retrieving the removal message as in Figure 17

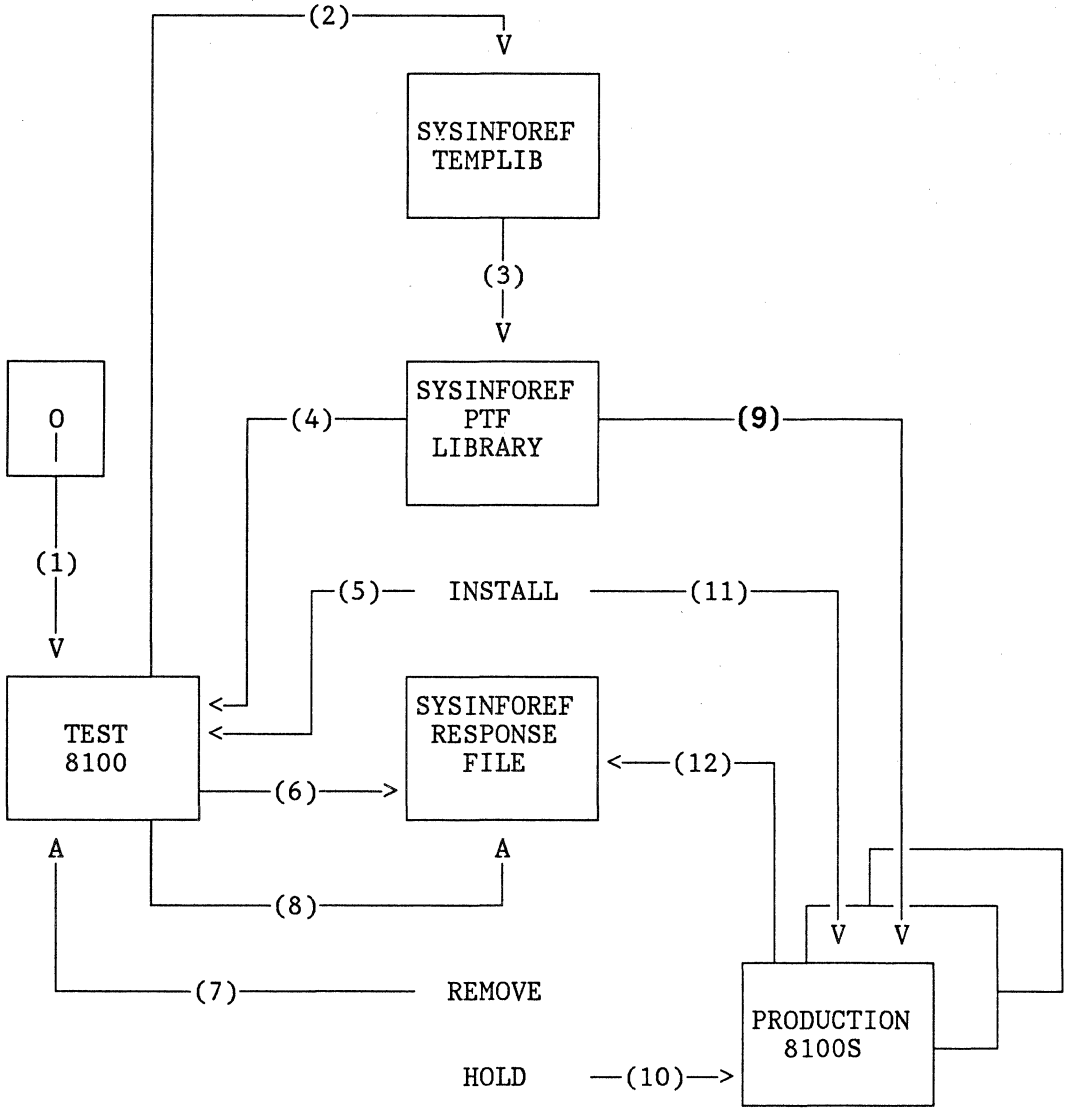
Specifying ACTION(HOLD) on the SYSPTF process prevents the PTF from being inadvertently re-installed. Note that a PTF cannot be deleted from the DPCX PTF dataset; it will remain until the next update install of a release of DPCX erases the contents of the PTF dataset.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     REMOVE       PTF(UC54321)         ACTION(HOLD)        ;
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)     ;
```

Figure 16. Removing an erroneous PTF: these control statements will request that the install process for PTF UC54321 on the test 8100 be reversed.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     RETRIEVE     PTF                  MSG                 ;
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)     ;
```

Figure 17. Determining the results of test system PTF removal: these control statements retrieve the messages generated by removing a PTF from the test DPCX system.

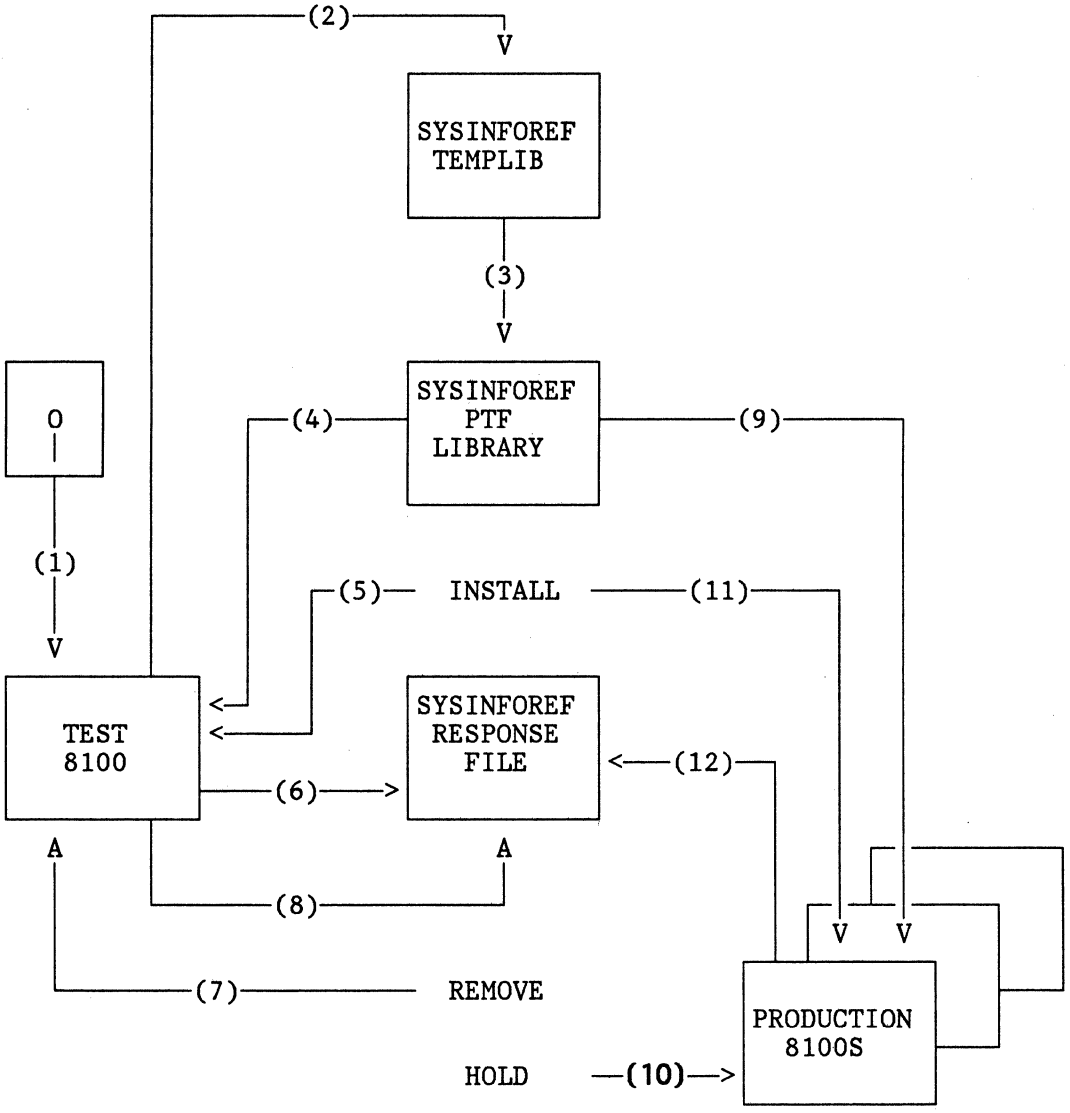

(2)
V
SYSINFOREF
TEMPLIB
(3)
V
O
(4)
SYSINFOREF
PTF
LIBRARY
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

Step 9: Send the PTFs to the production DPCX systems: Again generation is required for level R02M02, feature 6001F and feature DOSF21. By restricting generation to class 1, transmitting the PTFs a second time to the test 8100 (class 900) is avoided. Although there may be 8100s at the lower software level included in class 1, nothing will be sent to them as nothing was generated for their software levels or features. The sample control statements are in Figure 18.

The diskette identification is used again to transmit only the new PTFs, and ACTION(INSTALL) is specified in order to be able to request installation of only the PTFs which are being sent. The (10) operand of the STARTCOM indicates the number of transmissions which will be run concurrently by SYSINFOREF as subtasks. Increasing this up to the number of production systems will reduce the elapsed time of the job, but it may be necessary to keep it lower to contain the traffic loading on network resources.

```
.&     GENERATE      LEVEL(R02M02)        CLASS(1)              ;
.&     SEND          PTF
.&                   DISKETTE(CD20023)    ACTION(INSTALL)       ;
.&     END                                                      ;
.&     GENERATE      FEATURE(6001F)       CLASS(1)              ;
.&     SEND          PTF
.&                   DISKETTE(CD20023)    ACTION(INSTALL)       ;
.&     END                                                      ;
.&     GENERATE      FEATURE(DOSF21)      CLASS(1)              ;
.&     SEND          PTF
.&                   DISKETTE(CD20023)    ACTION(INSTALL)       ;
.&     END                                                      ;
.&     STARTCOM(10)  CLASS(1)                                   ;
.&     LIST          RESPONSE(ALL)                              ;
```

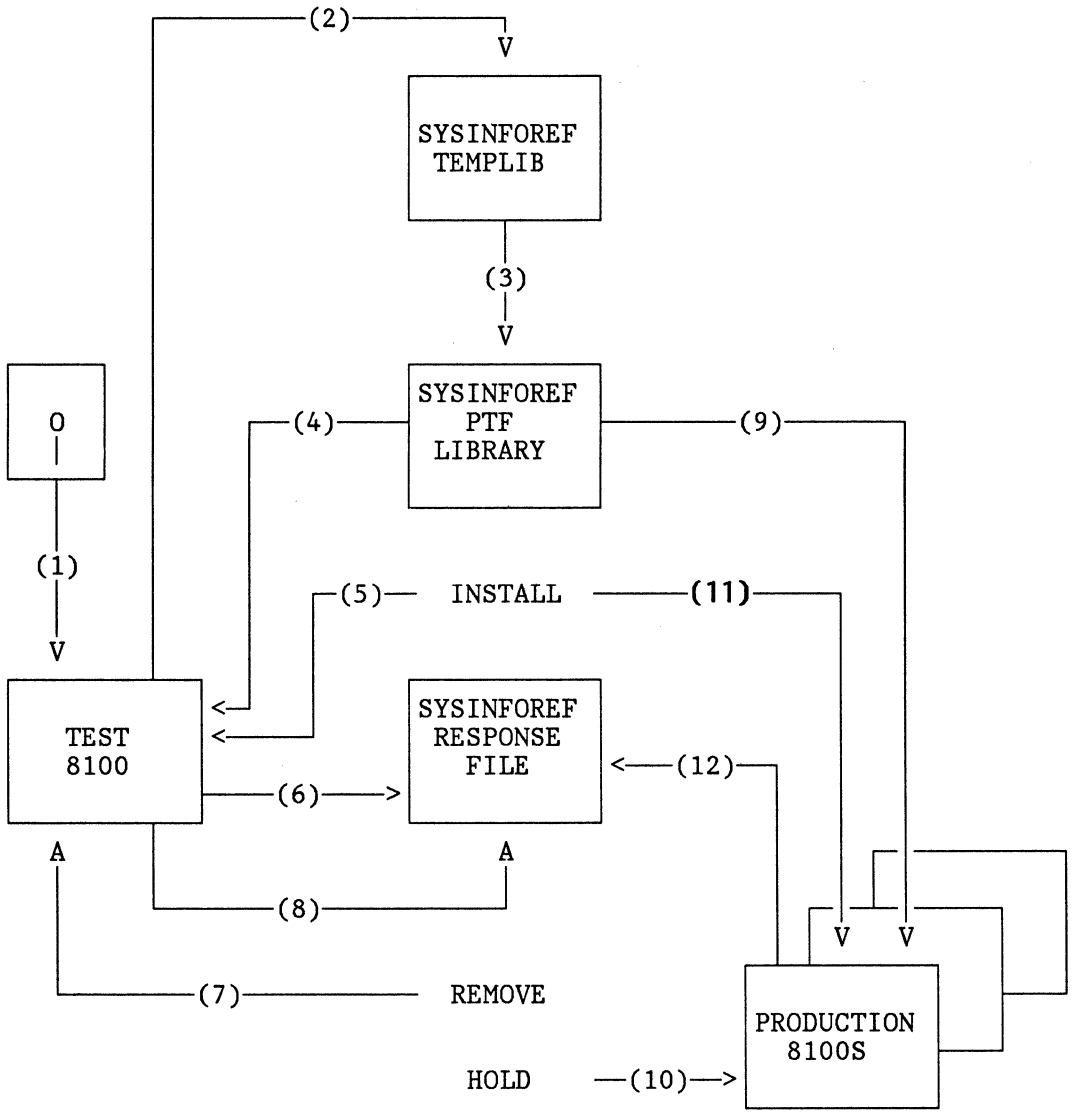
Figure 18. Sending PTFs to the production systems: these control statements will transmit the new PTFs to the production DPCX systems.


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

Step 10: Hold any erroneous PTFs: If any PTFs had to be backed off the test system, they should be placed in a HOLD status on the production systems before installing. The control statements in Figure 19 show how to do this for PTF UC54321. The UPDATE statement may instead be placed after the last SEND statement in Figure 18 on page 37.

```
.&     GENERATE      LEVEL(R02M02)        CLASS(1)              ;
.&     UPDATE        PTF(UC54321)         ACTION(HOLD)          ;
.&     END                                                      ;
.&     STARTCOM(10)  CLASS(1)                                   ;
.&     LIST          RESPONSE(ALL)                              ;
```

Figure 19. Placing an erroneous PTF in HOLD status: these control statements will place PTF UC54321 in HOLD status for each production DPCX system to prevent its installation.


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
O
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

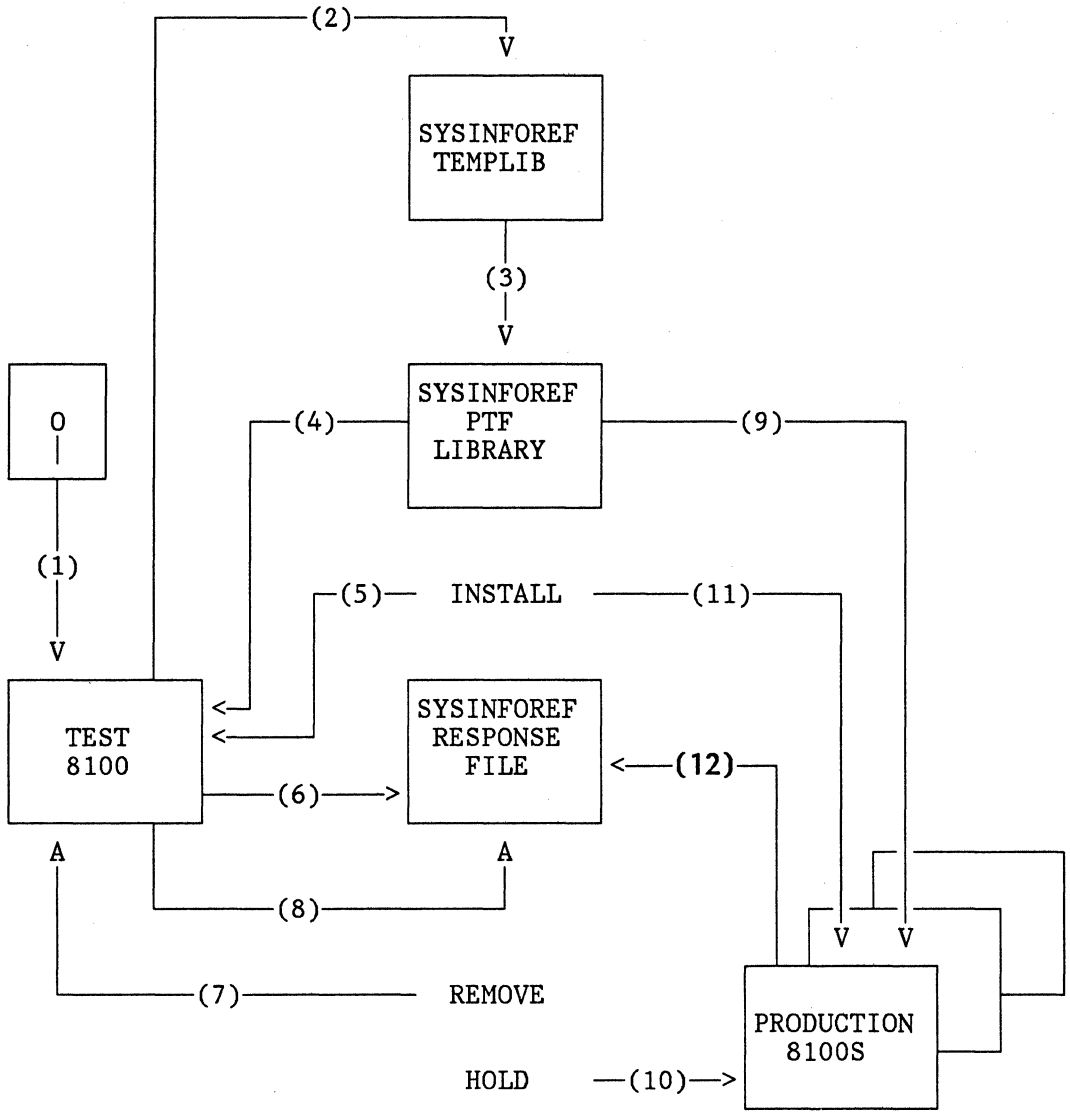
Step 11: Install the PTFs on the production DPCX systems.: Again the ACTION option causes only the new PTFs to be installed. Control statements to cause installation at DPCX shutdown are shown in Figure 20, and those (for SYSINFOREF with Host Prep Release 5.0 or later) to cause a DPCX IPL and subsequent installation in Figure 21. For installation to occur at DPCX shutdown, the control operator must properly close down active tasks and request shutdown with option 4 of the SYSTERM System Service.

```
.&     GENERATE     LEVEL(R02M02)        CLASS(1)              ;
.&     INSTALL      SHUTD                ACTION                ;
.&     END                                                     ;
.&     STARTCOM     CLASS(1)                                   ;
.&     LIST         RESPONSE(ALL)                              ;
```

Figure 20. Installing PTFs at DPCX shutdown at production systems: these control statements will transmit to the production DPCX systems requests for PTF installation when all other tasks terminate.

```
.&     GENERATE     LEVEL(R02M02)        CLASS(1)              ;
.&     INSTALL      IPL                  ACTION                ;
.&     END          IPL                  IMMEDIATE             ;
.&     STARTCOM     CLASS(1)                                   ;
.&     LIST         RESPONSE(ALL)                              ;
```

Figure 21. Installing PTFs using immediate IPL at production systems: these control statements will transmit to the production DPCX systems requests for PTF installation at IPL and then cause the IPL (SYSINFOREF Release 5.0).


(2)
V
SYSINFOREF
TEMPLIB
(3)
V
0
SYSINFOREF
PTF
LIBRARY
(4)
(9)
(1)
V
(5)
INSTALL
(11)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(12)
(6)
A
A
(8)
V
V
(7)
REMOVE
PRODUCTION
8100S
HOLD
(10)

Step 12: Determine the results of PTF installation: As with the test machine, so with the production systems, the statements in Figure 22 should be executed later to verify the results of the installation. See the comments under Step 6 above for the uses to which these listings should be put.

For systems at the same level, these listings should be similar, if maintenance has been applied consistently across the network.

```
.&     GENERATE      LEVEL(R02M02)        CLASS(1)              ;
.&     RETRIEVE      PTF                  MSG                   ;
.&     RETRIEVE      PTF                  VERIFY                ;
.&     RETRIEVE      PTF                  STATUS                ;
.&     END                                                      ;
.&     STARTCOM(10)  CLASS(1)                                   ;
.&     LIST          RESPONSE(ALL)                              ;
```

Figure 22. Determining the results of production system PTF installation: these control statements retrieve the messages generated by PTF installation from the production DPCX systems, and test and list the status of PTFs there, and print the results from the response file.

The preceding steps must be repeated to install the PTFs on the systems with the lower software levels. It is advisable either to:

- restore the equivalent system, dumped to tape before installing the higher software level, on the test system, and install first on the test system. The updated system would then be saved.

or to

- select one of the production systems at the lower level as a pilot system and use it in the same fashion as the test system, installing the PTFs on it first.

Eventually, when all DPCX systems are upgraded to the higher software levels, it will be possible to delete the PTFs for the earlier levels from the host PTF library. The control statements in Figure 23 will do that.

```
.&     DELETE       PTF                  LEVEL(R02M01)        ;
.&     DELETE       PTF                  FEATURE(6001E)       ;
.&     DELETE       PTF                  FEATURE(DOSF20)      ;
```

Figure 23. Deleting PTFs for redundant software levels: these control statements delete from the host PTF library copies of PTFs which applied to DPCX Release 2.1 and features 6001E and DOSF Release 2.0.

The procedure which has been described above will never quite be complete, since from time to time special instructions are supplied for the installation of PTF diskettes. For example, both the following have occurred:

- A requirement to run the PTF installation twice in order to install correctly a complicated sequence of inter-dependant PTFs.

- A need to install one PTF before the rest of those on the diskette.

Also, if you use a national language feature, if the Memo to Users which accompanies the PTF diskette indicates that a PTF applicable to your feature is included, you will need to take it out of HOLD status and install it. Because it will also be flagged as privileged, the installation must be performed using the DPCX SYSPTF System Service. If host communications are enabled so to as allow only one batch (host-initiated) session, this may be done via HCF, using the signon for operators 1 or 255, but all other users must be signed off and tasks stopped. Use option 8 of SYSPTF to enable the PTF, and option 1 to install it.

3.2.3 INSTALLING CORRECTIVE SERVICE TO SYSTEM SOFTWARE

This section suggests a procedure you may adopt to apply an individual PTF, which is required to fix a specific problem on a DPCX system, to DPCX and DOSF throughout the network. If the PTF has no adverse effect on system operation, installation of the PTF throughout the network is recommended, as this avoids the possibility of the problem recurring elsewhere, and overall ease of maintenance is enhanced by having as few as possible different software levels in the network. The PTF is retained on the host PTF library while the software level to which it applies is current anywhere in the network. It is assumed that SYSINFOREF has been initialised as in "Initialising SYSINFOREF" on page 127. Examples of control statements are included - the JCL is as in Figure 61 on page 127.

A PTF for corrective service will normally be in the form of a list of changes which have to be manually entered on to the host PTF library.

The steps entailed are as follows; Figure 24 on page 46 provides a schematic representation.

1. Use SYSINFOREF to add the PTF to the host PTF library using the supplied source information.

2. Send the PTF to the test DPCX system[6].

3. Install the PTF on the test system.

4. Retrieve the PTF installation messages.

Verify the operation of the test DPCX system - if it is impacted the problem should be referred to the IBM Support Centre.

5. Send the PTF to the DPCX system which encountered the problem.

6. Install the PTF on this system.

7. Retrieve the PTF installation messages.

Verify the operation of this 8100 - if the problem has not been resolved it should be referred to the IBM Support Centre.

8. Send the PTF to the other production systems.

9. Install the PTF on these systems.

10. Retrieve the PTF installation messages.

The examples assume that there are some DPCX systems in the network with DPCX Release 2.2, feature 6001F and DOSF Release 2.1 and it is on DOSF on one of these, the system whose SYSID is TEXTAD10, that a problem has been encountered. The test system is at this level. Furthermore, the production systems have been defined to the SYSINFOREF machine list as being in class 1 and the test 8100 as being in class 900. PTF UC12345 is to be installed.

---

[6] If the problem is urgent and severely impacting the affected 8100, it may be preferable to send first to that machine and test the PTF there.

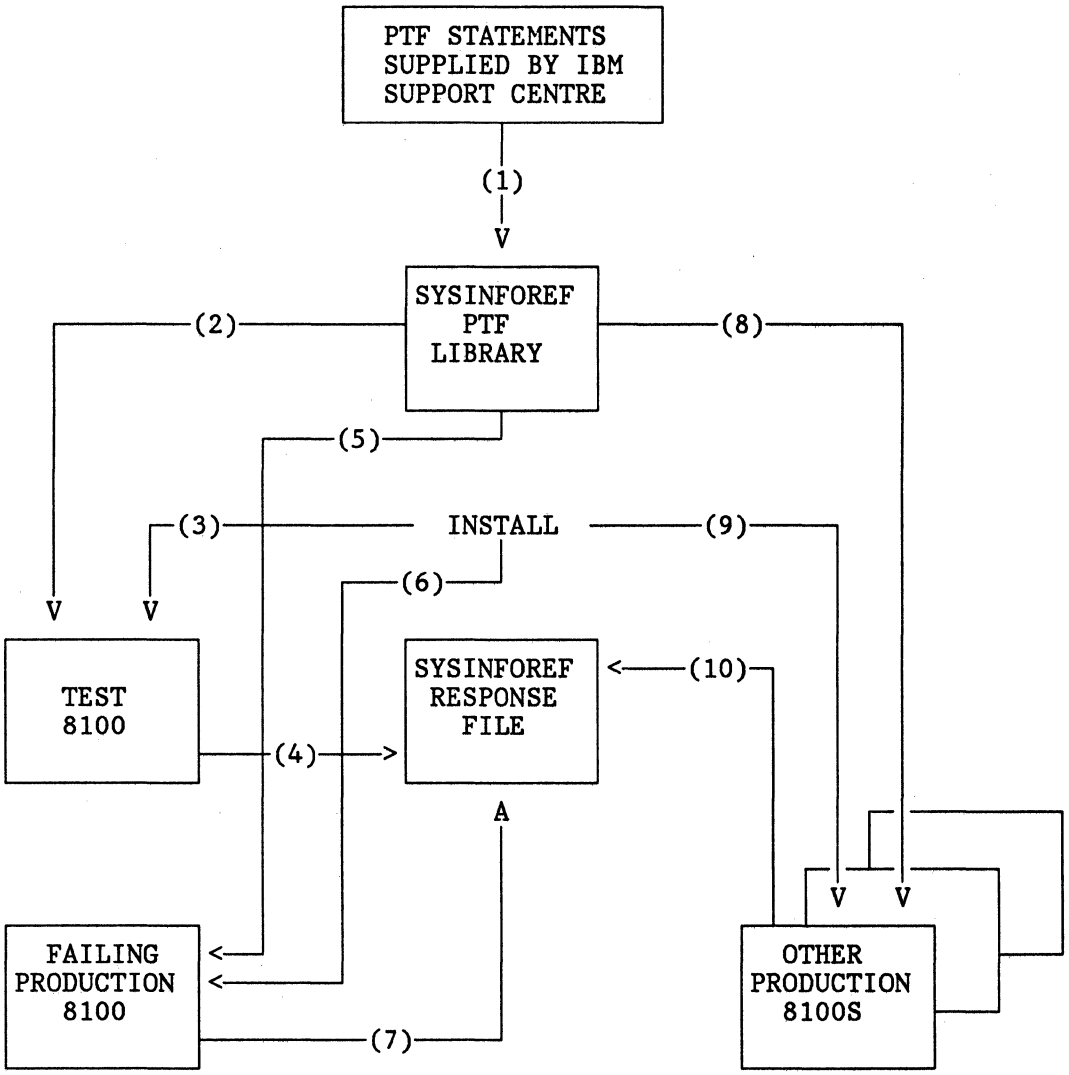
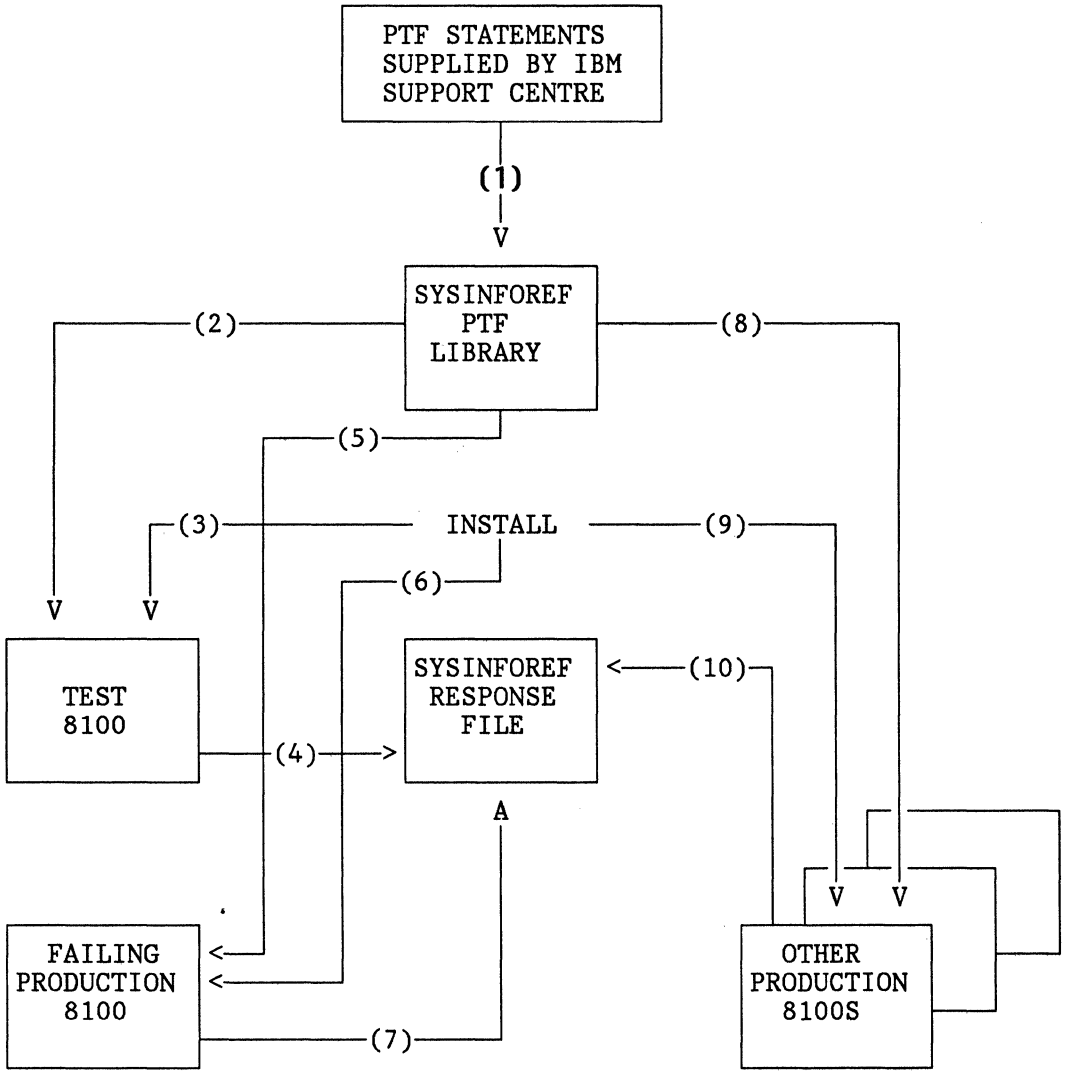


Figure 24. Network application of corrective service: this is a schematic of the processes involved in installing a diskette of PTF supplied by the IBM Support Centre across the network.

For convenience of illustration, the sample control statements show the SYSINFOREF maintenance, generation, communication and response phases being executed consecutively within a single jobstream. As described in "SYSINFOREF Phases and Timing of Sessions" on page 128, it may well be desirable to separate the execution of the different phases.

THIS PAGE LEFT INTENTIONALLY BLANK


PTF STATEMENTS
SUPPLIED BY IBM
SUPPORT CENTRE
(1)
V
SYSINFOREF
PTF
LIBRARY
(2)
(8)
(5)
(3)
INSTALL
(9)
(6)
V
V
TEST
8100
SYSINFOREF
RESPONSE
FILE
(10)
(4)
A
V
V
FAILING
PRODUCTION
8100
OTHER
PRODUCTION
8100S
(7)

Step 1: Create the PTF on the host PTF library: The control statements in Figure 25 are an example. If the PTF had been for DPCX instead of DOSF, LEVEL(R02M02) would have been coded in place of FEATURE(DOSF21). The PTF number, source statements and the CHECKSUM value will be supplied by the IBM Support Centre. Pre-requisite and co-requisite information might also need to be coded on the control statement.

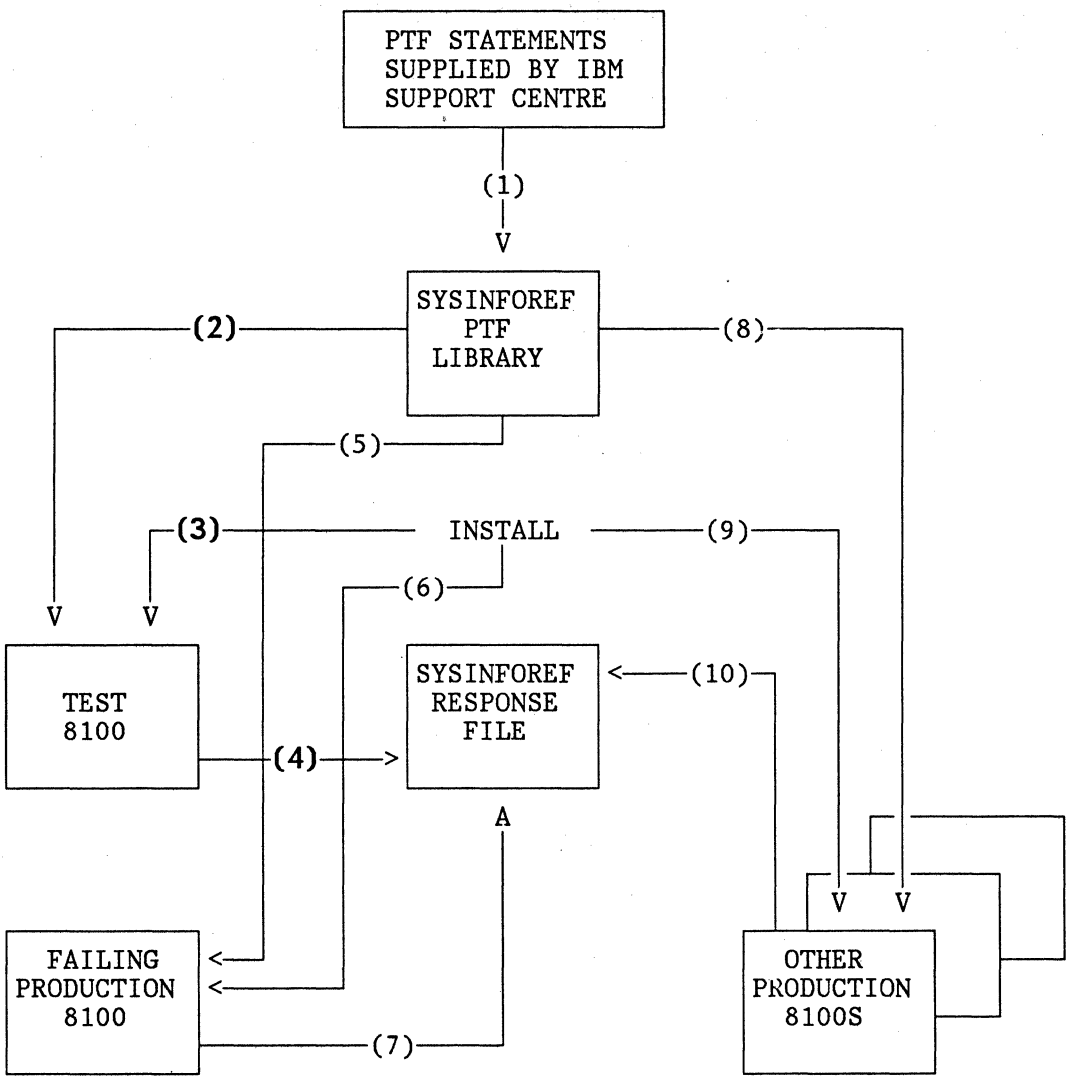
```
.&      ADD             PTF(UC12345)          FEATURE(DOSF21)
.&                      CHECKSUM(cccc)                        ;
1 nn mmmm bbbb
2 dd cccccccc rrrrrrrr
2    cccccccc rrrrrrrr
  . . . . . . . . . .
  . . . . . . . . . .
.&      LIST            PTF(UC12345)          FEATURE(DOSF21)
.&                      TEXT                                  ;
```

Figure 25. Adding a PTF to the host PTF library with source statements: these control statements will add PTF UC12345 to the host PTF library from source statements supplied by the IBM Support Centre.

If the PTF had a pre-requisite, ensure that this is either installed on DPCX or available on the SYSINFOREF library; in the latter case the pre-requisite PTF will be transmitted and installed along with the other. If the host PTF library was listed as in Figure 10 on page 27 when preventive service was last applied, and if PTF listings were retrieved from the network as in Figure 22 on page 43, then this information may already be available. If not, then the control statements in Figure 26 show how to elicit it for pre-requisite PTF UC01234. If a pre-requisite PTF is missing, DPCX does not issue a warning message and you will be left wondering why your PTF would not install.

```
.&      GENERATE        FEATURE(DOSF21)                       ;
.&      RETRIEVE        PTF(UC01234)          STATUS          ;
.&      END                                                   ;
.&      STARTCOM(10)    NETWORK                               ;
.&      LIST            RESPONSE(ALL)                         ;
.&      LIST            PTF(UC01234)          FEATURE(DOSF21) ;
```

Figure 26. Determining the status of a pre-requisite PTF: these control statements will show if a pre-requisite PTF is installed on each DPCX system in the network with the given software level, and whether it is present on the host library.


PTF STATEMENTS SUPPLIED BY IBM SUPPORT CENTRE
(1)
SYSINFOREF PTF LIBRARY
(2)
(8)
(5)
(3)
INSTALL
(9)
(6)
TEST 8100
(4)
SYSINFOREF RESPONSE FILE
(10)
FAILING PRODUCTION 8100
(7)
OTHER PRODUCTION 8100S

Steps 2, 3 and 4: Send the PTF to the test DPCX system and install it: Note that the SEND PTF request must be generated by the appropriate level or feature[7]. The control statements shown in Figure 27 will do this. (Prior to Host Prep R5, if generation is performed for SYSID(TESTZZ10) the PTF would only be sent if it applied to DPCX level R02M02.)

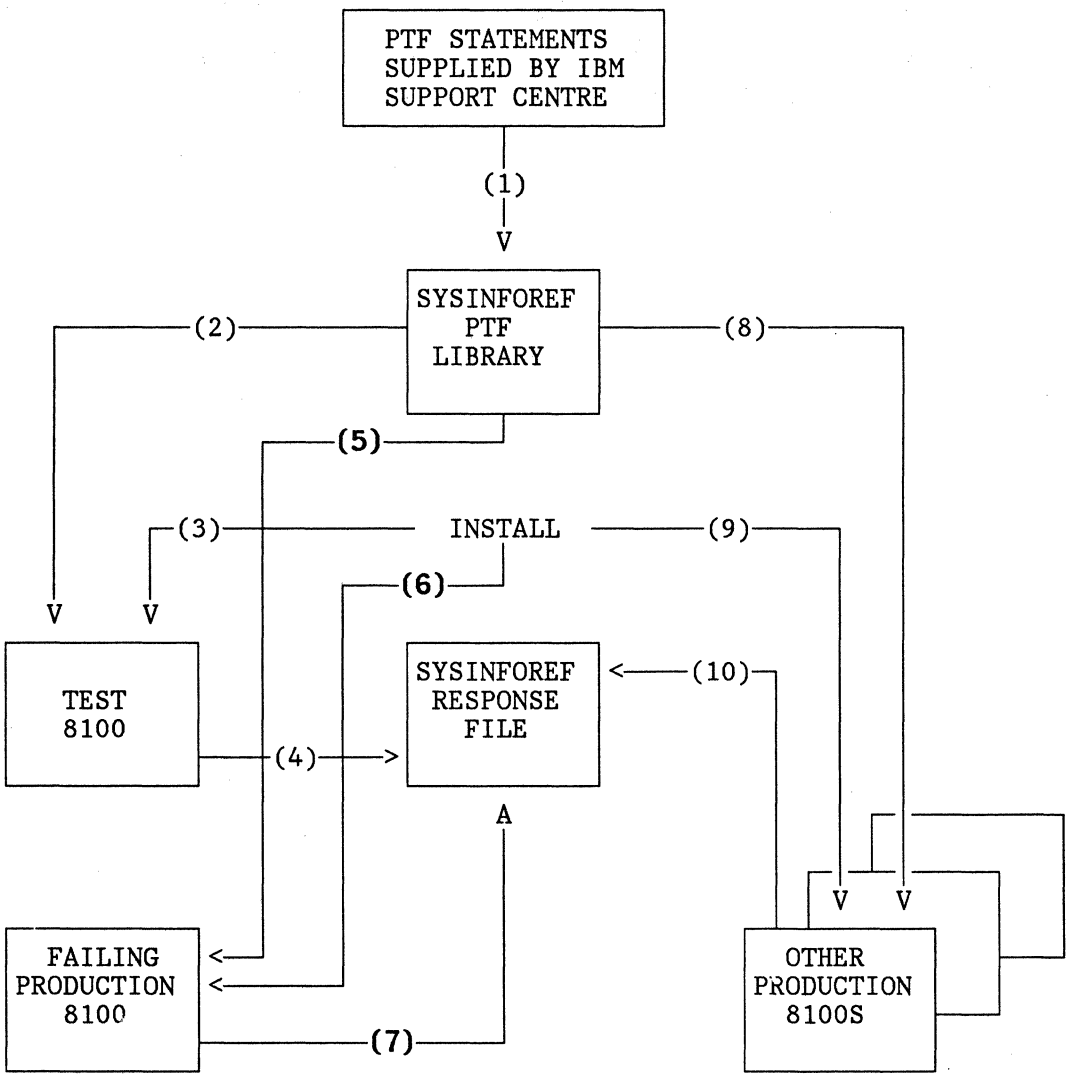
Specify ACTION(INSTALL) and INSTALL ACTION in order to request installation of only the PTF which has just been transmitted. The PTF will be installed at the next DPCX IPL.

```
.&     GENERATE     FEATURE(DOSF21)                          ;
.&     SEND         PTF(UC12345)          ACTION(INSTALL)    ;
.&     INSTALL      IPL                   ACTION             ;
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)         SYSID(TESTZZ10)    ;
```

Figure 27. Sending a PTF to the test system: these control statements will transmit PTF UC12345, created at the host, to the test 8100.

Once the test system has been IPLed, the control statements shown in Figure 17 on page 35 should be executed to retrieve the PTF installation messages.

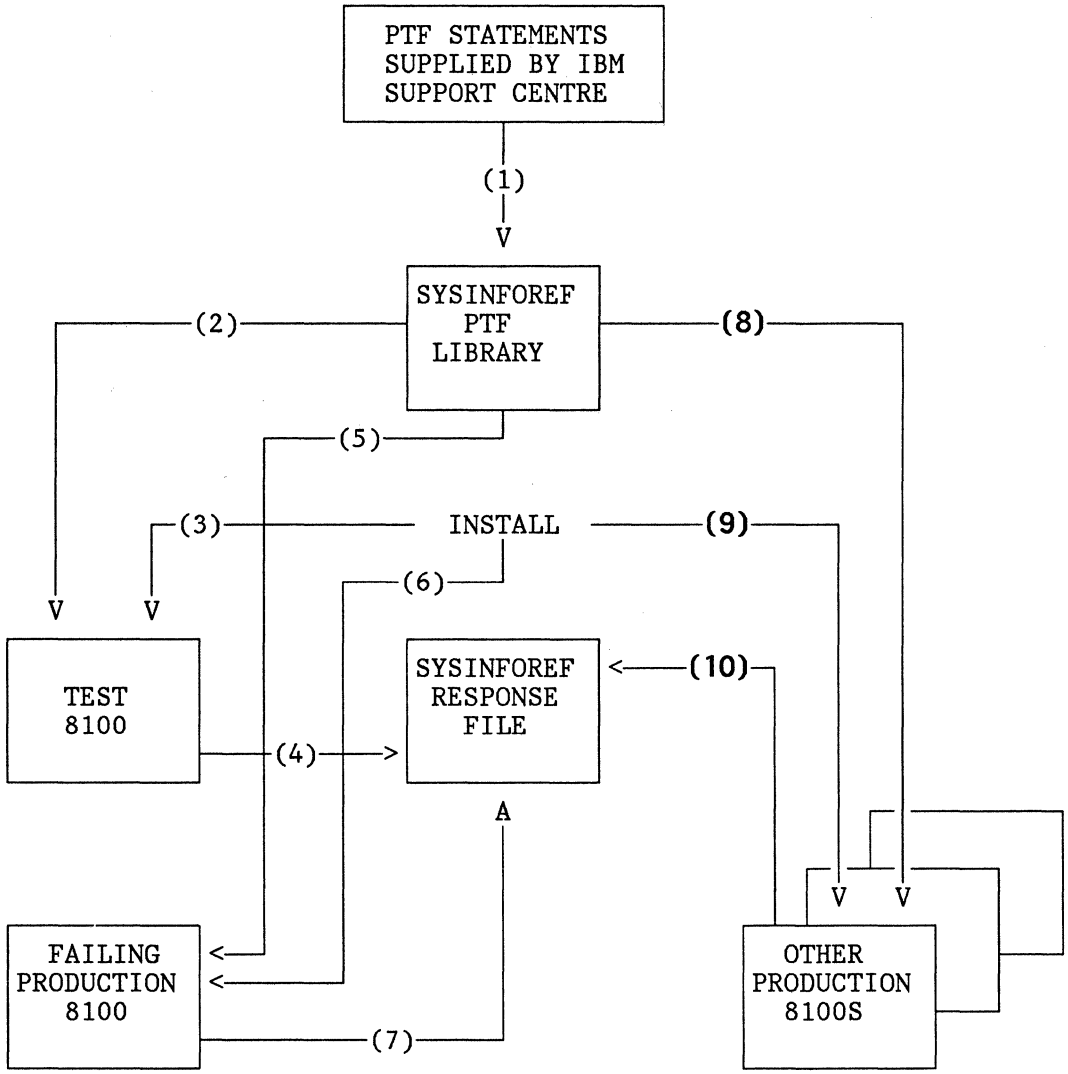
[7] For SYSINFOREF with Host Prep Release 5.0, it is possible to generate by SYSID and pick up the feature PTF.


PTF STATEMENTS
SUPPLIED BY IBM
SUPPORT CENTRE
(1)
V
SYSINFOREF
PTF
LIBRARY
(2)
(8)
(5)
(3)
INSTALL
(9)
(6)
V
V
TEST
8100
SYSINFOREF
RESPONSE
FILE
(10)
(4)
A
V
V
FAILING
PRODUCTION
8100
OTHER
PRODUCTION
8100S
(7)

Steps 5, 6 and 7: If it works, send the PTF to the affected system: The operation of the test DPCX system should be evaluated - if the PTF causes no adverse effect, it may be sent to DPCX system TEXTAD10 which first encountered the problem. If the problem was reproducible on the test system, it may be possible to tell at this point if the PTF will fix it.

The PTF is sent to the production DPCX system TEXTAD10 and installed. The same control statements as previously shown in Figure 27 on page 51 are used, the only change being on the STARTCOM statement to change SYSID(TESTZZ10) to SYSID(TEXTAD10). The PTF will be installed at the next DPCX IPL.

After IPLing this system, execute the control statements in Figure 17 on page 35 to retrieve the PTF installation messages; the SYSID value should be TEXTAD10.


PTF STATEMENTS
SUPPLIED BY IBM
SUPPORT CENTRE
(1)
SYSINFOREF
PTF
LIBRARY
(2)
(8)
(5)
(3)
INSTALL
(9)
(6)
TEST
8100
SYSINFOREF
RESPONSE
FILE
(10)
(4)
FAILING
PRODUCTION
8100
(7)
OTHER
PRODUCTION
8100S

Steps 8, 9 and 10: If successfull, send the PTF to the other systems: If the PTF is verified to have fixed the problem, it should be sent to the other 8100s in the network at the same software level. The control statements in Figure 28 will do this. Note that they will try and send the PTF again to the system with SYSID of TEXTAD10. The request for this machine will fail, and nothing will be installed on this system.

```
.&     GENERATE     FEATURE(DOSF21)                          ;
.&     SEND         PTF(UC12345)          ACTION(INSTALL)    ;
.&     INSTALL      IPL                   ACTION             ;
.&     END                                                   ;
.&     STARTCOM     CLASS(1)                                 ;
.&     LIST         RESPONSE(ALL)                            ;
```

Figure 28. Sending a PTF to the other production systems: these control statements will transmit PTF UC12345, created at the host, to other 8100s in the network at the specified software level.

Messages from the installation process should subsequently be retrieved with the control statements in Figure 29.

```
.&     GENERATE     FEATURE(DOSF21)       CLASS(1)           ;
.&     RETRIEVE     PTF                   MSG                ;
.&     END                                                   ;
.&     STARTCOM     CLASS(1)                                 ;
.&     LIST         RESPONSE(ALL)                            ;
```

Figure 29. Retrieving PTF installation messages from the other 8100s: these control statements retrieve the messages generated by PTF installation from the production DPCX systems and print them from the response file.

If the PTF in fact had an adverse effect on a system, or did not fix the problem, it can be removed using control statements such as those in Figure 16 on page 35, referencing the appropriate 8100.

3.2.4 INSTALLING PTFS AFTER AN INITIAL OR UPDATE INSTALL

When a new DPCX/DOSF system is installed, you will want to ensure that the software is at the same level as that on the currently installed systems. Also, if DPCX and DOSF are update installed on an 8100 to a new release, the contents of the DPCX PTF dataset are purged. You will need to send any PTFs, now applicable to the new release, to the 8100. Often you will receive a new PTF diskette with a new release of DPCX and DOSF.

The procedure to follow is conceptually the same as that in "Installing Preventive Service to System Software" on page 21, but there are some differences. The following cases are considered:

1. A new 8100 has been installed at the same level of DOSF and DPCX as currently in use at other systems in the network. It is desired to install those PTFs currently installed on other 8100s.

2. Some 8100s have been update installed to a new release of DPCX and DOSF. It is desired to install those PTFs which apply to the new level of the system software.

Case 1 - A new 8100 with the current release of DPCX and DOSF: Follow steps 9 through 12 of "Installing Preventive Service to System Software" on page 21:

Step 9: The new system requires all PTFs, not just those from the latest diskette. Figure 30 contains appropriate control statements; TEXTAK10 is the batch logical unit of the new 8100.

```
.&     GENERATE     LEVEL(R02M02)                          ;
.&     SEND         PTF                 ACTION(INSTALL)    ;
.&     END                                                 ;
.&     GENERATE     FEATURE(6001F)                         ;
.&     SEND         PTF                 ACTION(INSTALL)    ;
.&     END                                                 ;
.&     GENERATE     FEATURE(DOSF21)                        ;
.&     SEND         PTF                 ACTION(INSTALL)    ;
.&     END                                                 ;
.&     STARTCOM     SYSID(TEXTAK10)                        ;
.&     LIST         RESPONSE(ALL)       SYSID(TEXTAK10)    ;
```

Figure 30. Sending PTFs to a new 8100: these control statements will transmit all applicable PTFs to a new DPCX/DOSF system.

Step 10: Don't forget to hold any PTFs which you have not installed on or removed from other systems. Change the STARTCOM in Figure 19 on page 39 to specify SYSID(TEXTAK10) instead of CLASS(1).

Steps 11 and 12: Are the same as in "Installing Preventive Service to System Software" on page 21, the only alteration being to change the STARTCOM in Figure 20 or Figure 21 on page 41 and in Figure 22 on page 43 to specify SYSID(TEXTAK10) instead of CLASS(1).

Case 2 - Current 8100s with a new release of DPCX and DOSF: Follow steps 1 through 12 of "Installing Preventive Service to System Software" on page 21. It is assumed that the new level of code will be installed first on the test 8100 and that the PTFs will be sent first to this system. Don't forget to retain a tape copy of the DPCX/DOSF system from the test 8100 at the existing level for as long as this release is current anywhere in the network. The software levels in figures referenced in the steps cited below will of course need to be changed to reflect the new software release.

Steps 1, 2 and 3: If there is a new PTF diskette supplied with the new software release, you will need to retrieve the latest PTF diskette to the host and copy applicable PTFs into the host PTF library.

Step 4: Be sure to send all applicable PTFs: omit the DISKETTE() parameter from the SEND PTF statement in Figure 12 on page 29.

Steps 5 through 8: These are performed exactly as in "Installing Preventive Service to System Software."

Step 9: Be sure to send all applicable PTFs: omit the DISKETTE() parameter from the SEND PTF statement in Figure 18.

Steps 10, 11 and 12: Are performed exactly as in "Installing Preventive Service to System Software" on page 21.

3.2.5 USING HCF WITH THE SYSPTF SYSTEM SERVICE

As mentioned in "Overview of DPCX/DOSF PTF Handling" on page 134, the DPCX System Service SYSPTF may also be used for processing PTFs. All of its functions are available at the host using Host Command Facility; but as in order to install PTFs all other tasks must be stopped and only one host-initiated session enabled, its use for this purpose during office hours is somewhat impracticable. Moreover, either the control operator or service representative signon must be used to be able to invoke SYSPTF, with possible operational implications. However, HCF may be useful for the following purposes (which are not constrained by the need to be the only task active):

- Listing or verifying the status of PTFs installed on a particular 8100, especially in order to browse, for instance, a chain of pre-requisite PTFs. This function should not often be needed if records are kept as suggested in "Recording Changes to System Software," and a listing of PTF status retrieved after each application of preventive maintenance.

- Enabling or disabling a PTF on production 8100s, the SYSINFOREF control statements for which were illustrated in Figure 19 on page 39. Note, though, that the more 8100s there are in the network, the more efficient it is to use SYSINFOREF.

There is also one instance in which SYSPTF must be used, and that is:

- To install a privileged PTF. Privileged PTFs are only installed on instruction from the IBM Support Centre, but if it is necessary to install one it may be done using HCF (with all other tasks stopped, operators signed off and only one host-initiated session enabled).

3.3 RECORDING CHANGES TO SYSTEM SOFTWARE

A log should be kept for each DPCX system of software updates installed. A sample form for this can be found in "Sample Forms" on page 169. In use it might resemble the example in Figure 31 on page 58.

System 8100 SYS B	at RALEIGH	Batch LU TEXTAB10
Product Identifier	Update level, Fix Package or PTF id.	Date Installed
DPCX 5761-DS1	RELEASE 2.2	08AUG82
DPCX #6001 5761-DS1	LEVEL F	08AUG82
DOSF 5761-XR1	RELEASE 2.1	08AUG82
DPCX / #6001 / DOSF	FIX PACKAGE 23	12AUG82
DPCX #6001 F	PTF UC12345	19SEP82
. . . . .		
. . . . .		

Figure 31. Recording system software updates: this is how a log could be kept of installation of service to system software on a DPCX/DOSF system.

4.0 NETWORK MANAGEMENT OF APPLICATION SOFTWARE

There are four techniques available for software management of DPCX application software. They are appropriate both for installing initial and subsequent versions of IBM and user programs, and for managing PTFs for IBM programs. They are:

1. SSS (Subsystem Support Services)

2. Host Prep BDES (Batch Data Exchange, 5735-XR3)

3. Host Prep SYSINFOREF (Subsystem Information Retrieval Facility, 5735-XR3)

4. DSX (Distributed Systems Executive, 5748-XXG or 5668-986)

To use any of these products to manage DPCX application software, function such as is contained in Host Prep PVS (Program Validation Services, 5735-XR3) is pre-requisite.

Use of BDES requires manual intervention at the distributed system sites. SSS is the original network management tool; both SYSINFOREF and DSX are more efficient and are to be preferred. Of the latter two products the more appropriate technique for management of DPCX application software depends on a few simple factors:

- Is the number of DPCX systems more, or likely to become more, than about twelve?

- Is DSX already installed (perhaps for use with 8100/DPPX)?

- Are data applications to run under DPCX that require the host system to update DPCX datasets?

- Do you wish to use the Network Installation Management (NIM) capability of DPCX release 3?

If the answer to any of these questions is 'yes', then DSX is the preferred network management tool. DSX maintains an inventory of the software installed at each distributed system, and has better reporting capabilities and ease-of-use features. DSX will also be needed if host update of DPCX datasets is required, or in order to use NIM.

If, however, the answer to all these questions is 'no', then SYSINFOREF can do the job and avoid the complexity of installing another product. SYSINFOREF will probably be wanted already to maintain system software.

4.1 OVERVIEW OF DPCX APPLICATION SOFTWARE INSTALLATION

Application programs and panels in source form have to be assembled using the macros supplied with the Host Prep product. The assembler object module is supplied as input to Host Prep PVS, and, using the READY control statement, output is created on the BQBLIBI[8] dataset. DSCBs are defined

to PVS using control statements and likewise written to BQBLIBI[8].SYSINFOREF (or DSX) uses this dataset to update its library. This flow is illustrated in Figure 32 on page 61.

The SYSINFOREF library is a repository for all application entities (programs, panels and DSCBs) to be sent to DPCX/DOSF systems. Objects are keyed on the library by the number of the program, panel or DSCB; not by, for instance, program name. Different versions and levels, assigned by the PVS READY control statement, are supported by SYSINFOREF. Care should be taken to

- Retain in the SYSINFOREF library each version/level of each entity, either currently in use or potentially needed for fall-back, on any 8100 in the network,

and conversely

- Not to delete anything from the SYSINFOREF library until it is no longer in use, or possibly going to be needed, anywhere.

4.2 USING SYSINFOREF VERSIONS AND LEVELS FOR APPLICATIONS

Careful use of version and level is the key to easy software management. Conceptually, application entities that come from the same place and should be installed together (for example, all programs on a PTF tape) should have the same version and level numbers.

For IBM-supplied products:

- Use the version number in some meaningful way to represent the release number of the product
- Set the level number to 0 when installing a new release
- Increment the level number to refer to successive applications of preventative maintenance (for example, PTF tapes)
- For application of corrective service, such as APAR fixes, use level numbers greater than 50.

Some exceptions will have to be made:

- DSCBs for a RSDS and its indices are keyed under the same name. Assign a version, and use level 0 for the base RSDS, level 1 for index 1 up to level 8 for index 8. Fortunately DSCBs are not likely to be altered by the application of software maintenance.
- An application (DISOSS is an example) may have one or more programs or panels customised to specific controllers. In this case, use the level number to denote copies for different DPCX/DOSF systems.

For User-coded Applications:

- Set the version to 1 and the level to 0 initially

---

[8] In Host Prep Release 5.0, PVS can write its output directly to the SYSINFOREF or BDES library.

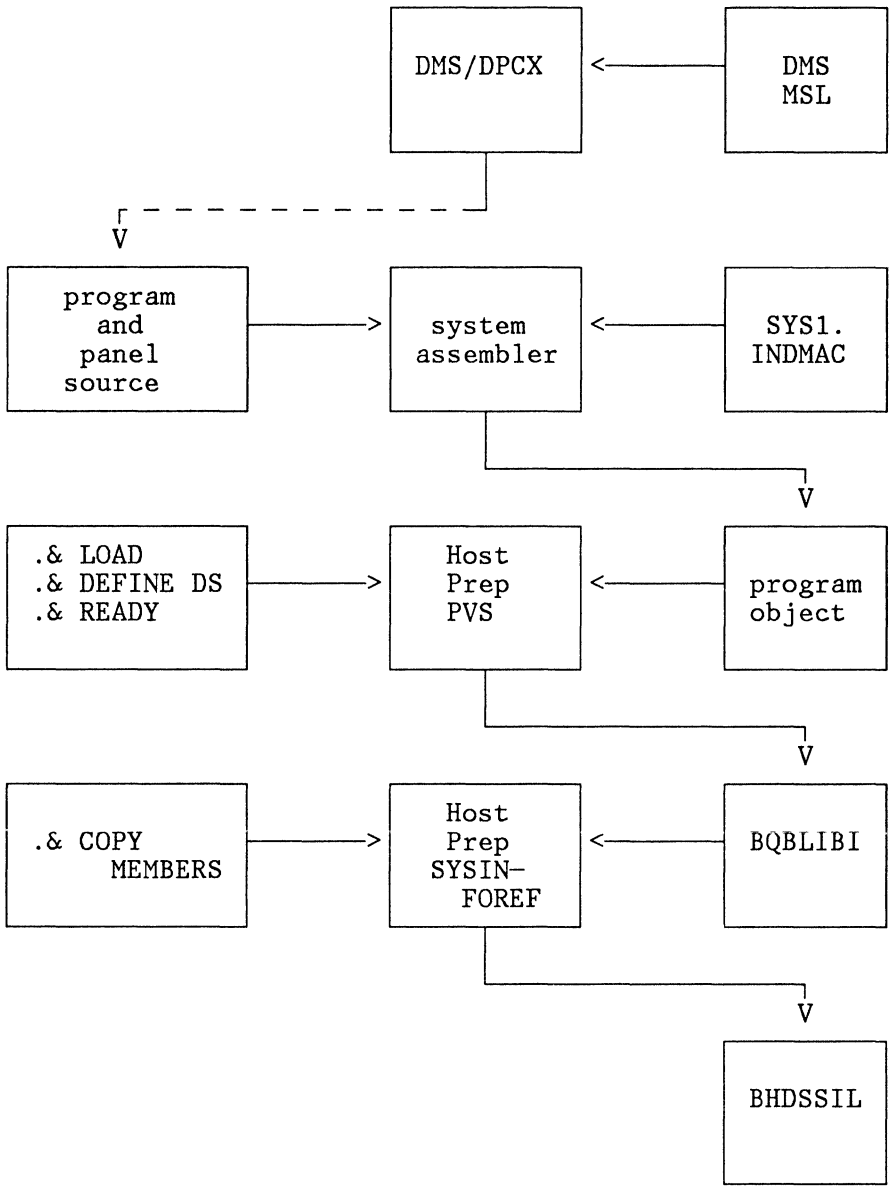


Figure 32. Application software generation: this summarises the flow of application entities from source statements to the SYSINFOREF library.

- If the function of the application is enhanced, changing and adding a lot of modules, make this a new version: change the version number and set the level number to 0
- When errors are corrected in modules, increment the level numbers of the new modules
- If the application needs to differ from 8100 to 8100, see if this can be achieved with common programs driven, for instance, by a customised control record in a dataset. If this is not possible, try to assign the slightly different programs different program numbers. Maintain-

ing functionally different versions of the same application in the SYSINFOREF library is likely to lead to confusion.

4.3 MAINTAINING APPLICATION SOFTWARE

This section suggests a procedure to adopt in order to apply service to application software across the network. There is no difference in the technique employed whether preventative or corrective service is being applied, or, indeed, if the product is being installed for the first time; though if a corrective service PTF is being applied, you may want to install the changed module on the DPCX system with the problem before releasing it to the rest of the network.

This procedure is designed to retain a central library of all levels of applications current anywhere in the network. It assumes that SYSINFOREF has been initialised as described in "Initialising SYSINFOREF" on page 127. Examples of control statements are included. Use the same JCL as that in Figure 61 on page 127.

The steps are as follows. They are illustrated in Figure 33 on page 63.

1. Load the application entities with PVS, write them to BQBLIBI[9] and copy them to the SYSINFOREF BHDSSIL dataset.

2. Send the modules to the test DPCX System.

3. Verify the operation of this 8100. If the new code has any adverse effect on the test system, some or all might have to be regressed. This is achieved by re-sending the previous version.

4. If Step 3 is successful, send the modules to production DPCX systems.

The examples assume that a PTF tape has been received for DISOSS/8100[10]. The tape is identified as PTF8303 and it is the second tape that you have received. It contains replacement modules for five FPs and four panels, in assembler object code which you have loaded into the OS partitioned dataset named 'DISOSS.PTF2'. The production systems have been defined to the SYSINFOREF machine list as being in class 1, and the test 8100 as being in a separate class, 900. (See "SYSINFOREF Classes" on page 126 for a discussion of SYSINFOREF classes.)

For convenience of illustration, the sample control statements show the SYSINFOREF maintenance, generation, communication and response phases being executed consecutively within a single jobstream. As described in "SYSINFOREF Phases and Timing of Sessions" on page 128, it may well be desirable to separate the execution of the different phases.

---

[9] In Host Prep release 5.0, PVS can write directly to the BHDSSIL dataset.

[10] This example is fictitious, for the purpose of illustration only. It should not be construed that any such PTF tape as PTF8303 will be produced for DISOSS/8100.

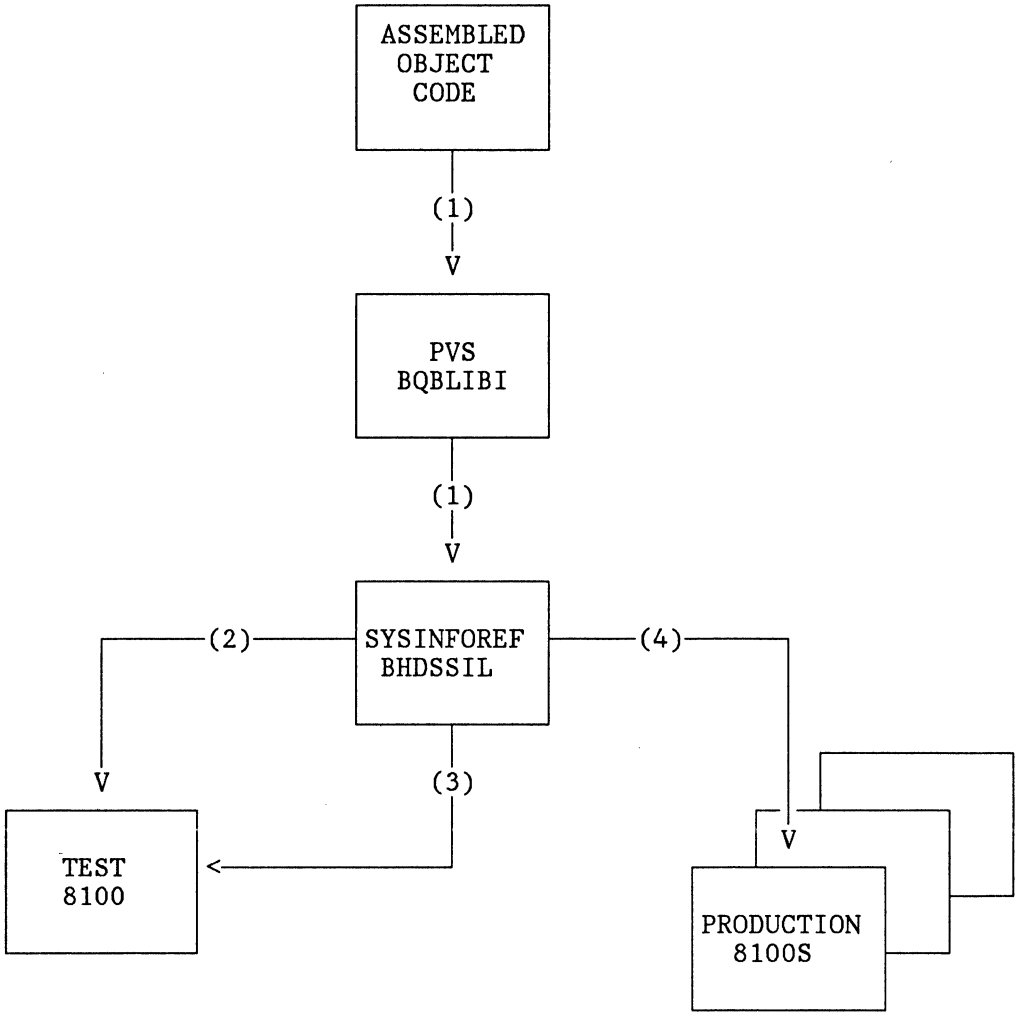
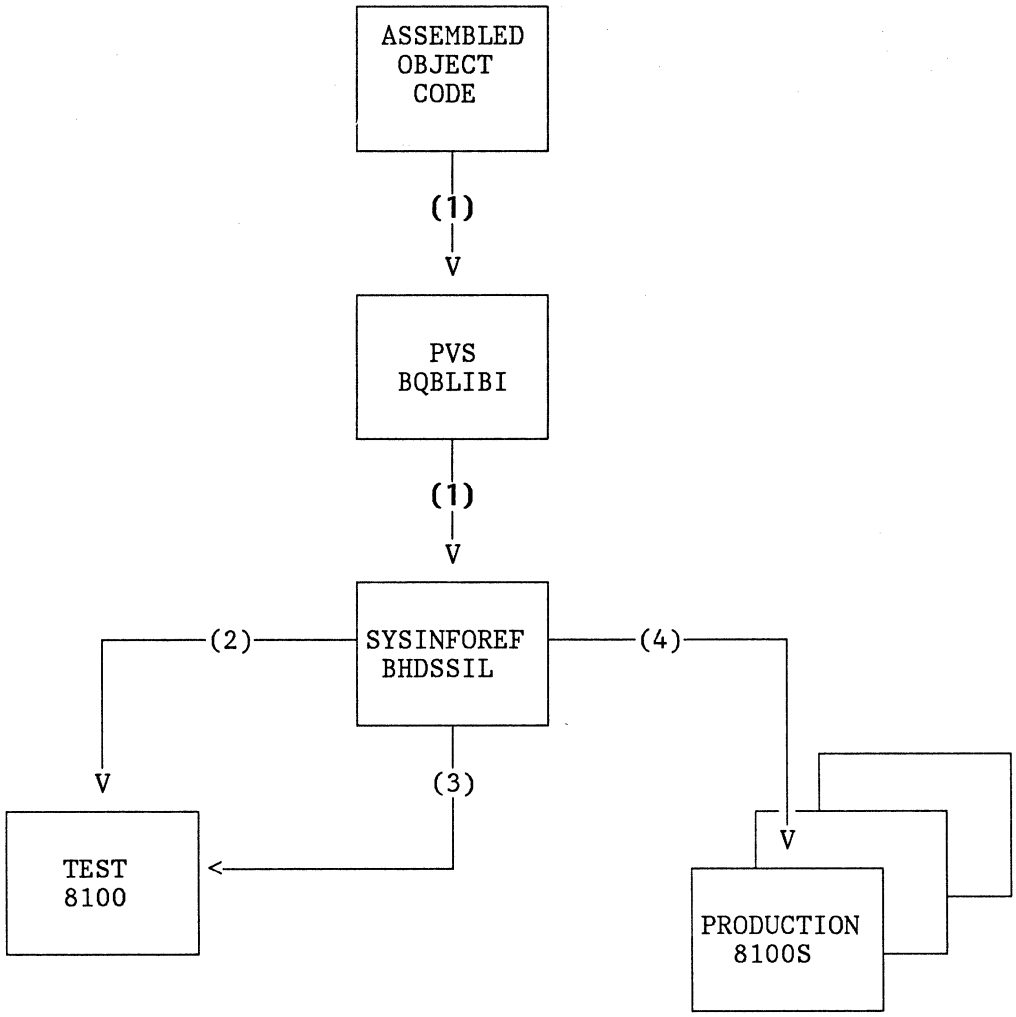


Figure 33. Network installation of service to application software: this is a schematic of the processes involved in installing application code across the network.


ASSEMBLED
OBJECT
CODE
(1)
V
PVS
BQBLIBI
(1)
V
(2)
SYSINFOREF
BHDSSIL
(4)
V
(3)
TEST
8100
V
PRODUCTION
8100S

Step 1: Load the modules on the SYSINFOREF library: The replacement modules must be taken from the library to which they were loaded from the PTF tape, and copied to the SYSINFOREF BHDSSIL dataset. For Host Prep releases prior to 5.0, the PVS job shown in Figure 34 followed by the SYSINFOREF control statements in Figure 35 on page 66 will achieve this, but for release 5.0 or later the PVS job illustrated in Figure 36 on page 67 does this in one step. Note the assignment of version 1 level 2 to this PTF on the PVS READY statements.

```
//PVS      JOB   HOSTPREP
//READY    EXEC  PGM=BQIPVS
//STEPCAT  DD    DISP=SHR,DSN=vsam_user_catalog        (if needed)
//STEPLIB  DD    DISP=SHR,DSN=Host_Prep_load_library  (if needed)
//BQBLIBI  DD    DISP=SHR,DSN=BQBLIBI
//SYSLIN   DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0005)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0008)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0017)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0018)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0053)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7715)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7718)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7745)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7794)
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
.& DEFINE CONTROLS NOPGMR;
.& LOAD PROGRAM INPUT(DISK);
.& READY PROGRAM(7805) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK);
.& READY PROGRAM(7808) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK);
.& READY PROGRAM(7816) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK);
.& READY PROGRAM(7817) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK);
.& READY PROGRAM(7828) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7715) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7718) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7745) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7794) VERSION(1) LEVEL(2);
/*
//
```

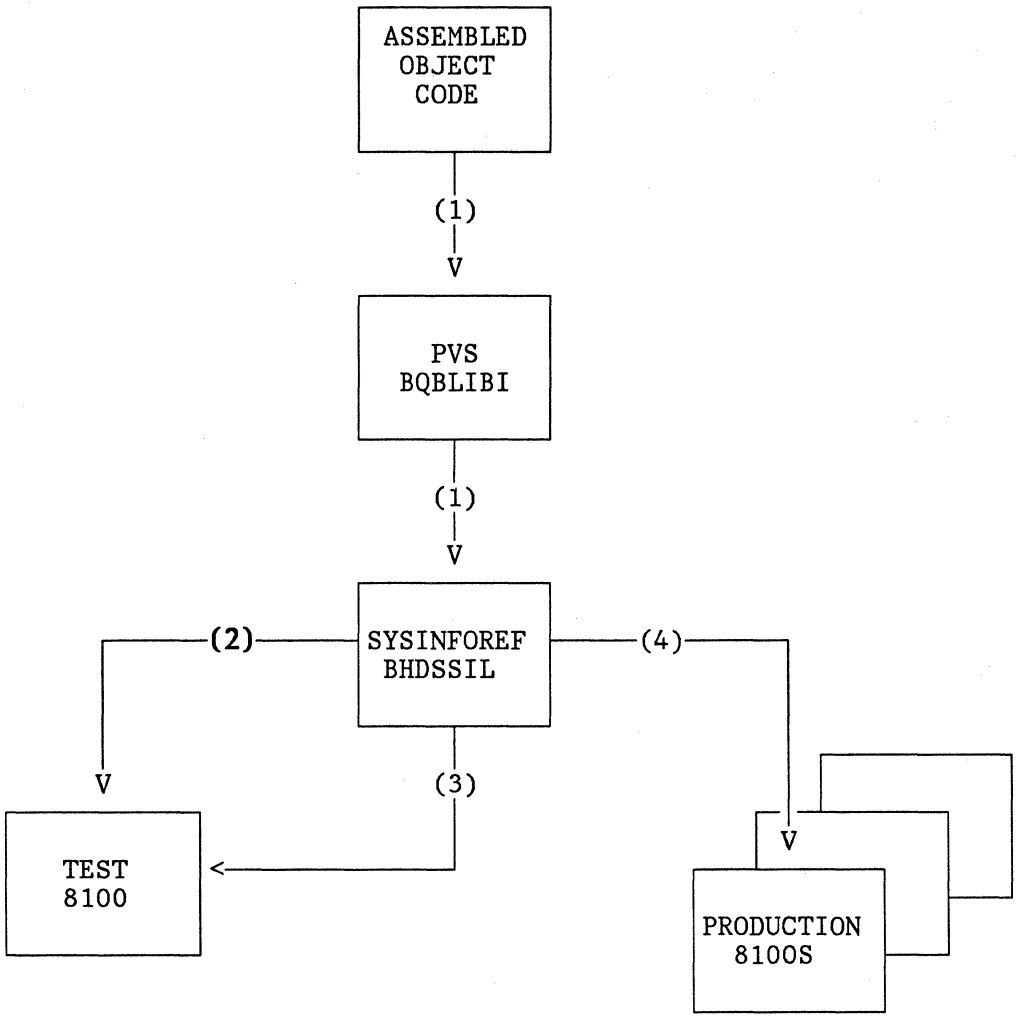
Figure 34. Loading application modules and readying on BQBLIBI: this job uses Host Prep PVS to load application modules and write them to the BQBLIBI dataset, assigning version 1 level 2.

```
.&     COPY          MEMBER                                          ;
FP7805,1,2
FP7808,1,2
FP7816,1,2
FP7817,1,2
FP7828,1,2
PN7715,1,2
PN7718,1,2
PN7745,1,2
PN7794,1,2
```

Figure 35. Copying application modules to BHDSSIL: these control statements cause SYSINFOREF to copy application modules from the BQBLIBI dataset to the BHDSSIL library.

```
//PVS      JOB   HOSTPREP
//READY    EXEC  PGM=BQIPVS
//STEPCAT  DD    DISP=SHR,DSN=vsam_user_catalog        (if needed)
//STEPLIB  DD    DISP=SHR,DSN=Host_Prep_load_library   (if needed)
//BQISSIL  DD    DISP=SHR,DSN=BHDSSIL
//SYSLIN   DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0005)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0008)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0017)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0018)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVF0053)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7715)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7718)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7745)
//         DD    DISP=SHR,DSN=DISOSS.PTF2(DSVP7794)
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    *
.& DEFINE CONTROLS NOPGMR;
.& READY USING BQISSIL;
.& LOAD PROGRAM INPUT(DISK) NOPASS3;
.& READY PROGRAM(7805) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK) NOPASS3;
.& READY PROGRAM(7808) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK) NOPASS3;
.& READY PROGRAM(7816) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK) NOPASS3;
.& READY PROGRAM(7817) VERSION(1) LEVEL(2);
.& LOAD PROGRAM INPUT(DISK) NOPASS3;
.& READY PROGRAM(7828) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7715) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7718) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7745) VERSION(1) LEVEL(2);
.& LOAD PANEL INPUT(DISK);
.& READY PANEL(7794) VERSION(1) LEVEL(2);
/*
//
```

Figure 36. Loading application modules and readying on BHDSSIL: this job uses Host Prep PVS Release 5.0 to load application modules and write them to the BHDSSIL dataset, assigning Version 1 Level 2.

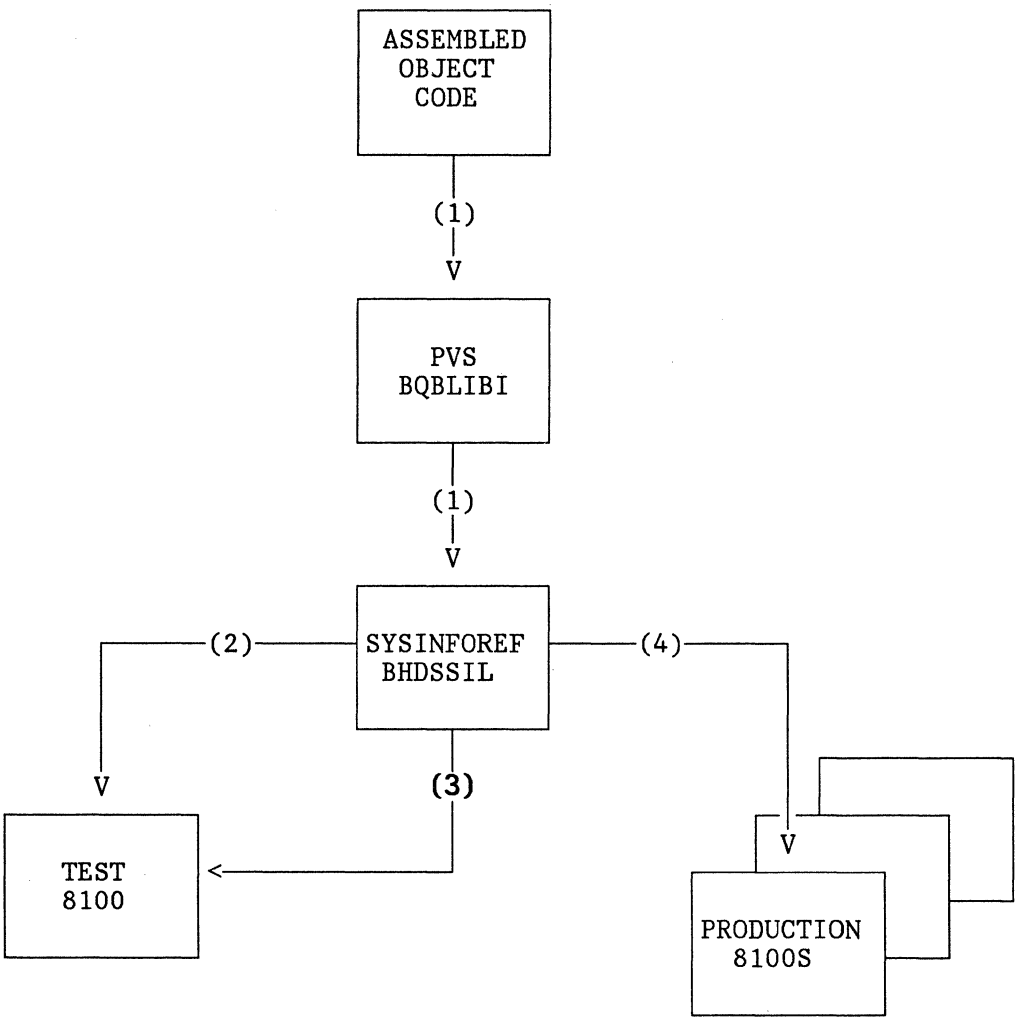

ASSEMBLED
OBJECT
CODE
(1)
V
PVS
BQBLIBI
(1)
V
(2)
SYSINFOREF
BHDSSIL
(4)
V
(3)
TEST
8100
V
PRODUCTION
8100S

Step 2: Send the modules to the test DPCX system: The control statements to do this are illustrated in Figure 37. REPLACE is specified with SEND AMEMBER to cause the previous version to be overwritten at the 8100.

Note that to transmit application modules to an 8100, the system must be quiesced such that only host communication is running with just one host-initiated session enabled, and no users (including the control operator) are signed on.

```
.&     GENERATE     SYSID(TESTZZ10)                          ;
.&     SEND         AMEMBER              REPLACE             ;
FP7805,1,2
FP7808,1,2
FP7816,1,2
FP7817,1,2
FP7828,1,2
PN7715,1,2
PN7718,1,2
PN7745,1,2
PN7794,1,2
.&     END                                                   ;
.&     STARTCOM     SYSID(TESTZZ10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TESTZZ10)     ;
```

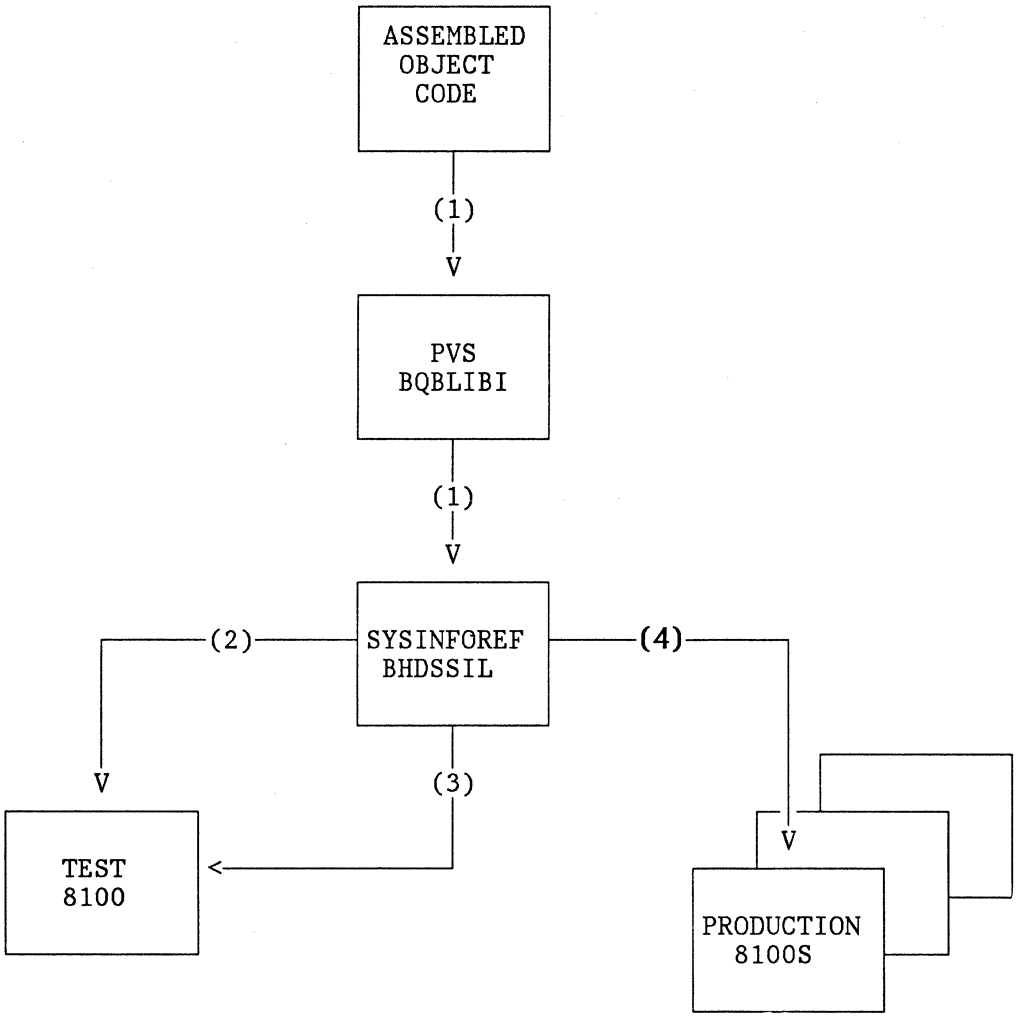
Figure 37. Sending application modules to the test 8100: these control statements cause SYSINFOREF to transmit application modules from the BHDSSIL library to the test 8100.


ASSEMBLED
OBJECT
CODE
(1)
V
PVS
BQBLIBI
(1)
V
(2)
SYSINFOREF
BHDSSIL
(4)
V
(3)
TEST
8100
V
PRODUCTION
8100S

Step 3: Test the new software and perform any regression needed: The test 8100 should be used until you are satisfied that the modules replaced by the PTF have no adverse effect on its operation. Should some or all modules need to be regressed, this is achieved by listing the contents of the BHDSSIL to establish what versions of each module exist - Figure 38 shows how this is done, and then determining, from the application software change log, which version was current at the test 8100. (The application software change log is described in "Recording Software Changes to Application Software" on page 74.) The previous version of each affected module is transmitted using the same control statements as those in Figure 37 on page 69, with the versions and levels altered as appropriate.

```
.&      LIST          MEMBER                ALL                    ;
```

Figure 38. Listing modules on the BHDSSIL library: this control statements causes SYSINFOREF to list all versions and levels of all application entities on its BHDSSIL library.


ASSEMBLED
OBJECT
CODE
(1)
V
PVS
BQBLIBI
(1)
V
(2)
SYSINFOREF
BHDSSIL
(4)
V
(3)
TEST
8100
V
PRODUCTION
8100S

Step 4: Send the modules to the production DPCX systems: Figure 39 contains the control statements for this.

Note that to transmit application modules to an 8100, the system must be quiesced such that only host communication is running with just one host-initiated session enabled, and no users (including the control operator) are signed on.

```
.&     GENERATE     NETWORK                                    ;
.&     SEND         AMEMBER              REPLACE               ;
FP7805,1,2
FP7808,1,2
FP7816,1,2
FP7817,1,2
FP7828,1,2
PN7715,1,2
PN7718,1,2
PN7745,1,2
PN7794,1,2
.&     END                                                     ;
.&     STARTCOM     CLASS(1)                                   ;
.&     LIST         RESPONSE(ALL)                              ;
```

Figure 39. Sending application modules to the production 8100s: these control statements cause SYSINFOREF to transmit the application modules from the BHDSSIL library to production DPCX systems.

4.3.1 PROGRAMS DISTRIBUTED ON DISKETTE

Some IBM products, typically Field Developed Programs, are distributed on BDES format diskettes. To install such products across the network you have two options:

- Take or send the diskettes to each 8100 site in turn and install the product with the DPCX SYSBDISK system service.
- Read the BDES data-stream with a user program and construct a BQBLIBI dataset from it, which can be input to SYSINFOREF. You may distribute the modules across the network as described above.

A program to copy the BDES data-stream is not difficult to write; the data formats are described in full detail in licensed documentation which may be ordered with the Host Prep product. A sample written in PL/1 is to be found in "Program to Copy a BDES Tape back to BQBLIBI" on page 137.

If a diskette reader is available at the host system, the BDES diskette may be read directly by the program; if not, two alternatives exist:

- Convert the diskette to tape with an IBM 3747 or similar device.
- If you have the DPCX Program and Panel Dataset Maintenance Utility field developed program, 5798-DHX, installed on the test 8100, you may install the BDES diskette there, and using the utility unload its con-

stituent panels and datasets to tape. This tape, which is in BDES format, becomes the input to the sample program.

If you adopt the latter method, be aware that the utility does not unload DSCBs, but these may easily be keyed into PVS (DEFINE DSCB and READY DSCB) and generated onto BQBLIBI that way.

4.4 RECORDING SOFTWARE CHANGES TO APPLICATION SOFTWARE

To manage the software in the network, it is necessary to know:

- From what sources the various versions and levels in the SYSINFOREF library came. SYSINFOREF will report when they were loaded.
- Which version and level of which applications are on a given 8100, and when it was installed.

A log should be kept for each of these purposes; sample forms are in appendix "Sample Forms" on page 169. In use, in the context of the example above, such forms might resemble Figure 40 and Figure 41 on page 75.

Product or Module Identifier	Product, PTF or Apar Level	SYSINFOREF Version	SYSINFOREF Level
DISOSS-8100 5668-955	RELEASE 1.0	1	0
DISOSS-8100	PTF 8301	1	1
DISOSS-8100	PTF 8303	1	2
. . . . .			
. . . . .			

Figure 40. Recording SYSINFOREF versions and levels: this is how a log could be kept of the version and level numbers assigned to application entities when they are loaded on the SYSINFOREF library.

System 8100 SYS B at RALEIGH Batch LU TEXTAB10

Product or Module Identifier	SYSINFOREF Version	Level	Date Installed
DISOSS V2	1	0	15DEC82
DISOSS V2	1	1	21FEB83
DISOSS V2	1	2	09MAY83

. . . . . .

. . . . . .

Figure 41. Recording application software installation: this is how a log may be kept of the installation of application software on a DPCX/DOSF system.

PART 2: CONFIGURATION MANAGEMENT

This part of the manual is intended to be read by network administrators and systems programmers responsible for the installation of 8100 DPCX/DOSF systems.

5.0 INTRODUCTION TO CONFIGURATION MANAGEMENT

Configuration management is not an end in itself, but it is a necessary pre-requisite to problem management and to the usability of the DPCX/DOSF systems by the operators. It is treated as a separate topic because the time to think about configuration management is at installation and subsequently when systems are changed, not when systems are already operating in production and problems have occurred.

The essence of configuration management is to have at your finger-tips the necessary information concerning the shape of each DPCX system, both in hardware and software terms, so as to be able to answer queries and resolve problems that arise. This data also becomes of vital importance should catastrophic hardware or software errors or external physical damage ever result in the loss of an entire DPCX/DOSF system. Good configuration management can enable an operational system to be created again more rapidly than is otherwise possible.

This part of the manual is divided into two further chapters:

- "Configuration Information that Should Be Recorded" on page 81 considers the data concerning DPCX systems which needs to be retained, and how the values may be set on the 8100.
- "Planning for Disaster Recovery" on page 93 investigates how a system may be re-created if the need arises, and what precautions should be taken against such an eventuality.

6.0 CONFIGURATION INFORMATION THAT SHOULD BE RECORDED

The purpose of this chapter is to assist in identifying the information that needs to be recorded, and to this end sample forms are referenced on which data can be manually recorded. There are various software products which will act as filing systems for this information, but it has to be manually supplied to all of them. One of these products, Information/Management, is described briefly in "Overview of the Information/Management Program Product" on page 90 at the end of this chapter.

Where the data is used to configure DPCX by the use of commands, this information can be entered in the form of a DPCX PROC at the host, and sent to DPCX. The host library where the PROC has been placed may then be listed to obtain current configuration documentation.

Suggestions for recording configuration information are included for the following:

- System definitions (system parameters established with the DPCX DEFINE command) in "Establishing System Values" on page 83.
- Operator definitions in "Establishing Operator Profiles" on page 85.
- Host tables (SSCP, batch and LU/PGMID) in "Establishing SYSHOST Values" on page 86.
- SYSCONFG parameters in "Establishing SYSCONFG Values" on page 86.
- SYSIMOD parameters in "Establishing SYSIMOD Values" on page 87.
- RJE signon parameters and FCBs in "Establishing SYSRJE Values" on page 88.
- DOSF system values in "Establishing DOSF Values" on page 88.
- DOSF queues in "Establishing DOSF Queue Definitions" on page 88.
- DTF sessions in "Establishing DTF Session Definitions" on page 88.
- Loop maps in "Establishing the Layout of Loops" on page 89.
- Network terminal data in "Establishing Terminal Addresses and Setup Options" on page 89.
- Network resource names in "Establishing Network Resource Information" on page 90.

6.1 DISTRIBUTING PROCS FROM A HOST TO DPCX SYSTEMS

The technique recommended for managing PROCs is to maintain them in a partitioned data set on an OS/VS host or a source statement library on a DOS/VSE host, and to transmit them to DPCX using RJE. The examples that follow all assume this method of operation.

If your installation does not wish to use RJE, the following alternatives exist:

- If DSX is employed for network management, you may prefer to use it to handle PROCs. They may be created on the DSX library as CLISTs, and distributed with DSX. (This requires the SPE for DSX Version 1 contained in PTF UP90061.) An example of this technique may be found in Figure 2 on page 12 under "DPCX/DOSF Initial Installation" on page 8.

- PROCs may be created as documents on the test 8100, archived to the host with DISOSS and retrieved to the production systems where they can be made into PROCs with the CPROC command. This requires intervention on each production 8100, though HCF could be employed to perform the retrieval from a host attached terminal.

- If the installation employs DIF, the PROCs could be created as documents in the host DLF library and retrieved to the production systems with DIF where they can be made into PROCs with the CPROC command. This requires intervention on each production 8100, though HCF could be employed to perform the retrieval from a host attached terminal.

- Procs may be created as documents on the test 8100, archived to diskette and distributed manually to the production systems where they can be made into PROCs with the CPROC command. This requires intervention on each production 8100.

Note: Before it will be possible to distribute PROCs from the host, SYSCONFG, SYSIMOD and RJE will need to be correctly configured, and the DEFINE HOST,ID= command will require to be issued to identify the host SSCP to DPCX. More information on sending PROCs to DPCX systems is to be found under "RJE System Programmer Features" in the DPCX RJE Installation and Operation manual.

PROCs may be developed and tested at the test 8100 and sent to the host with RJE, so that they are known to work before being distributed throughout the network. (If DSX is used, they may be retrieved from the test 8100 to the host.)

There is no need to limit the host library of PROCs to only those that relate to defining the configuration; logon, operational and other PROCs could be maintained, with either one, network-wide version or a tailored copy for each DPCX system. The DPCX DOSF Command Procedures manual documents the control statements used in creating PROCs, and the DPCX DOSF Command Reference describes the commands which may be used in them.

```
//SENDPROC JOB   GENER
//REPRO    EXEC  PGM=IEBGENER
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    DUMMY
//SYSUT1   DD    DSN=SYS2.DPCX.PROCS(SYSTEMAB),DISP=SHR
//SYSUT2   DD    SYSOUT=B,DEST=R91
//
```

Figure 42. Sending a PROC to a DPCX system from OS/VS: This job will send the 'SYSTEM' PROC for system 'AB', to the DPCX system which is JES remote 91, from OS/VS.

```
* $$JOB JNM=SSERV,CLASS=0
// JOB SSERV
// DLBL IJSYSSL,'DPCX.PROCS'
// EXTENT SYSSLB,SYSWK4
// ASSGN SYSSLB,3330,VOL=SYSWK4,SHR
// ASSGN SYSPCH,X'00D'
// EXEC SSERV
  PUNCH  X.SYSTEMAB
/*
/&
* $$ EOJ

* $$JOB JNM=SENDPROC,CLASS=0
* $$ PUN REMOTE=5
// JOB SENDPROC
// DLBL IJSYSCL,'PROD.CORE.IMAGE.LIBRARY.C'
// EXTENT SYSCLB,DOSRES
ASSGN SYSCLB,SYSRES
// ASSGN SYSPCH,X'00D'
// UPSI 1
// EXEC DITTO
$$DITTO CC
.....................................................
.                                                   .
. Remove CATALS and BKEND cards and insert PROC here. .
.                                                   .
.....................................................
/*
$$DITTO EOJ
/*
ASSGN SYSCLB,UA
/&
* $$ EOJ
```

Figure 43. Sending a PROC to a DPCX system from DOS/VSE: These jobs will send the 'SYSTEM' PROC for system 'AB', to the DPCX system which is POWER remote 5, from DOS/VSE. The example utilises the VSE/DITTO program product.

The examples show how to send a PROC from an OS/VS host system to a DPCX system (Figure 42 on page 82) and from a DOS/VSE host system to a DPCX system (Figure 43). Note that it is easy to send one PROC to each DPCX system in the same job, but host job entry subsystems may concatenate output to the same remote station from different job steps so that DPCX would receive just one PROC. In consequence, transmit no more than one PROC to a given DPCX system in any one job.

A technique which could be used to execute a PROC on each 8100 in the network is described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

6.2 ESTABLISHING SYSTEM VALUES

The jobs illustrated in Figure 44 on page 84 (OS/VS) and Figure 45 on page 84 (DOS/VSE) show the possible content of a PROC to establish system val-

ues and how it may be filed in a host library. Rather than have individual 'SYSTEM' PROCs for each 8100, your installation may determine that these values will be identical at each 8100 and one network-wide PROC could then be maintained and filed in the host library under the name of 'SYSTEM'.

```
//LOADPDS  JOB   GENER
//REPRO    EXEC  PGM=IEBGENER
//SYSPRINT DD    SYSOUT=A
//SYSIN    DD    DUMMY
//SYSUT2   DD    DSN=SYS2.DPCX.PROCS(SYSTEMAB),DISP=OLD
//SYSUT1   DD    *
:REPLACE  SYSTEM              10/18/82
*PROC
*CO    ***** THIS IS THE 'SYSTEM' PROC FOR SYSTEM AB *****
DEFINE CMDC=*
DEFINE JOURNAL=YES
DEFINE PROMPT=?
DEFINE STARTUP=IPL
DEFINE SEARCH=(PROC,CMD,PGM),MODE=BASE
DEFINE SEARCH=(PROC,CMD,PGM),MODE=TEXT
*CO  DEFINE COMMAND ACCESS CODES  COULD GO HERE
*CO  DEFINE SYSTEM SERVICE ACCESS CODES  COULD GO HERE
*EXIT
/*
//
```

Figure 44. Filing a 'SYSTEM' PROC at an OS/VS Host: This job will place the 'SYSTEM' PROC for DPCX system 'AB' into an OS partitioned data set.

```
* $$ JOB JNM=LOADLIB,CLASS=0
// JOB    LOADLIB
// DLBL   IJSYSSL,'DPCX.PROCS'
// EXTENT SYSSLB,SYSWK4
// ASSGN  SYSSLB,3330,VOL=SYSWK4,SHR
// EXEC   MAINT
  CATALS   X.SYSTEMAB
  BKEND
:REPLACE  SYSTEM              10/18/82
*PROC
*CO    ***** THIS IS THE 'SYSTEM' PROC FOR SYSTEM AB *****
DEFINE CMDC=*
DEFINE JOURNAL=YES
DEFINE PROMPT=?
DEFINE STARTUP=IPL
DEFINE SEARCH=(PROC,CMD,PGM),MODE=BASE
DEFINE SEARCH=(PROC,CMD,PGM),MODE=TEXT
*CO  DEFINE COMMAND ACCESS CODES  COULD GO HERE
*CO  DEFINE SYSTEM SERVICE ACCESS CODES  COULD GO HERE
*EXIT
  BKEND
/*
/&
* $$ EOJ
```

Figure 45. Filing a 'SYSTEM' PROC at a DOS/VSE Host: This job will place the 'SYSTEM' PROC for DPCX system 'AB' into a DOS source statement library.

6.3 ESTABLISHING OPERATOR PROFILES

The PROC illustrated in Figure 46 is an example of how operator values might be established at a DPCX system. The JCL used to file it in a host library is the same as in Figure 44 on page 84 (OS), changing the PDS member name on the SYSUT2 DD statement from 'SYSTEMAB' to 'USERSAB', or in Figure 45 on page 84 (DOS), changing the CATALS card similarly.

DOSF operator characteristics are assigned with the SYSXPRFL System Service, or for release 3 of DOSF with the PROFILE command. The command is an interface to full-screen menus, and not all of the different options can be entered in a PROC. In either case the values assigned should be recorded on a form such as the "Operator Profile Worksheet" in Appendix A of the DOSF Planning manual, and are set by the control operator or other authorised operator, possibly from the host using HCF.

To what extent operator definitions are established from the host as opposed to being generated by the DPCX control operator will depend on the way your installation is organised. It is probable that some definitions, such as those for the control operator, the RJE operator and the DOSF pattern operator will be established from the host. If DISOSS is employed, these operator definitions will probably be determined at the host, as they must be defined to the DISOSS host product.

```
:REPLACE  USERS                10/18/82
*PROC
*CO    ***** THIS IS THE 'USERS'  PROC FOR SYSTEM AB *****
DEFINE OPID=1,ACCESS=11111111,LOGPROC=LOGONCTL,MODE=COMMAND
DEFINE OPID=1,PAUTH=YES,PSWD=PWCTLOP,SYSACC=11111111
DEFINE OPID=1,JO=YES,DESC='CONTROL OP.'
DEFINE OPID=2,ACCESS=11111111,LOGPROC=LOGONCTL,MODE=COMMAND
DEFINE OPID=2,PAUTH=YES,PSWD=BACKUP,SYSACC=11111111
DEFINE OPID=2,JO=NO,DESC='BACKUP OP.'
DEFINE OPID=3,ACCESS=10000111,LOGPROC=LOGON,MODE=COMMAND
DEFINE OPID=3,PAUTH=NO,PSWD=PW3XYZ,SYSACC=00010001
DEFINE OPID=3,JO=NO,DESC='FANNY SLUMBER'
DEFINE OPID=4,ACCESS=00000111,LOGPROC=LOGON,MODE=COMMAND
DEFINE OPID=4,PAUTH=NO,PSWD=PW4ABC,SYSACC=00000001
DEFINE OPID=4,JO=NO,DESC='MAY CHATTERBOX'
DEFINE OPID=5,ACCESS=00000111,LOGPROC=LOGON,MODE=COMMAND
DEFINE OPID=5,PAUTH=NO,PSWD=PW5PQR,SYSACC=00000001
DEFINE OPID=5,JO=NO,DESC='SUSIE DOLITTLE'
TEXT
SPELPRFL 1,,,YES,YES
SPELPRFL 2,,,YES,YES
SPELPRFL 3,,,YES,YES
SPELPRFL 4,,,YES,NO
SPELPRFL 5,,,YES,NO
BASE
*EXIT
```

Figure 46. A 'USERS' PROC to define operators: This PROC will define the operators for DPCX system 'AB'.

6.4 ESTABLISHING SYSHOST VALUES

The 'DEFHOST' PROC illustrated in Figure 47 is an example of how SYSHOST values might be established at a DPCX system. The JCL used to file it in a host library is the same as in Figure 44 on page 84 (OS), changing the PDS member name on the SYSUT2 DD statement from SYSTEMAB to HOSTAB, or in Figure 45 on page 84 (DOS), changing the CATALS card similarly.

It may well be the case that the configuration of host options is the same for all 8100s, and in this case only a single, network-wide PROC need be maintained, and filed in the host library under the name of 'HOST'.

```
:REPLACE  DEFHOST              10/18/82
*PROC
*CO    ***** THIS IS THE 'DEFHOST' PROC FOR SYSTEM AB *****
DEFINE HOST,BATCHID=SIRF
DEFINE HOST,BATCHID=DSX
DEFINE HOST,ID=11
DEFINE HOST,LUPGMID=(10,958)
DEFINE HOST,LUPGMID=(11,958)
DEFINE HOST,LUPGMID=(12,2570)
DEFINE HOST,LUPGMID=(13,2570)
DEFINE HOST,LUPGMID=(14,2570)
DEFINE HOST,LUPGMID=(45,932)
DEFINE HOST,LUPGMID=(49,977)
DEFINE HOST,LUPGMID=(50,932)
DEFINE HOST,LUPGMID=(52,932)
DEFINE HOST,LUPGMID=(55,932)
*EXIT
```

Figure 47. A 'DEFHOST' PROC to define SYSHOST values: this PROC will define the SYSHOST values for system 'AB'

6.5 ESTABLISHING SYSCONFG VALUES

SYSCONFG values may be displayed, changed or printed from an HCF-connected terminal. Note that you must sign on with either user number 1 (control operator) or 255 (service representative). Also, if you attempt to process the configuration (SYSCONFG option 3), you will need to be the only signed-on user with only one host-initiated session enabled and no other tasks started. Moreover, as this SYSCONFG option concludes with a re-IPL of DPCX you will only be able to regain your DPCX session without intervention at the 8100 if the DPCX system has an appropriate startup PROC specified which enables the host link. Processing incorrect SYSCONFG values may well necessitate intervention at the 8100 site to recover.

For recording SYSCONFG values, there are good work-sheets appended to the DPCX DOSF Installation manual; alternatively, the values may be printed with SYSCONFG and the print records retrieved and printed at the host with SYSINFOREF and the listing retained. See "Capturing DPCX Displayed or Printed Data at the Host" on page 110 for details of the procedure.

6.6 ESTABLISHING SYSIMOD VALUES

SYSIMOD values may be changed from an HCF-connected terminal. Note that you must sign on with either user number 1 (control operator) or 255 (service representative), and that changes made with SYSIMOD are only effective after the next DPCX IPL. Specifying incorrect SYSIMOD values for option 1 (the host link) may well require intervention at the 8100 site to recover as it may not subsequently be possible to access the 8100 with HCF.

For recording the values, work-sheets are recommended as follows according to SYSIMOD option group:

- Option 1 - Host Link: use a form such as that in Figure 72 on page 173.
- Option 2 - Print Group Table: use a form such as that in Figure 73 on page 174.
- Option 3 - Transaction Data Set Group Table: if you have coded your own applications and need to define this table, use a form such as that in Figure 74 on page 175 to record the information.
- Option 4 - System IOCBs: you should not normally require this option - the DPCX system will determine what devices are directly attached.
- Option 6 - LU/LA Table: use forms such as those in Figure 75 on page 176 and Figure 76 on page 177.
- Option 7 - RJE Parameters: use a form such as that in Figure 77 on page 178.
- Option 8 - System Options: use a form such as that in Figure 78 on page 179.
- Option A - Data Link Adaptors: use forms such as those appended to the DPCX DOSF Installation manual.
- Option B - Directly Attached Loops: use forms such as those appended to the DPCX DOSF Installation manual.
- Option C - Printer Matrix: use a form such as that in Figure 79 on page 180.

With DPCX release 3.0 and its full-screen SYSIMOD function, the alternative exists of printing the values with SYSIMOD, retrieving the print records and printing them at the host with SYSINFOREF and retaining the listing. See "Capturing DPCX Displayed or Printed Data at the Host" on page 110 for details of the procedure.

6.7 ESTABLISHING SYSRJE VALUES

There is some data which needs to be defined before RJE may be used. It is set with the SYSRJE System Service, in field by field mode of operation rather than command mode. These values may be set from the host by signing on as the RJE operator via HCF.

There are two things to be established:

- RJE signon parameters. These are set by SYSRJE option 7.1, and up to 5 sets may be supplied. More than one set will be wanted if you want to communicate with different host systems, or if you want to have different logmode tables available; for instance, to select from various sets of pacing parameters, or to have an option to use SNA compaction when desired. A sample form which may be used to record this data is shown in Figure 80 on page 181.
- RJE FCBs. Any FCB which an RJE data-stream may request needs to be defined to DPCX. This is done with SYSRJE option 7.2; a sample form for recording the data is included in Figure 81 on page 182. It is

assumed that the definition of an FCB will be the same at each 8100 which may utilise it - that is, these definitions are network-wide.

6.8 ESTABLISHING DOSF VALUES

DOSF values may be set (with SYSXGEN) or changed (with SYSXMNT) from an HCF-connected terminal. Note that you must sign on with an authorised user number.

For recording SYSXGEN/SYSXMNT values, you may use a worksheet such as that in Figure 82 on page 183, or, once set, the values may be printed by any user with SYSXFEA3 option 5, the print records retrieved and printed at the host with SYSINFOREF and the listing retained. See "Capturing DPCX Displayed or Printed Data at the Host" on page 110 for details of the procedure.

6.9 ESTABLISHING DOSF QUEUE DEFINITIONS

The definitions for standard DOSF queues to be used at a DPCX system should be determined and recorded on a form such as the Print Queue Worksheet in Appendix A of the DOSF Planning manual.

The worksheets may be sent to the control operator who can then define the queues, or they may be defined by host personnel with SYSXCTRL from an HCF-connected terminal using an authorised user number.

6.10 ESTABLISHING DTF SESSION DEFINITIONS

If your installation uses DISOSS or DIF, you will need to define some document transfer facility (DTF) sessions using the SYSXSDEF System Service. Record the values to be used on forms such as those in Figure 83 on page 184 and Figure 84 on page 185.

You may obtain a listing at the host of what is actually defined by:

- Invoking SYSXSDEF
- For each defined session, taking the option to Document it

Either:

- Store the resultant documents at the host with DISOSS
- Print the documents at the host using DISOSS Host Print

Or:

- Send the resultant documents to a host DLF library with DIF
- Print the documents at the host using DCF

Or:

- Using the CPROC command turn each document into a PROC
- Add appropriate host JCL (to print the input stream at the host) to the PROCs with SYSEDIT
- Submit these jobs to the host using RJE

Those of the above processes which are done on the 8100 may of course be performed from the host using HCF.

6.11 ESTABLISHING THE LAYOUT OF LOOPS

In order to be able to resolve errors that occur in the physical components of loops it is essential that you know the sequence in which wrap loop station connectors and loop wiring concentrators occur, and which radial loop station connector is connected to which spur of the concentrator. It is imperative that each loop station be assigned a number with which it should also be labelled, and that the identity of the adjacent stations be known.

In order that this information be available at the host, some form of loop map should be maintained and a copy be kept at the host. The loop layout chart referred to in the IBM Multiuse Communication Loop Planning and Installation Guide is an example. This manual contains very detailed information on procedures to follow when planning and installing the loop, and it should be consulted.

6.12 ESTABLISHING TERMINAL ADDRESSES AND SETUP OPTIONS

It is important to maintain a record of what devices are configured on a DPCX/DOSF system, and how they should be set up. DPCX has a concept of logical addresses (LAs) by which it accesses all its devices. There is a pre-determined addressing architecture: LAs exist for so many of each device, whether or not they are present, and the logical addresses determine the physical addresses of devices.

Appendix A of the DPCX System Services manual contains a chart setting out all the logical addresses in the system. Use this, in conjunction with forms such as that in Figure 85 on page 186, to record addresses, description, identity and location of each device. Use one (or more) forms for each 8100 system, and create entries just for the subset of possible devices which is actually present on the system.

A record of setup options for those devices designated as customer setup should also be kept at the host. The data to be recorded will vary with the device in question, but there may be an appropriate form shipped with it on which you may record the options.

6.13 ESTABLISHING NETWORK RESOURCE INFORMATION

You will need to keep a record of the network resource names by which VTAM knows the DPCX/DOSF systems. The relevant portion of the data already

recorded by network administrators or system programmers when defining the resources to VTAM, which should at the very least be NCP source statements with comments imbedded, should be adequate. The information should correlate the VTAM PU and LU names with the identification of the DPCX system and the NCP local addresses.

Against this information it is convenient to record the DOSF system identification code if DISOSS is used and the work station remote number if RJE is used. In addition, the naming convention used for defining logical units for DPCX systems should be documented with a record of the purpose that each is put to. It is suggested that you have a convention whereby the first part of the LU name identifies the DPCX system to which it belongs, while a suffix corresponds to a given NCP local address and DPCX usage. The DPCX usage should be consistent across 8100s; if an application is not used at some DPCX system and so requires no LUs, those LUs can be omitted from the NCP generation for that 8100, but should not be used for some other application.

If the NLDM program product, 5668-971, is available it provides the capability to display readily network resource names and associated information, and can reduce the amount of information which needs to be written down. See the NLDM General Information manual for more information.

6.14 OVERVIEW OF THE INFORMATION/MANAGEMENT PROGRAM PRODUCT

Information/Management is a feature of the Information/System program product, 5735-OZS. This product offers management facilities in three areas:

- Configuration Management
- Change Management
- Problem Management

Within these categories, Information Management offers functions as follows:

1. Configuration Management: maintaining inventory records for:
    - Data Centres (network computing locations)
    - Systems at a centre
    - Hardware and software components comprising a system
    - Features of a component
    - Connections between components
    - Financial data
    - Service organisations
2. Change Management:
    - Change requesting

- Assessing impact of changes
- Change scheduling
- Change approval
- Change coordination
- Change implementation
- Post-change review

3. Problem Management:
   - Problem reporting
   - Problem assigning
   - Problem coordination
   - Problem tracking
   - Problem analysis
   - Problem resolution

To support these functions, Information/Management has a database with these capabilities:

- Record creation, update, copying and deletion.
- Record search and retrieval, using a flexible free-form search argument.
- Report generation: printing online search results lists and standard or customised reports.

For more information on this product, read the Information/System General and Preinstallation Information manual.

7.0 PLANNING FOR DISASTER RECOVERY

What would you do if a fire destroyed one of your DPCX/DOSF systems?

Disaster recovery is the process of regaining an operational system after a catastrophic hardware or software error, or external physical damage, has rendered it unusable. It is assumed that such an eventuality will require the system to be re-created, either by re-generation or restoring from some form of backup, or some combination of both techniques.

In planning for disaster recovery, it is important to strike a balance between on the one hand consuming a large amount of time preparing for a contingency which may never arise, and on the other being unable to recover vital information if the unthinkable does happen. The minimum that should be done is to classify all that goes to make up a DPCX/DOSF system into that which can be re-created if a system is lost and that which can not, and should hence be backed up in some way. Procedures may then be put in place to ensure that data which could not be re-created is periodically saved.

This chapter has two sections. They describe:

- What could be re-created, and how to re-generate it.
- What could not be re-created, and how to save it.

The lists are not exhaustive; the purpose of including them is to focus awareness on the need for disaster recovery planning. They should prompt you to give a little thought to the procedures required to ensure that recovery of a DPCX/DOSF system after a catastrophic error is at least possible.

7.1 DATA WHICH CAN BE RE-CREATED

This section lists some of the data on the system which it will be possible to re-create, and the way in which this could be done.

- System software. This would be re-installed by an initial install of the current releases as in "Base Code Initial and Update Installation" on page 7. PTFs would be sent again as mentioned in "Installing PTFs after an Initial or Update Install" on page 55.
- Application software. This, being maintained on the SYSINFOREF BHDSSIL dataset while in use at DPCX systems, could be sent again.
- PROCs - originated at the host. Those PROCs developed centrally at the host should be retained there while in use in the network, and these would be sent again (see "Distributing PROCs from a Host to DPCX Systems" on page 81).
- Configuration Information. Some of this may be available in PROCs developed at the host. The rest should be available for reconstruction of a DPCX/DOSF system if it was recorded as suggested in "Configuration Information that Should Be Recorded" on page 81.

- Operator definitions - originated at the host. If the suggestions of "Establishing Operator Profiles" on page 85 are followed, these would be available in part in PROCs, with a manual record of DOSF parameters.

- DOSF queue and DTF session definitions - originated at the host. If these are documented as suggested in "Configuration Information that Should Be Recorded" on page 81 the information to re-create them would be available.

- User Datasets. If the application is one in which the 8100 holds a copy of some host data, it should be possible to rebuild the datasets from the host.

7.2 DATA WHICH CANNOT BE RE-CREATED

This section lists some of the data on the system which it will not be possible to re-create, unless precautions are taken to save it periodically, with the tools indicated.

- PROCs - originated at DPCX. Locally written PROCs should be saved on diskette if the control operator wants to be sure that they will always be available in the future.

- Operator definitions - originated at DPCX. If the control operator establishes operator profiles, they should be listed periodically with the DOSF SYSXPRFL system service and the information retained in case you ever need to re-create them.

- DOSF queues and DTF sessions - originated at DPCX. If the control operator defines queues or sessions the definitions should be saved on to diskette with DOSF system services SYSXCTRL and SYSXSDEF respectively.

- DOSF documents: any document which you are not prepared to risk having to re-key should be archived, either to diskette or to the host.

- User datasets: any user datasets containing data which cannot otherwise be re-created may be saved on diskettes. Relative and indexed datasets can be dumped using the DPCX SYSCOPY system service. RSDS datasets can be dumped using the 8100/DPCX RSDS File Re-organisation and Backup Utility field developed program (number 5798-DJJ) or similar. If the DPCX system ever has to be rebuilt, the datasets could be restored from the diskettes.

  Note that the DOSF supplemental spelling dictionary and data dictionary are user datasets of which you may wish to save copies.

This section of the manual should be read by host Help Desk personnel responsible for assisting DPCX users, and by network administrators and system programmers responsible for problem determination throughout the network.

8.0 INTRODUCTION TO PROBLEM MANAGEMENT

Problem Management is the process of resolving problems in the network; it can be described as a system containing the following elements:

- Problem Recognition
- Problem Reporting and Logging
- Problem Determination
- Problem Bypass and Recovery
- Problem Resolution
- Management Review

Problem Recognition occurs when the existence of a problem is identified through a symptom or trend, or a deviation from what is usual. The discussion in this part of the manual presumes the recognition of problems will have already occurred.

Problem Reporting and Logging is the process of recording information about the problem for coordination, subsequent problem determination and eventual resolution. A basic assumption is that some sort of user help desk exists in your organisation, to which DPCX control operators, having recognised a problem, will report it. Help desk personnel, working with network administrators and system programmers, are responsible for problem determination.

Problem Determination involves the collection and analysis of data to isolate and identify the failing component. It is the major topic considered in this chapter, since much is specific to the DPCX/DOSF environment.

Problem Bypass and Recovery entails restoring the affected function through bypass and recovery procedures.

Problem Resolution means the correcting of the problem by repairing the failing component. For problems that are determined to be caused by software malfunctions, this will usually involve the application of a PTF, and the processes involved are discussed in "PART 1: SOFTWARE MANAGEMENT" on page 3.

Management Review is the process of ensuring that the problem management system functions effectively. It is not discussed here.

For more information on a structured approach to problem management, with particular reference to the help desk and problem recording and logging, ask your IBM representative for an appropriate installation management guide. Half the effort involved in problem determination may be eliminated by properly recording a description of the problem and subsequently tracking progress on the investigation of the incident. The Information/Management feature of the Information/System program product, 5735-OZS, described in "Overview of the Information/Management Program Product" on page 90, contains functions to assist in problem management.

A diversity of tools for problem determination is considered in this chapter. Figure 48 on page 98 and Figure 49 on page 99 attempt to indicate to what type of problems each one is applicable. All these tools are dis-

	DASD	Tape	Dskt	Host	Loop	Link	Term	Perf	DPCX	DOSF	Apps
HCF	X	X	X	X	X	X	X	X	X	X	X
.& RETRIEVE SYSDUMP									X	X	
.& RETRIEVE DUMPS											X
.& RETRIEVE CIL	X	X	X	X	X	X	X		X	X	X
.& SEND / RETRIEVE DLA...					X	X	X				
.& RETRIEVE FIDLOG							X				

Dskt= diskette, Term= terminal, Perf= performance, Apps= Application

Figure 48. Matrix of problem types against applicable host tools: the diagram indicates which tools available at the host may be used in determining various types of problems.

cussed in more detail in other publications; Figure 50 on page 100 contains a list indicating to which manual to refer. This part of the manual focuses in two further chapters on two areas of problem determination of specific relevance to a network of DPCX/DOSF systems:

- "Aids for Determination of Network Problems." on page 101. General purpose communication network management tools are described briefly; DPCX techniques in more detail.
- "Performing DPCX Problem Determination from the Host" on page 109. Specific techniques for this purpose are discussed here.

	DASD	Tape	Dskt	Host	Loop	Link	Term	Perf	DPCX	DOSF	Apps
Journal	X	X	X	X	X	X	X		X	X	
Messages	X	X	X	X	X	X	X		X	X	X
*DISPLAY				X							
*QUERY		X	X	X	X	X	X				
SYSCONFG	X	X	X	X	X	X	X				
SYSDC	X							X	X		
SYSDEBUG											X
SYSDVAR	X										
SYSIMOD				X	X	X	X				
SYSLDSA	X										
SYSLTSD		X									
SYSRIS	X		X						X	X	
SYSSPREC	X										
SYSTCM		X			X	X	X				
SYSTEST											X
SYSTRACE		X		X	X	X	X		X		
SYSXFEA3	X									X	

Dskt= diskette, Term= terminal, Perf= performance, Apps= Application

Figure 49. Matrix of problem types against applicable DPCX tools: the diagram indicates which tools available at the 8100 may be used in determining various types of problems.

HCF	HCF Guide and Reference
SYSINFOREF	Host Prep Sysinforef Guide and Reference
Journal	DPCX DOSF Command Reference
Messages	DPCX DOSF Messages and Codes
***DISPLAY**	DPCX DOSF Command Reference
***QUERY**	DPCX DOSF Command Reference
SYSCONFG	DPCX DOSF Installation
SYSDC	DPCX System Services
SYSDEBUG	DPCX Terminal Operations Program Execution Monitor Guide
SYSDVAR	DPCX DOSF Diagnosis Guide
SYSIMOD	DPCX DOSF Installation
SYSLDSA	DPCX System Services
SYSLTSD	DPCX System Services
SYSRIS	DPCX DOSF Diagnosis Guide
SYSSPREC	DPCX System Services
SYSTCM	DPCX System Services / DPCX DOSF Diagnosis Guide
SYSTEST	DPCX Terminal Operations Program Execution Monitor Guide
SYSTRACE	DPCX DOSF Diagnosis Guide
SYSXFEA3	DOSF System Services

Figure 50. Reference of manuals for problem determination tools: the list gives the manual which documents the named problem determination tool.

9.0 AIDS FOR DETERMINATION OF NETWORK PROBLEMS.

This chapter contains brief descriptions of two general purpose communication network management products:

- Network Communications Control Facility
- Network Problem Determination Application

Although it is desirable to be able to resolve all network problems from the host site, there will be times when information is needed from DPCX. In consequence, this chapter also contains information on specific DPCX tools which are of use in diagnosing network problems.

9.1 OVERVIEW OF THE NETWORK COMMUNICATIONS CONTROL FACILITY.

The Network Communications Control Facility (NCCF), 5735-XX6, is a program product designed to let you control, record and automate various operator tasks entailed in running your network. NCCF can be used as an operator interface to VTAM. It provides optional logging to disk or printers of operator interactions. NCCF operates as a VTAM application program in OS/VS1, OS/VS2 MVS or DOS/VSE, using 3270 devices as operator terminals and hardcopy printers.

NCCF is a program base for communication network management. In addition to providing facilities for control of network operation and automation thereof, programs may be run in conjunction with NCCF and using its services, to address the following areas:

- Network operations management
- Problem determination
- Configuration management
- Change management
- Problem management

These programs may be IBM program products or user written.

A brief summary of some NCCF facilities is set out below. More information may be found in the NCCF General Information manual.

9.1.1 MAJOR NCCF FACILITIES

This is a list of some of the more important facilities that NCCF provides.

- Multiple network operators are supported. Each may have different assigned responsibilities, and they may be at various locations

throughout the network. They may use VTAM commands to display and modify the network.

- NCCF running under ACF/VTAM with the multisystem Networking Facility allows an operator to communicate with NCCF in another domain, and effect execution of commands in other domains and receive responses.

- An operator's Span of Control may be restricted to a subset of the network's resources.

- A Scope of Commands and operands may be defined so as to make those designated available only to certain operators.

- Security is provided by checking operator logons.

- Command Lists of commands and control statements may be stored and subsequently executed. Variables may be substituted and conditional execution is provided.

- Commands or command lists may be initiated by timer events. An initial command or command list can be executed when NCCF is started.

- An optional Terminal Access Facility feature is available which can enable an NCCF terminal to function concurrently as a CICS, IMS and HCF console.

9.2 OVERVIEW OF THE NETWORK PROBLEM DETERMINATION APPLICATION

The Network Problem Determination Application (NPDA) Version 2, 5668-983, is a program product which provides an orderly and systematic approach to isolating and correcting problems within a communications network and the host system.

Running as a command processor under NCCF, NPDA collects event and statistical records from those components that issue them, stores them in a data base, and displays them at an NCCF operator station.

By reporting the activities of the hardware and software components indicated by these records, NPDA identifies potential problems and permits them to be resolved before they seriously affect system operation. In addition, NPDA helps to locate failing components and suggests actions to restore their function.

The information generated by resources is collected, organised and interpreted for easy understanding, and may be displayed on a terminal for ready analysis. Included in this information are user-defined and NPDA-generated alert messages, which are high-priority messages used to identify the more serious problems and bring them to the attention of appropriate persons.

A brief summary of NPDA facilities is set out below. More information may be found in the NPDA General Information manual.

9.2.1 NPDA FACILITIES

To assist you in performing problem determination, NPDA:

- Collects, stores and interprets information about detected events from those resources that provide such information.
- Collects, stores and interprets statistical information from those resources that provide such information.
- Issues situation alerts about potential problems to appropriate personnel. Alerts are generated by the occurrence of certain events or when user-set thresholds are exceeded.
- Recommends possible user activities to locate and relieve problems.
- Can transfer selected information to the data base of the Information/Management feature of the Information/System program product, 5735-OZS.
- Allows a user access, via his NCCF display terminal, to the accumulated information, which includes:
  - Identification of the resource.
  - A description of alerts and events.
  - The probable cause of alerts and events based on an analysis of the recorded data.
  - The recommended action that the user may follow to correct or circumvent the problems described in an alert or event.
  - Accumulated statistics about temporary, or recoverable, error events that may be used in the analysis of performance degradation or intermittent failures.

9.3 NETWORK PROBLEM DETERMINATION WITH DPCX

The presence of DPCX/DOSF systems in a network has implications for the network problem determination which is carried on from the host. In particular, the host TP access method cannot "see through" DPCX, and does not know what devices lie beyond it. This section discusses problem determination for DPCX as seen from the host, and the DPCX facilities applicable to the devices beyond DPCX.

9.3.1 DPCX CONSTRAINTS ON HOST NETWORK PROBLEM DETERMINATION

Because an 8100 with DPCX is a computing system in its own right, capable of functioning when the host is offline, the first priority of DPCX when an error occurs is to tell its control operator. Under many circum-

stances, the host system will not know when something goes wrong with DPCX or any terminal downstream of it.

- No alerts are generated by DPCX

  The Physical Unit type 2 in the 8100/DPCX/DOSF system generates no messages to the VTAM SSCP reflecting errors occurring in DPCX.

- No alerts are passed to the host on behalf of, or errors reported concerning, devices downstream of DPCX.

  When devices downstream of DPCX generate messages to the SSCP to report errors, DPCX does not pass them through to the host SSCP.

- There is no programmed operator facility.

  DPCX does not support the capability to create a program to intercept messages to the control operator and handle the conditions to which they refer according to user specifications.

Although DPCX is not transparent to host network management software, it does contain a variety of aids of its own. "Aids for Network Problems Downstream of DPCX" on page 107 suggests how these may be used to perform problem determination in the network downstream of DPCX. "Performing DPCX Problem Determination from the Host" on page 109 discusses accessing these aids from the host site.

9.3.2 DIAGNOSING HOST LINK PROBLEMS

If communications cannot be established between DPCX and the Host, the DPCX tools for link problem determination cannot be used by host personnel as their access to the DPCX system is via the very link which is not working. In this event, the problem should be investigated as far as possible from the host; if this does not reveal the cause of the trouble, then host personnel must specify a course of diagnostic action to the DPCX control operator, who should report the results to them.

9.3.2.1 Diagnostic Actions from the Host

The procedure which follows assumes the use of 386x modems on the host link in conjunction with NPDA on the host. If 386x modems and/or NPDA are not employed, various manual modem wrap tests and other procedures will have to be used to gather equivalent information.

If the host link is not working, it will probably be the DPCX control operator who reports the problem. Follow these steps:

1. Use the VTAM display commands to determine the status of the network resources constituting the DPCX system and the network path to it. If there was a failure in the host or in the network out to and including the NCP with which the DPCX system communicates, correct that first.

2. If the DPCX system had been communicating with the host, and then a failure occurred, NPDA should have reflected an alert to an NCCF console. By selecting that alert, from the information obtained from the

386x modems, NPDA should venture its opinion as to whether the failure was in the local modem, TP link, remote modem or in the 8100. In the last instance, all NPDA can know is whether or not the DPCX system is responding to the modem.

3. If the DPCX system was in a normal inactive state, but the PU does not activate, use the NPDA TEST command to obtain the same information as is collected automatically in the event of a failure.

4. If the above steps indicate no errors, you know that you have a good network path to the DPCX system and you will need to follow the course of diagnostic actions from DPCX which follows this section.

In the event that the PU is active, but some part of host communications will not work or malfunctions, VTAM traces are available to assist in diagnosing the problem by tracing sessions to LUs or the PU.

The NLDM program product, 5668-971, has very useful trace capabilities. It offers significant usability features, such as:

- Supporting interactive display of traced session data from an NCCF terminal

- Optionally having the trace switched on for all sessions, or all of a certain type, (for instance SSCP to LU), so that errors may be traced as they occur without having to re-create the problem.

See the NLDM General Information manual for more information

9.3.2.2 Diagnostic Actions from DPCX

If network problem determination from the host fails to identify the cause of the problem, or suggest some specific thing to investigate at DPCX, have the control operator follow this course of action:

1. Issue the QUERY TASKS command. If program number 954 is not present running on LA E0, host communications has not been started or has terminated: issue the ENABLE HOST command.

2. If the host link had been started, but has terminated, or if program 954 is not running after an ENABLE HOST command, check that a system message has not been issued to the control operator. If necessary, checkpoint and print the journal to review the messages to operator 1. If there was a message, follow the action suggested in DPCX DOSF Messages and Codes.

3. Issue the DISPLAY HOST command. Report the results to the help desk.

The procedure from here on should be followed in co-operation with the help desk or network operations.

4. If the modem is not shown as connected, there may be a fault with the TP line or modem equipment. If the host link employs a switched connection, it is possible that it is not connected. Re-dial it to make certain. Host network operators should be able to diagnose a fault by specifying modem wrap tests to be carried out, or using NPDA if the modems have link test capability.

5. If the host tests work, there may be a fault in the 8100 adaptor or modem interface. The diagnostic tests available with SYSHOST option group 4 should fail if this is the case.

   Note: It is strongly recommended that you configure DPCX using SYSIMOD option group 1 for automatic tests, so that this diagnostic is run automatically each time the host is enabled or it fails.

6. If the host wrap tests were successful but the DPCX diagnostic failed, inspect the physical cable connection between the 8100 and the host link modem. If this properly fastened in, use the 8100 system verification diskette obtained when the 8100 was installed to check the 8100 hardware and modem interface. If this test fails, contact your service representative for assistance.

7. If the modem is shown as connected, but the 3705 is not, the NCP is not polling the 8100. The network operator should ensure that the NCP is functioning correctly and the host network resources which represent the 8100 are not deactivated.

8. If either of the two above conditions applies, and the suggested actions have not corrected the problem, make certain that the SYSIMOD option group 1 values which define the host link parameters are set correctly. Prior to release 3 of DPCX you can do this by stepping through the responses to SYSIMOD option group 1, not changing those values which are correct. For release 3, you can display the values.

   In particular, having any of the following incorrect will ensure that host communications **cannot** be enabled.

   - Station Id (for a switched connection)
   - Station address
   - NRZI encoding
   - Request-to-send setting

   If anything is wrong, correct it with SYSIMOD.

9. If the 3705 is shown as connected, but the host is not, the host system has deactivated the processor. The network operator should attempt to re-activate it.

10. If everything so far is active, check

    a. The 8100 has enabled some LUs; if it has not use SYSHOST option group 1 to enable logical units.

    b. The host has enabled some LUs; if it has not the network operator should attempt to activate them.

    c. The host and DPCX have enabled the same LUs. Use the DISPLAY HOST command with the LU operand to look at the status of the individual LUs which you expect to use. If different LUs are being activated you have some network administration to sort out.

If a DPCX software malfunction is ever suspected in connection with host processing, SYSTRACE (for releases prior to 3.0) and the TRACE command (for release 3.0 or later) contain the capability to perform host internals traces. Prior to DPCX release 3.0 the data is wrapped in a storage

buffer; for release 3.0 and subsequent it is spooled to a user dataset. See the DPCX DOSF Diagnosis Guide for details.

You should not normally expect to use this trace unless requested to do so by the IBM Support Centre. Tracing communications on the link is better performed from the host using standard VTAM traces[11].

9.3.3 AIDS FOR NETWORK PROBLEMS DOWNSTREAM OF DPCX

Because the DPCX system is not transparent to host network management software, it is necessary to use DPCX techniques to perform problem determination in the network downstream of the 8100. These tools are accessible from the host in one of two ways:

- By using them from a terminal connected via Host Command Facility (HCF), which is described in "Introduction to Host Command Facility" on page 109, in the same way as they would be used from a native DPCX terminal.

- Some are implemented as functions of Host Prep SYSINFOREF, and these are described under appropriate headings in "Performing DPCX Problem Determination from the Host" on page 109.

These are the DPCX facilities which will be of use for problem determination in the network downstream of DPCX:

- Journalling the control operator: messages which report errors from devices downstream of DPCX will be issued to the DPCX control operator, user number 1. It is possible to "journal" the control operator so as to retain a log of these messages.

- Displaying the network: the QUERY TASKS command will display all devices currently allocated by DPCX. QUERY DEVAD= may be used to display the status of an individual downstream network resource.

- Network Configuration: the SYSCONFG and SYSIMOD DPCX System Services define, and can display, the configuration of the downstream network. The ENABLE and DISABLE commands may be used to activate and deactivate network resources.

- Retrieving incident records: errors occurring in the network downstream result in entries being logged in the DPCX Condition Incident Log (CIL). The CIL has five types of records; those resulting from incidents relating to downstream network resources are type 3. Type 1 incidents result from errors with 3270 or 3730 terminals attached to the display printer adaptor.

  SYSINFOREF provides a function to retrieve these records to the host. The RETRIEVE CIL statement followed by record selection options can be used to select just the records for one device.

  Alternatively, the ELSA function of DPCX R3 could be used via HCF. This allows CIL records to be interpreted and displayed on a screen,

---

[11] The host NLDM program product, 5668-971, if installed, has function which can reduce the need to take traces.

and the resulting report can be saved in the DPCX spool and retrieved as print records by SYSINFOREF.

- Retrieving control unit error logs: SYSINFOREF contains function to cause DPCX to retrieve, and pass back to the host, the control unit error logs of 3276 and 5210 devices. The RETRIEVE FIDLOG statement performs this service.

- Testing the downstream network: The DPCX System Service SYSTCM will perform online tests of DPCX links and loops and the downstream terminals that attach via them. SYSTCM also supports terminals which are connected to the 8100 display printer adaptor.

  In DPCX R3, SYSTCM has a new English-language (rather than Hex.) user interface, making it much easier to use than in earlier releases; also new in DPCX R3 is the TESTLNK command, which allows the diagnostic functions of IBM 386x modems on a downstream link to be exercised from a DPCX terminal or, via HCF, from the host.

- Invoking traces: DPCX supports four types of trace of its downstream communication ports:

  - SYSTRACE PCA SNA internals (DPCX release 3: TRACE PCA link); this traces internals of the DPCX code driving PCA loops and links.

  - SYSTRACE PCA I/O internals (DPCX release 3: TRACE PCA terminal); this traces internals of the DPCX code driving PCA terminals.

  - SYSCOMTF PCA SNA protocol trace; this can be invoked and retrieved with SYSINFOREF.

  - SYSCOMTF PCA data trace; the trace can be invoked, and its data retrieved, with SYSINFOREF.

  Of these, you are more likely to need the last two traces than the first two.

The DPCX DOSF Diagnosis Guide for release 3 contains a chapter "Guidelines for Handling Link and Loop (PCA) Problems" to guide you in the use of these tools for determining problems arising with terminals thus attached.

10.0 PERFORMING DPCX PROBLEM DETERMINATION FROM THE HOST

This chapter describes how DPCX facilities may be accessed and used by host personnel for diagnosing problems in the DPCX system or in the network downstream of it. It does not give a comprehensive description of the purpose or capability of these tools. More information on each may be found by referring to the publication indicated in Figure 50 on page 100. Education courses are available on DPCX problem determination; contact your IBM representative for details.

An introduction to Host Command Facility is followed by a consideration of ways to capture DPCX displayed or printed data at the host, and then the various aids are considered in the following broad categories:

- General aids
- System problems (DPCX, DOSF and 8100 system units)
- Disk and dataset
- Application software
- Network

10.1 INTRODUCTION TO HOST COMMAND FACILITY

The Host Command Facility (HCF), 5668-985, is a program product which gives a host system terminal user the ability to:

- Interactively operate and control DPCX/DOSF operations within an 8100 on an SDLC communication network.
- Use the service facilities of an 8100.
- Use DPCX application programs in a connected 8100.
- Perform problem determination and error diagnostics throughout the network.

HCF allows several host users to access different, or the same, DPCX systems at the same time. There is a limit of three users who may concurrently access any one DPCX system via HCF. HCF also supports 8100s running the DPPX operating system.

Using HCF, any VTAM controlled display terminal (or NCCF terminal using the Terminal Access Facility), can operate as if it were a 327x 1920 character display directly attached to DPCX. In this way, a user may issue DPCX or DOSF commands, use DPCX and DOSF System Services and run user or IBM-supplied application software from the host terminal, with the following exceptions:

- Programs cannot use a host-attached display or printer as a secondary or tertiary device.
- Programs requiring DOSF text edit and entry capability, will not work if invoked from an HCF-connected terminal. Even if the terminal is an

IDTF 8775 accessing HCF from an 8100 via DSC, DPCX does not support text edit and entry on an HCF terminal.

HCF is an invaluable asset when attempting to perform DPCX problem determination from the host, as in many instances it will allow the host user, who has greater expertise than the users at the 8100 site, to re-create and study for himself a problem which they have encountered.

For more information on HCF, and details of its operation, consult the HCF Guide and Reference.

10.2 CAPTURING DPCX DISPLAYED OR PRINTED DATA AT THE HOST

Many of the procedures that are described in the following sections involve the use of DPCX System Services to display or print data from a DPCX system. This section suggests ways in which this data may be captured at the host, according to whether the System Service is invoked on a DPCX terminal at the 8100 site, or at the host site on an HCF-connected terminal.

1. Control operator using a DPCX System Service on a DPCX terminal at the 8100 site:

   a. Small quantities of data may be displayed or printed and the information manually reported to host personnel.

      Note: For a problem with which host communications cannot be made to work, you will need to resort to this technique.

   b. Data may be printed - that is, generated as a print sequence number on the spool dataset - and retrieved to the host with SYSINFOREF[12]. The control operator must tell host personnel which PSN contains the data.

2. Host personnel using a DPCX System Service from the host on an HCF-connected terminal:

   a. Small quantities of data may be displayed and the information manually recorded, or captured using control-unit local copy to a printer associated with the HCF terminal.

   b. Data may be printed - that is, generated as a print sequence number on the spool dataset - and retrieved to the host with SYSINFOREF[12].

   Note: HCF cannot be used to investigate a problem involving DOSF text edit and entry.

Examples of DPCX data which you might want to obtain in hard copy at the host for problem determination include:

- Control Operator Journals from the time of an error.

---

[12] With releases of Host Prep prior to 5.0, SYSINFOREF does not format the print records, which consist of compressed SNA character strings, and the listing of them is not readily intelligible.

- Configuration (SYSCONFG and SYSIMOD) data, to ensure that the system is correctly set up.

- SYSRIS dumps of disk or storage records

10.2.1 PURGING PSNS AFTER RETRIEVAL TO THE HOST

Particularly when print spool records are being generated by host personnel, procedures need to be instituted to ensure that the DPCX control operator leaves them alone - that is, neither prints nor deletes them. Conversely, after SYSINFOREF has retrieved them to the host, a procedure for deleting print spool records is needed; SYSINFOREF does not delete them after retrieval.

Print spool records can be purged by command from an HCF-connected terminal, or by the control operator at the 8100 site, but it may be considered more convenient to delete them in the same SYSINFOREF jobstream as retrieves them. What follows is a technique by which this may be accomplished[13].

The method involves DPCX user programming, and allows a SYSINFOREF (or DSX) host session to invoke any DPCX PROC as a subtask. Success or failure messages, and messages from the PROC, are reflected to the host and can be retrieved by a subsequent SYSINFOREF job.

To use the process, a DPCX user number is designated to receive messages from the host; the messages are in fact DPCX commands for execution by the host-initiated subtask. It is suggested that the DPCX user number be reserved - that is, not used by any DPCX operator, to avoid any possible retrieval of the messages from the host which would give confusing results both for the operator and for host personnel.

This process only gives the SYSINFOREF user access to DPCX commands, and only those that issue no prompts to the operator. It gives him no access to DPCX System Services. As what is available to him is only a portion of the function provided by HCF, and is often less convenient to use, this technique should in no way be seen as a generally viable alternative to HCF. Its use for any given purpose should be carefully evaluated. However, the following are instances in which it may be effectively employed:

- For the purpose addressed in this section, of deleting print records previously retrieved with SYSINFOREF.

- To execute a PROC on each 8100 in the network, for instance those described under "PART 2: CONFIGURATION MANAGEMENT" on page 77.

The SYSINFOREF control statements to retrieve and delete PSN 31745 from DPCX system 'AA' are shown in Figure 51 on page 112 (use the same JCL as in Figure 61 on page 127), and the PROC in use at the DPCX system in Figure 52 on page 112. (How the PROC could be distributed to the DPCX systems is suggested under "Distributing PROCs from a Host to DPCX Systems" on page 81.) A PROC will be necessary in most instances to allow the DPCX command to be issued from user number 251 under which the host-initiated

---

[13] This way of handling print spool records requires Host Prep release 5.0 for the specification of 'ALL' retrieving the PSN and to initiate the subtask.

subtask runs. The example shown uses a further capability of the sample program to return messages to the host when invoked from a PROC with an *CALL statement.

The example assumes that user number 150 has been reserved for the purpose of communicating the instructions from the host, and that the program that the host initiates as a subtask is number 304. The source code of a working sample is appended in "Sample Programs" on page 137. Once assembled, such a program may be distributed in the manner described under "Maintaining Application Software" on page 63. The status messages generated by the subtask may subsequently be retrieved by the SYSINFOREF control statements in Figure 53 on page 113. If this technique were in regular use the RETRIEVE HOST MSG statement would probably be used at the start of every SYSINFOREF session to retrieve any messages from previous host-initiated subtasks.

```
.&     GENERATE     SYSID(TEXTAA10)                              ;
.&     RETRIEVE     RECORDS(PRINT)                               ;
31745 ALL
.&     SEND         MSG
.&                  OPID(150)            PGMID(304)              ;
XPURGE PSN,31745
.&     INITSUB(304)                                              ;
.&     END                                                       ;
.&     STARTCOM     SYSID(TEXTAA10)                              ;
.&     LIST         RESPONSE(ALL)        SYSID(TEXTAA10)         ;
```

Figure 51. Retrieving and deleting print spool records with SYSINFOREF: these control statements will retrieve and delete PSN 31745 if the sample program 304 is available at DPCX system 'AA'. Host Prep release 5.0 is required.

```
*PROC ,
*CMOFF
*IF &USERNO EQ 251 *SKIP 2
*CO XPURGE: PROC FOR HOST USE ONLY.
*EXIT
*IF '&1' EQ 'PSN' *SKIP 2
*CALL 304 XPURGE - OPERAND 1 INVALID.
*EXIT
*IF '&2' EQ '' *GOTO -ERROR2
PURGE PSN=&2
*IF &RETCODE EQ 0 *EXIT
*CALL 304 XPURGE - RETURN CODE &RETCODE FROM PURGE PSN=&2.
*EXIT
-ERROR2 *CALL 304 XPURGE - OPERAND 2 INVALID.
*EXIT
```

Figure 52. Sample PROC to delete print spool records: the 'XPURGE' PROC allows a host-initiated subtask to delete print spool records.

```
.&      GENERATE     SYSID(TEXTAA10)                          ;
.&      RETRIEVE     HOST                MSG                  ;
.&      END                                                   ;
.&      STARTCOM     SYSID(TEXTAA10)                          ;
.&      LIST         RESPONSE(ALL)       SYSID(TEXTAA10)      ;
```

Figure 53. Retrieving host-initiated subtask status messages: these control statements will retrieve messages generated by the host-initiated subtask.

10.3 USING GENERAL AIDS

The aids referred to below are of general applicability to all sorts of DPCX problems.

10.3.1 JOURNALLING THE CONTROL OPERATOR

It is suggested that the DPCX control operator be journaled. This will help to ensure that control operator actions and system messages are recorded in the event of problems arising. To set this up on a DPCX system

- Issue the DEFINE JOURNAL=YES command from user number 1.
- Ensure that the control operator's number (1) is defined with the operand JO=YES.

If problems arise which need investigation from the host, issue the command JOURNAL CHKPT from user number 1, or other authorised user. This will cause the current PSN to be closed. The spool records may then be browsed at the host using an HCF-connected terminal, or retrieved to the host and printed as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

Having journaling enabled causes a certain overhead on the DPCX system. To limit this:

- Use the control operator signon only for system control functions.

If, however, the system is very busy, journaling may be suspended temporarily by issuing the command JOURNAL OFF. It will also be necessary to suspend journaling in order to use certain control operator functions which will not tolerate concurrent activity, but these are not often required in day to day operation of the system.

10.3.2 RETRIEVING THE CONDITION INCIDENT LOG

SYSINFOREF contains function supporting the retrieval of condition incident log (CIL) records from a DPCX system. Using the RETRIEVE CIL state-

ment it is possible to retrieve the latest incident, all incidents or a count of incidents. These functions may be applied to all incidents in the log, all incidents of a particular type (types 1 through 5 exist), all incidents of a given type and a specific device address or system condition, or a range of incidents using their sequence numbers.

Unfortunately a range of incidents by date and time is not supported. In consequence, the best policy is to retrieve the CIL records from all DPCX systems regularly, say once a week, and clear the logs (this function is also supported from the host). Figure 54 shows how to do this. Use JCL as in Figure 61 on page 127, though you might want to leave the output held on the spool, or write it to disk or tape, and print or inspect it only if it is required to support problem determination.

If you are clearing the CIL regularly, when problems arise you may run a transmission to retrieve all the relevant incident records, without receiving overwhelming volumes of out-of-date data.

```
.&      GENERATE      NETWORK                                    ;
.&      RETRIEVE      CIL                                        ;
2
11
8
.&      END                                                      ;
.&      STARTCOM      NETWORK                                    ;
.&      LIST          RESPONSE(ALL)                              ;
```

Figure 54. Retrieving condition incident log records: these control statements will retrieve all CIL records from each 8100 in the network and print them. The logs are cleared.

10.4 SYSTEM PROBLEMS

The tools that follow are applicable to general system problems with DPCX and DOSF. Errors occurring in disk space allocation or in datasets are considered in "Disk and Dataset Problems" on page 117.

10.4.1 VALIDATING THE CONFIGURATION

Using HCF, it is possible to display a system's SYSCONFG and SYSIMOD parameters at a host terminal. Selecting the print option of SYSCONFG and, for DPCX release 3 or later, of SYSIMOD you may generate configuration listings as records in the spool dataset. These may be retrieved to the host as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

The DOSF System Service SYSXFEA3 may also be invoked. This has an option to display the DOSF options configured with SYSXGEN. The data may either be displayed or spooled to a PSN where it is available for retrieval to the host.

Before looking further for the cause of a problem, it is wise to start here and check that no changes to the configuration have been made which might have caused the trouble.

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSCONFG or SYSIMOD.

10.4.2 EXAMINING PERFORMANCE

SYSDC may be started or stopped from the host by signing on via HCF with any DPCX operator number. It monitors processor, DASD and storage utilisation. On a system installed with ASSIST the disk layout is predetermined, reasonably optimal and there is little opportunity to change it while still retaining ASSIST capability. The data captured by SYSDC in relation to storage utilisation comprises statistics of buffer pool usage. You may use this data to determine how best to distribute storage (with SYSCONFG) between resident programs, associative storage, transient buffers, multiple task storage and PCA buffers.

The data SYSDC collects may be displayed on the HCF terminal, or it may be printed and the PSN retrieved to the host with SYSINFOREF as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

10.4.3 DPCX SYSTEM FAILURES

There should be procedures which instruct the control operator to take a dump of processor storage when a DPCX system terminates abnormally, or enters a loop or wait condition. Instructions for this are in the DPCX DOSF Basic Operations manual. The dump program is IPLed from a diskette, and the dump is written either to diskettes or to an area on disk.

A system dump may be retrieved to the host, and formatted there, using SYSINFOREF. If the dump is on diskettes the control operator has to load them in response to codes which appear on the 8100 operator panel. Note that you might have to retain the dump on diskette to send to the IBM support centre.

Figure 55 on page 116 and Figure 56 on page 116 show how to retrieve a dump from disk or diskette respectively. The numbers following the RETRIEVE SYSDUMP statement indicate the components to be retrieved. See the DPCX DOSF Diagnosis Guide for details as they change from release to release of DPCX. The LIST RESPONSE statement will produce an unformatted listing of the dump, while the LIST FORMDUMP statement yields a formatted one. The JCL to use is that in Figure 61 on page 127.

You may also retrieve the task virtual storage paging areas; see the Host Prep SYSINFOREF Guide and Reference for the RETRIEVE SYSDUMP TVSPA statement.

```
.&     GENERATE     SYSID(TEXTAC10)                          ;
.&     RETRIEVE     SYSDUMP              FDISK               ;
01 02 03 04 05 20 21 24 75 77 80
.&     END                                                   ;
.&     STARTCOM     SYSID(TEXTAC10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TEXTAC10)     ;
.&     LIST         FORMDUMP             SYSID(TEXTAC10)     ;
```

Figure 55. Retrieving a system dump from disk: these control statements will retrieve a system dump which was taken on disk from a DPCX system.

```
.&     GENERATE     SYSID(TEXTAC10)                          ;
.&     RETRIEVE     SYSDUMP              RDISK               ;
01 02 03 04 05 20 21 24 75 77 80
.&     END                                                   ;
.&     STARTCOM     SYSID(TEXTAC10)                          ;
.&     LIST         RESPONSE(ALL)        SYSID(TEXTAC10)     ;
.&     LIST         FORMDUMP             SYSID(TEXTAC10)     ;
```

Figure 56. Retrieving a system dump from diskette: these control statements will retrieve a system dump which was taken on diskette from a DPCX system.

10.4.4 TRACING SYSTEM ACTIVITY

The DPCX traces may be used from the host via an HCF-connected terminal. They trace various system events, in a way that differs according to the release of DPCX. The DLA protocol and data traces are considered separately under "Network Problems" on page 120.

1. For DPCX systems prior to release 3, traces are invoked with the SYSTRACE System Service, and the data is collected in a buffer in storage. Using the SYSRIS DPCX System Service this buffer may be displayed at an HCF terminal, or printed to the spool dataset. The spool records may be retrieved to the host as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110. They are also formatted along with a system dump, if a trace was active when the dump was taken.

   The SNA internals trace, SYSTRACE type 7, is an exception in that it is spooled to a system dataset. This trace may be formatted to the spool dataset with SYSTROUT, and the resulting PSN browsed, or retrieved to the host as above.

   Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSTRACE, SYSTROUT or SYSRIS.

2. With DPCX release 3 and subsequent, SYSTRACE is replaced by the TRACE command. Signing on via HCF, a host user can use this to allocate a user dataset to which traced events are spooled when a trace is start-

ed (with the TRACE command). Another variation of the command causes these records to be formatted to the print spool. From here the HCF user can browse the PSN, or retrieve it to the host as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

10.4.5 DISPLAYING PROCESSOR STORAGE

Logical storage of an 8100 may be inspected from the host while DPCX is operating. Use the SYSRIS DPCX System Service from a terminal connected via HCF. Storage blocks may also be printed with the print option, and the resulting print spool records retrieved to the host as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSRIS.

10.4.6 ANALYSING TAPE VOLUME ERRORS

A host user may use the DPCX SYSLTSD System Service from an HCF-connected terminal to analyse tape errors by volume. SYSLTSD will display errors by volume for up to eighty tape volumes. The same data may be printed, and the spool records produced may retrieved to the host as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSLTSD.

10.5 DISK AND DATASET PROBLEMS

The tools described below can be used to report on disk and dataset layouts, and to examine blocks on disk and records of datasets.

10.5.1 LISTING CATEGORY AND DATASET ALLOCATIONS

A user at the host can, from a terminal signed on to HCF, display the assignment of bitmaps and datasets to categories and the allocation and characteristics of datasets. The DPCX System Service SYSLDSA is used for this purpose.

The same information may be printed to the spool, again using SYSLDSA, and the print records retrieved to the host with SYSINFOREF as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSLDSA. This System Service will not tolerate concurrent activity in

the DPCX system. All tasks must be stopped, other users signed off and only one host-initiated session enabled (which is used by HCF).

10.5.2 EXAMINING DISK AND DATASET RECORDS

Blocks on disk and dataset records at an 8100 may be inspected and optionally altered from the host while DPCX is operating. Use the SYSRIS DPCX System Service from a terminal connected via HCF. The same data may also be printed with the print option, and the resulting print spool records retrieved to the host as described in "Capturing DPCX Displayed or Printed Data at the Host" on page 110. (SYSRIS may also be used to inspect and alter diskette records).

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSRIS.

10.5.3 VALIDATING AND RECOVERING DISK SPACE

A user at the host may validate the integrity of datasets and even recover lost disk space, subject to the constraints mentioned below. The SYSDVAR DPCX System Service is used for this purpose, and it also includes facilities to hash index dataset keys to member pointers, and decode bit map entries to disk addresses and the converse.

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSDVAR. The function which recovers disk space will not tolerate concurrent activity in the DPCX system. All tasks must be stopped, other users signed off and only one host-initiated session enabled (which is used by HCF).

The disk space recovery function insists on using a printer to maintain an audit trail. This is a tertiary printer, not spooled, and must be a device attached to the DPCX system. In consequence, before attempting to recover disk space from the host, you need to identify a DPCX printer which is ready with sufficient paper available.

As recovering disk space must be preceded by validating datasets, this process may take several hours (the time required will increase with the number of disks that DPCX is using). You may conclude that these considerations make it preferable to run the disk space recovery function at the DPCX site.

10.5.4 RECOVERING THE PRINT SPOOL DATASET

A user at the host can, from a terminal signed on to HCF, invoke the DPCX System Service SYSSPREC to recover the print spool dataset. Use SYSSPREC when spool file errors are indicated at IPL or during system operation. It produces no output except a completion message indicating success or failure.

Note: The host user must sign on either as the DPCX control operator (user number 1) or as the service representative (user number 255) in order to use SYSSPREC. This System Service will not tolerate concurrent activity in the DPCX system. All tasks must be stopped, other users signed off and only one host-initiated session enabled (which is used by HCF).

10.5.5 DETERMINING THE STATUS OF DOSF FILES

The DOSF System Service SYSXFEA3 may be used to obtain a summary of the status of the DPCX datasets used by DOSF. This program can be invoked at an HCF-connected terminal by a user signed on with any DPCX operator number, and will display its output at the terminal or spool it to a PSN. The latter option allows the retrieval of the information to the host using the method of "Capturing DPCX Displayed or Printed Data at the Host" on page 110, but as there is only one screen image involved it is probably more easily read from the display terminal.

10.6 APPLICATION PROBLEMS

The following techniques may be of use in isolating problems in application software.

10.6.1 TRACING FUNCTION PROGRAMS

The DPCX Program Execution Monitor, accessed by the two system services SYSDEBUG and SYSTEST, gives a DPCX user the capability to monitor the execution of application Function Programs (FPs). SYSDEBUG is invoked from the monitor terminal, and can either trace a subtask, which it will initiate for you, or programs executing on a primary device. In the latter case, the device is linked to SYSDEBUG by invoking the SYSTEST System Service from that terminal.

The services both of SYSDEBUG and SYSTEST are available to a user at the host with an HCF-connected terminal. If you wish to trace a task executing on a primary device, you will need two HCF-connected terminals, one on which to run SYSDEBUG and one on which to invoke SYSTEST and the program to be monitored.

When using SYSDEBUG via HCF, it is not possible to use its printing or screen copy facilities. These require a printer on the same control unit as the SYSDEBUG terminal, but DPCX cannot connect to a host-attached printer. You may of course use device-initiated control unit local copy to copy display screens to an available printer.

10.6.2 RETRIEVING FUNCTION PROGRAM ABEND DUMPS TO THE HOST

Host Prep SYSINFOREF will retrieve FP ABEND dumps to the host and print

them there. The control statements shown in Figure 57 on page 120 will cause the retrieval of all dumps at the the DPCX system and delete them from the DPCX FP ABEND dataset.

There is no capability to selectively print one dump, and the dump dataset can hold a considerable amount of dump data, so if you will need to look at FP ABEND dumps you should institute procedures so that the dataset is purged regularly with the PURGE PGMDUMP command when dumps are not required. Alternatively, you may prefer to purge the dataset after an ABEND from which you wish to see the dump. Then re-create the ABEND so the dump of this condition is the only one present, and then retrieve dumps to the host. If the dumps are neither purged nor retrieved they will eventually fill the dataset and no more will be taken.

Remember that IBM application software such as DISOSS is implemented as user FPs, and a problem involving such a product might result in the IBM Support Centre requesting you to print an ABEND dump. The same could hold true of an ABEND in a system service; System Services are also DPCX FPs.

```
.&     GENERATE     SYSID(TEXTAF10)                              ;
.&     RETRIEVE     DUMPS                                        ;
.&     END                                                       ;
.&     STARTCOM     SYSID(TEXTAF10)                              ;
.&     LIST         RESPONSE(ALL)        SYSID(TEXTAF10)         ;
```

Figure 57. Retrieving Function Program ABEND dumps to the host: these control statements will retrieve all function program ABEND dumps from a DPCX system to the host.

10.7 NETWORK PROBLEMS

The techniques here are applicable to the network downstream of an 8100.

10.7.1 TESTING THE DOWNSTREAM NETWORK

The QUERY command may be used by a host user on an HCF-connected terminal to inspect the status of devices attached to DPCX. SYSTCM may be used in a similar way to run its variety of tests to DPCX devices. Neither of these functions produces any printout, and the information desired, if not used immediately, can usually be noted down easily. The response to QUERY DEVICES will be copious if the DPCX system has several loops and links configured; capturing this data would be a good candidate for control unit local copy from the HCF screen. Alternatively, the command could be issued from an operator which is being journaled and, once check-pointed, the PSN retrieved to the host using the technique of "Capturing DPCX Displayed or Printed Data at the Host" on page 110.

10.7.2 CONTROLLING PCA TRACES FROM THE HOST

DPCX supports SNA protocol or data traces of its links and loops, spooling the traced data to a system dataset for subsequent formatting. SYSINFOREF contains function to allow you to control this from the host, and to receive and print the information there. Figure 58 and Figure 59 show respectively:

- How to start PCA data and protocol traces on DPCX using SYSINFOREF
- How to stop DPCX PCA data and protocol traces with SYSINFOREF and retrieve the spooled information to the host.

This SYSINFOREF option invokes the same traces from the host as does the System Service SYSCOMTF on DPCX.

```
.& GENERATE SYSID(TEXTAA10);
.& SEND DLADATA START DLA(81) LNG(16);
.& END;
.& GENERATE SYSID(TEXTAB10);
.& SEND DLAPROTOCL START DLA(10);
.& END;
.& STARTCOM NETWORK;
.& LIST RESPONSE(ALL);
```

Figure 58. Starting PCA protocol and data traces at DPCX systems: these control statements will start PCA data and protocol traces on respective 8100s.

```
.& GENERATE SYSID(TEXTAA10);
.& SEND DLADATA STOP DLA(81);
.& RETRIEVE DLADATA DLA(81) TIMEON(000001) TIMEOFF(235959);
.& END;
.& GENERATE SYSID(TEXTAB10);
.& SEND DLAPROTOCL STOP   DLA(10);
.& RETRIEVE DLAPROTOCL TIMEON(000001) TIMEOFF(235959);
.& END;
.& STARTCOM NETWORK;
.& LIST RESPONSE(ALL);
```

Figure 59. Stopping and retrieving PCA protocol and data traces: these control statements will stop PCA data and protocol traces on respective 8100s and retrieve the spooled information to the host.

10.7.3 RETRIEVING CONTROL UNIT ERROR LOGS WITH SYSINFOREF.

Host Prep SYSINFOREF has the capability to cause DPCX to retrieve and return to the host the error logs from attached 3276 or 5210 control units. The counters may be reset to zero. It is a good idea to institute a procedure to retrieve and clear the logs of all such devices regularly,

perhaps once a week, as a high error rate may be indicative of a low-quality communications link, or a loop with intermittent problems.

The control statements in Figure 60 show how to retrieve and reset the error logs for one 3276 on one DPCX system. Use JCL as in Figure 61 on page 127, and for a 5210 omit the RETRIEVE statement for TYPE(2) as this log is not used by this device.

This SYSINFOREF option performs the same function from the host as does the System Service SYSRSLOG on DPCX.

```
.& GENERATE SYSID(TEXTAB10);
.& RETRIEVE FIDLOG TYPE(1) STATION(0100) RESET;
.& RETRIEVE FIDLOG TYPE(2) STATION(0100) RESET;
.& RETRIEVE FIDLOG TYPE(3) STATION(0100) RESET;
.& RETRIEVE FIDLOG TYPE(5) STATION(0100);
.& END;
.& STARTCOM SYSID(TEXTAB10);
.& LIST RESPONSE(ALL) SYSID(TEXTAB10);
```

Figure 60. Retrieving control unit error logs: these control statements will retrieve and reset all the control unit error logs for a 3276.

This part of the manual should be referred to as necessary when more information or examples are required in relation to a topic.

A.0 SYSINFOREF INITIALISATION AND USE

The information in this appendix is intended to assist the user who is commencing work with SYSINFOREF. It is complementary to that contained in the Host Prep SYSINFOREF Guide and Reference which should also be consulted.

A.1 SYSINFOREF VSAM DATASET ALLOCATIONS

The Host Prep SYSINFOREF Guide and Reference describes the VSAM datasets that SYSINFOREF uses, their organisation and the AMS commands to define them, but it gives no guidance as to how much space to allocate. SYSINFOREF uses a minimum of two datasets, BHDWK1 and BHDSSIL, in addition to reading from BQBLIBI.

The BHDWK1 dataset: is where the machine list, the PTF library, the TEMPLIB, the generation file and the response file all reside. Its name is misleading; it should not be thought of as a work dataset, because the machine list and the PTF library comprise permanent data. If you fill this dataset, you should allocate a new, larger one and use AMS REPRO to copy the full dataset into the new one.

Use the following guidelines to calculate the disk space required for the data component of the BHDWK1 dataset:

- Allow four megabytes for the PTF TEMPLIB.
- Allow one megabyte for the PTF library for each set of DPCX and DOSF software levels supported. So if you are keeping PTFs for DPCX release 2.2, feature 6001f and DOSF release 2.1, and also for DPCX release 3.0 and DOSF 3.0, then allow two megabytes for the PTF library.
- Allow half a megabyte for the generation file.
- Initially allow one megabyte per 8100 for the response file, for as many systems as you will contact in one execution of SYSINFOREF. This is particularly variable, as large amounts of disk space can be consumed by retrieving a full complement of FP ABEND dumps or all the condition incident log data from all the 8100s.
- The storage required for the machine list is negligible.

One megabyte of storage in the BHDWK1 dataset occupies about four and a half cylinders of 3330 disk.

The OS/VS version of SYSINFOREF with Host Prep release 5.0 introduces two additional VSAM datasets which off-load BHDWK1. The datasets are BHDWK1 and BHDWK2, and the data which may now be stored in them, instead of in BHDWK1, is as follows:

- BHDWK2 can contain the PTF TEMPLIB
- BHDWK3 can contain the generation and response files

Their use is not mandatory, and they are only used if available, but it is recommended that you do employ them. Having the truly temporary data (the

TEMPLIB and the generation and response files) in separate datasets allows it to be deleted independently of the permanent library data. The advantages of this are:

- The TEMPLIB and the generation and response files may be deleted by deleting and redefining the BHDWK2 and BHDWK3 datasets with AMS. This gives a significant performance improvement over SYSINFOREF deleting this data (which it does one VSAM record at a time), especially when large amounts of data are involved.

- It is easier to adjust the storage allocation for the generation and response files, which depends considerably on how the network is operated and what SYSINFOREF is being used for. BHDWK3 may simply be deleted and re-allocated with more space, without the need to worry about preserving the machine list and the PTF library.

The BHDSSIL dataset: requires storage for each version of each application that resides in it, and varies according to the number of modules entailed and their size. IBM products may give you guidance to the amount of disk storage required in the memo to users which accompanies the product; if a figure is given for BQBLIBI, the storage needed in the data component of BHDSSIL will be similar.

A.2 INITIALISING SYSINFOREF

Before SYSINFOREF can be used to handle PTFs, it has to be identified to VTAM and the 8100/DPCX systems have to be identified to it. This can be done by coding control statements to describe the 8100s to SYSINFOREF, but it is easier to run the job illustrated in Figure 61 on page 127, which will establish a session to each DPCX/DOSF system and determine its software level and features.

The first statement names the VTAM application id that the installation has defined for SYSINFOREF. Subsequent ones refer to the batch logical unit of each 8100 that is to be used for SYSINFOREF communications. These must be active to VTAM before running the job. The example shown is for a network of one test and ten production 8100s.

Because SYSINFOREF hasn't actually been asked to do anything with these systems, it may print out the contents of some control blocks in anticipation of a user error. This may be ignored. The LIST MLIST statement will show what SYSINFOREF has found out; check that each system is represented. The LIST RESPONSE statement will report any errors that occurred.

When the network increases with the addition of further 8100s, they may be defined in the same way.

A.3 SYSINFOREF CLASSES

The DPCX/DOSF systems should be grouped into classes for ease of operation. Classes should be selected on the basis of including together systems to which it is desired to make the same changes at the same time. The simplest method, which will suffice in many cases, is to have one class

```
//SIRFINI  JOB   SYSINFOREF
//SETUP    EXEC  PGM=BHDLMS
//STEPCAT  DD    DISP=SHR,DSN=vsam_user_catalog        (if needed)
//STEPLIB  DD    DISP=SHR,DSN=Host_Prep_load_library   (if needed)
//SYSPRINT DD    SYSOUT=A
//BHDWK1   DD    DISP=SHR,DSN=BHDWK1     Replace these dataset
//BHDSSIL  DD    DISP=SHR,DSN=BHDSSIL    names with those defined
//BQBLIBI  DD    DISP=SHR,DSN=BQBLIBI    by your installation
//SYSIN    DD    *
.&     DEFINE       CONTROL              APPLID(SIRF)          ;
.&     STARTCOM     SYSID(TESTZZ10)                            ;
.&     STARTCOM     SYSID(TEXTAA10)                            ;
.&     STARTCOM     SYSID(TEXTAB10)                            ;
.&     STARTCOM     SYSID(TEXTAC10)                            ;
.&     STARTCOM     SYSID(TEXTAD10)                            ;
.&     STARTCOM     SYSID(TEXTAE10)                            ;
.&     STARTCOM     SYSID(TEXTAF10)                            ;
.&     STARTCOM     SYSID(TEXTAG10)                            ;
.&     STARTCOM     SYSID(TEXTAH10)                            ;
.&     STARTCOM     SYSID(TEXTAI10)                            ;
.&     STARTCOM     SYSID(TEXTAJ10)                            ;
.&     ADD          MLIST                SYSID(TESTZZ10)
.&                                        CLASS(900)            ;
.&     ADD          MLIST                SYSID(TEXTAA10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAB10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAC10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAD10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAE10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAF10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAG10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAH10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAI10)
.&                                        CLASS(1)              ;
.&     ADD          MLIST                SYSID(TEXTAJ10)
.&                                        CLASS(1)              ;
.&     LIST         MLIST                                      ;
.&     LIST         RESPONSE(ALL)                              ;
/*
//
```

Figure 61. SYSINFOREF initialisation: sample JCL and control statements to identify SYSINFOREF to VTAM and to identify the DPCX/DOSF systems to SYSINFOREF.

for the test/development machine and another for all production machines. Figure 61 on page 127 includes control statements which will achieve this.

A.4 SYSINFOREF PHASES AND TIMING OF SESSIONS

SYSINFOREF processing is divided into four phases. They are:

Maintenance Phase for updating the SYSINFOREF libraries

Generation Phase for defining the transmissions that are required

Communication Phase for transmitting to the DPCX/DOSF systems

Response Phase for determining the results of these transmissions

These are described in more detail in the introduction to the SYSINFOREF guide and reference.

These phases are distinct stages of activity, which, for planned, controlled maintenance will often be separated in time. For instance

- In the morning SYSINFOREF might to be invoked to copy PTFs retrieved the previous day from the TEMPLIB to the PTF library. This uses the maintenance phase of SYSINFOREF.

- Once that has been achieved, SYSINFOREF generation phase could be invoked in the afternoon to define transmissions to production DPCX/DOSF systems to send and install PTFs.

- A job might be submitted for execution overnight to perform these transmissions (communications phase). This is to run out of prime shift because there is a lot of data to be transmitted and the PTF install process will not tolerate any other active users.

- The next day SYSINFOREF would be used in the response phase to see the results of the transmission.

Important Note: the generation, communication and response phases must be executed serially for one purpose at a time. Invoking the generation phase will:

- Delete any generated data from previous executions of SYSINFOREF.

- Delete any data generated prior to the last invocation of the communication phase.

- Delete any response data.

It is possible to have a number of generates following each other in the same control statement stream - all the generated data will aggregate - but all of it must be transmitted with the communication phase and the responses listed before generating more data.

There will be times when it is needed to send (for instance, a corrective service PTF) to a particular 8100 as fast as possible. SYSINFOREF may also be used for other purposes such as initiating traces or retrieving trace data during the day. As stated above, each SYSINFOREF job that invokes the generation phase will delete any data previously generated for transmission and any response records, so it is no good running the generation phase to set up scheduled maintenance to run overnight and then running a job to generate a transmission to send an urgent PTF to an 8100.

9.00 a.m.		Run response phase for yesterday s transmissions
10.00 a.m.	A PTF is urgently needed at an 8100. Run all four phases to create it in the PTF library, generate a transmission, run it and list responses.	
11.00 a.m.		Perform maintenance phase, loading PTFs into the library for transmission tonight
2.00 p.m.	A DLA link trace is needed at an 8100. SYSINFOREF is run several times to start, stop and retrieve the trace.	
4.00 p.m.		Perform generation phase, defining the transmissions to be run tonight.
5.00 p.m.	A PTF is urgently needed at an 8100. However, as the over-night transmissions are already set up, it must wait until tomorrow.	
6.00 p.m.		A job is released to execute the transmissions

Figure 62. Possible scheduling of SYSINFOREF phases: care must be taken as the generate / communicate / response process works serially.

In consequence, plan to run the generation for the overnight maintenance as the last task of the day. This procedure is represented by the diagram in Figure 62. Remember, the generate / communicate / response process is serial; each set of data generated must be communicated and the responses referenced before generating more data.

Care must also be taken when aggregating generated data for different purposes that it does go only where it is wanted. For instance, generating a PTF transmission for feature DOSF21 with intent to send it to 8100A and generating a request to retrieve a PTF listing from 8100B, and then starting communications to each of these systems thus:

```
.& GENERATE FEATURE(DOSF21);
.& SEND PTF .......   ;
.& END;
.& GENERATE SYSID(LU8100B);
.& RETRIEVE PTF STATUS;
.& END;
.& STARTCOM SYSID(LU8100A);
.& STARTCOM SYSID(LU8100B);
```

would have the additional, undesired result of sending the PTF to 8100B if this had DOSF release 2.1. Moreover, if the same system is referenced by more than one STARTCOM statement, perhaps once by CLASS and once by SYSID, then all the generated data which is applicable to that system will be executed twice. Only the response records for the second set of transmissions will be available - they will have overlaid those from the first set.

It is possible to run multiple copies of SYSINFOREF (see "Running Multiple Copies of SYSINFOREF" on page 131), so that one might be used for scheduled maintenance and another for problem determination and problem resolution activities, but the complexities which this introduces in terms of maintaining multiple libraries and avoiding attempts to transmit concurrently make this worthwhile probably only for larger networks.

A.5 RECOVERY OF SYSINFOREF TRANSMISSIONS

Should a network resource be unavailable or a network component fail, or if the host operating system or SYSINFOREF abnormally terminates, some sessions may not be completed. Each transmission (that is, each matching of a set of generated data against an 8100) will be in one of three states:

- Not started: either SYSINFOREF processing never reached this far, or the 8100 could not be contacted.
- Completed: the failure occurred after this transmission had finished.
- Incomplete: the transmission started, but a failure occurred while it was running.

In the event of a network problem or other failure for which SYSINFOREF can continue processing, it will give an error message and a snap dump of some control blocks, indicating that a transmission failed or could not be started. If host software abnormally terminates, the SYSINFOREF output listing will indicate which transmissions were started. If a level of multi-tasking was not requested on the STARTCOM statement, transmissions will take place serially, and all prior to the last will have been completed; and if there is more than one transmission to a given 8100, all prior to the last one will have been completed. The status of the last transmissions started will be indeterminate.

After a failure, the first thing is to list the response file - .& LIST RESPONSE(ALL) - if this was not done in the job which attempted the transmissions. This may give some indication of the state of sessions which were active when a failure occurred. There are several options for retrying:

- Resubmit just the .& STARTCOM control statements: all transmissions will be retried, including those which completed successfully.

- If all transmissions to some 8100s were complete, submit .& STARTCOM control statements for the other systems: all transmissions for these other systems will be retried, including those which had completed successfully.

- Submit control cards to regenerate data just for those sessions which were incomplete or not started, and transmit just to those 8100s which were affected: this may be quite complicated to achieve when several sessions were being run to a number of 8100s, and it is recommended that, where possible, one of the previous options is used.

When a partially completed transmission is performed again, depending on the function being performed different results may be experienced on the second occasion. For instance, with .& SEND AMEMBER for application products, a re-transmission may encounter some entities already at DPCX from the first attempt and these will not be sent again.

A.6 INTERACTIVE SYSINFOREF

SYSINFOREF has an option to allow a single user to control it interactively. The support allows the control statement inputs to be read from a VTAM 3270 terminal, on which the SYSINFOREF messages will also be displayed. The processing of the control statements is exactly as in batch mode, and all the considerations of the serial execution of the different SYSINFOREF phases still apply. The interactive support does have, though, the advantages of a capability to change incorrect control statements as they are encountered and the possibility, for instance, of browsing a chain of pre-requisite PTFs on the host PTF library.

SYSINFOREF is still initialised with a batch job; the JCL is as for Figure 61 on page 127. The control statement shown in Figure 63 will put the local 3278 with logical unit name TD781305 in session with SYSINFOREF if it is active to VTAM and not in session with another application.

```
.&     STARTDEV     SYSID(TD781305)      MODE(L)                ;
```

Figure 63. Using SYSINFOREF interactively: this control statement causes SYSINFOREF to take subsequent input from 3278 TD781305

A.7 RUNNING MULTIPLE COPIES OF SYSINFOREF

There is one circumstance in which the user with a small network of DPCX systems may want to run two copies of SYSINFOREF on the host system, and that is when user applications are being coded and SYSINFOREF is being used to transmit programs to the development 8100 for testing. This sending of test versions of programs is likely to go on throughout the day, and it does not want to interfere with network maintenance or problem determination on the production machines which also requires SYSINFOREF. Nor is it desirable to fill the production SYSINFOREF libraries with test versions of user applications.

In this instance a second version of SYSINFOREF can be operated by creating second BHDWK1 and BHDSSIL datasets, and by initialising the second copy of SYSINFOREF to have a different VTAM APPLID and defining only the development machine logical unit (TESTZZ10) to it. See Figure 64 on page 132 for an example. Care must still be taken to avoid transmitting to the development 8100 with both copies of SYSINFOREF at the same time, but network management can be performed for production systems while programmers are sending applications to this machine.

```
//SIRFINI  JOB   SYSINFOREF
//SETUP    EXEC  PGM=BHDLMS
//STEPCAT  DD    DISP=SHR,DSN=vsam_user_catalog        (if needed)
//STEPLIB  DD    DISP=SHR,DSN=Host_Prep_load_library   (if needed)
//SYSPRINT DD    SYSOUT=A
//BHDWK1   DD    DISP=SHR,DSN=TESTWK1   *** Different VSAM datasets
//BHDSSIL  DD    DISP=SHR,DSN=TESTSSIL  *** for second SYSINFOREF
//SYSIN    DD    *
.&     DEFINE       CONTROL              APPLID(SIRF2)        ;
.&     STARTCOM     SYSID(TESTZZ10)                           ;
.&     LIST         MLIST                                     ;
.&     LIST         RESPONSE(ALL)                             ;
/*
//
```

Figure 64. Establishing a second copy of SYSINFOREF: sample JCL and control statements to identify a second copy of SYSINFOREF to VTAM and to identify the development system to SYSINFOREF.

B.0 DPCX AND SYSINFOREF PTF HANDLING CAPABILITIES

B.1 OVERVIEW OF DPCX/DOSF PTF HANDLING

This section briefly outlines the possible flows of PTF data between DPCX systems and a host system. It is to be read as background information to the suggested procedures that follow, and as complementary information to that contained in the Host Prep SYSINFOREF Guide and Reference, the DPCX DOSF Diagnosis Guide and the DPCX Programming Guide to Host Communication for System Programmers. Figure 65 on page 134 shows the possible routes that the PTFs can take prior to eventual installation on DPCX/DOSF.

- Usually PTFs arrive on diskettes from IBM.

  Before release 3 of DPCX this diskette must be IPLed, and this process copies the PTFs to a system dataset 8. From release 3 onwards, the operations referred to as acting on system dataset 8 work directly from the diskette and system dataset 8 is not used.

- From system dataset 8 (DPCX R3: the diskette) the PTFs may be copied with the DPCX SYSPTF System Service to system dataset 12, which is the PTF dataset.

  This process selects the PTFs applicable to the software level and features of the DPCX system.

- PTFs may be installed from the PTF dataset onto the DPCX operating system with SYSPTF.

For a DPCX/DOSF ASSIST installation, all the above may be done automatically.

- SYSPTF is also able to dump PTFs to diskette from the PTF dataset.

- PTFs from the diskette can be restored onto another DPCX system's dataset with SYSPTF.

- PTFs may be retrieved to the host using SYSINFOREF.

  - An entire diskette image can be obtained from system dataset 8 (DPCX R3: the diskette).

  - PTFs can be taken from the PTF dataset, selectively if desired.

  In both cases they are retrieved into the SYSINFOREF temporary library, or TEMPLIB.

- From the TEMPLIB, PTFs may be copied to the SYSINFOREF PTF library with the ADD PTF function; selection is possible.

- Once on the SYSINFOREF PTF library, PTFs are available for sending to (production) DPCX systems. They are transmitted to the DPCX PTF dataset.

- From the PTF dataset, PTFs may be installed onto the DPCX system with SYSPTF or SYSINFOREF.

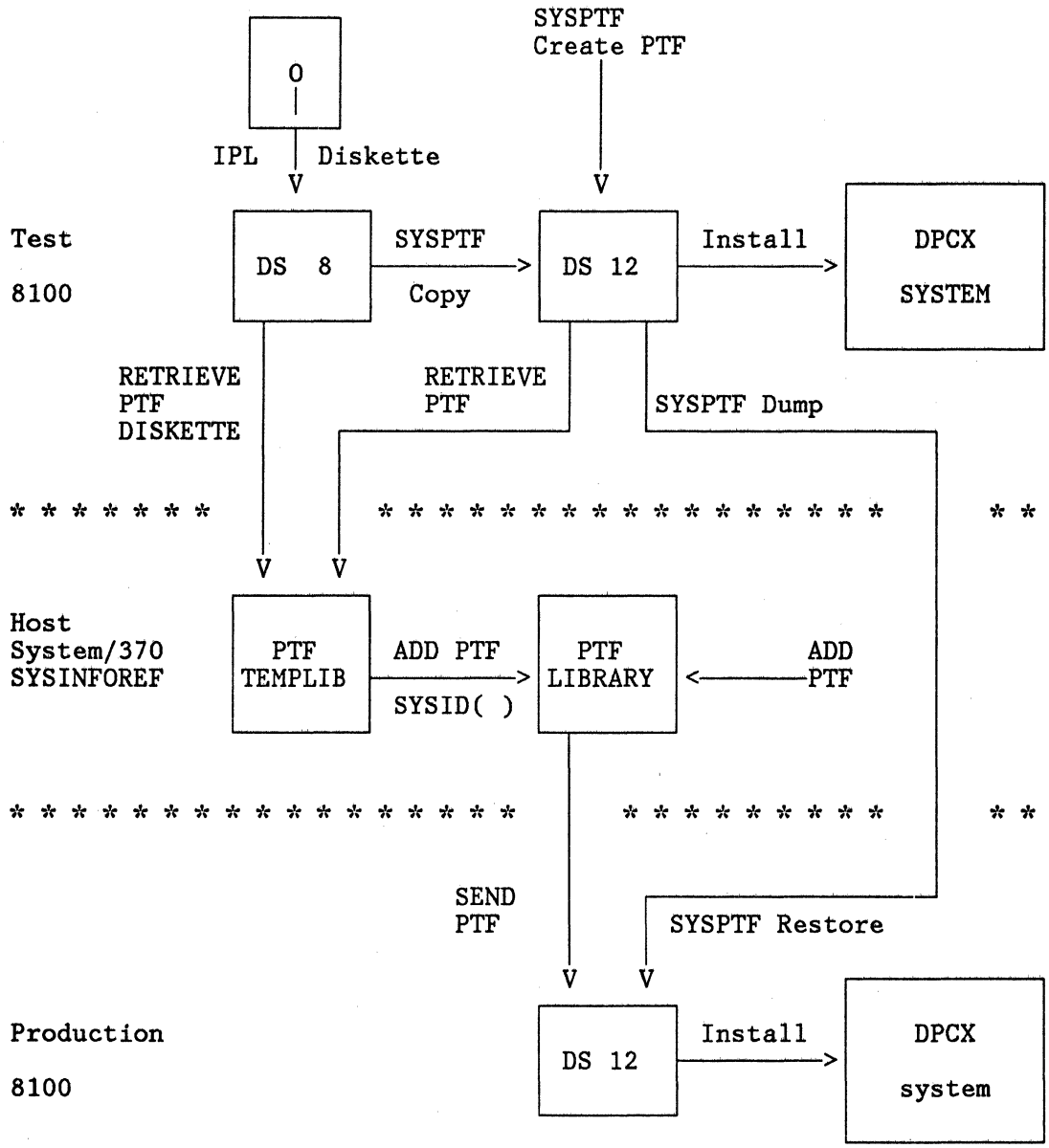


Figure 65. DPCX and SYSINFOREF PTF flows: This figure illustrates the possible flows of PTF records between DPCX systems and a host system/370.

- PTFs for corrective service may be keyed directly into the DPCX PTF dataset using SYSPTF.
- They may also be defined into the SYSINFOREF PTF library at the host using the ADD PTF control statement.

B.2 COMPARISON OF SYSPTF AND SYSINFOREF PTF FUNCTIONS

This section indicates for each option of the DPCX SYSPTF system service, the SYSINFOREF statements that achieve the same function. It also includes some description of these functions. It is provided for the benefit of the user who may have used SYSPTF, but is not yet familiar with SYSINFOREF.

SYSPTF OPTION:	SYSINFOREF STATEMENT(S):
1 Install PTFs	.& INSTALL ALL or .& INSTALL ACTION;

SYSPTF option installs one or all PTFs; unless the PTF is HOLD status SYSPTF will attempt to install it whether or not it is already installed. SYSPTF will report that:

1. It installed the PTF if the target data matched the PTF verify data.
2. The PTF is already installed if the target data matches the PTF replace data.
3. The PTF conflicts if the target data matches neither the verify or the replace data. This may arise if a subsequent PTF having this PTF as a prerequisite has changed the data again.

SYSINFOREF install can either specify installation of all PTFs (that are not held) or just those PTFs which have an action status of install. This status may be assigned when the PTF is sent to DPCX, or by a subsequent .& UPDATE PTF(nnnnnnn) ACTION(INSTALL) statement.

2 Remove PTFs	.& REMOVE PTF(nnnnnnn);
	.& INSTALL ACTION;

SYSPTF will remove a single, named PTF from the system. It remains in the PTF dataset in READY status.

SYSINFOREF will flag individual PTFs for removal; they are removed when the PTF install process is invoked.

3 Dump/Restore PTFs	.& RETRIEVE PTF TEXT(TEMPLIB);
	.& SEND PTF;

The SYSPTF dump/restore function can dump the PTF dataset to diskette on one DPCX system and restore it to the PTF dataset on other DPCX systems.

SYSINFOREF can retrieve PTFs from the PTF dataset on one DPCX system and transmit them to the PTF dataset on other DPCX systems.

4 List/Test PTF Status	.& RETRIEVE PTF STATUS; and
	.& RETRIEVE PTF VERIFY;

SYSPTF list option gives the status of PTFs as recorded in the PTF dataset. The test option compares the PTF verify and replace data to the target data and reports one of the three conditions described for the install option: the PTF is installed, not installed or conflicts.

SYSINFOREF RETRIEVE PTF STATUS functions as SYSPTF list; RETRIEVE PTF VERIFY functions as SYSPTF test.

5 Copy PTFs to PTF dataset .& RETRIEVE PTF DISKETTE;

.& SEND PTF DISKETTE;

SYSPTF option 5 copies PTFs from the temporary dataset (8) - or for DPCX release 3, directly from diskette - to the PTF dataset.

SYSINFOREF will retrieve PTFs from the temporary dataset (8) - or for DPCX release 3, directly from diskette - to the host. From here they may be sent to the PTF dataset of a DPCX system.

6 Verify/Update PTF .& RETRIEVE PTF TEXT(RESPONSE);

SYSPTF option 6 will display the PTF requisites and each patch unit of compare and replace data. It is possible to update this data if the PTF is not installed.

SYSINFOREF will retrieve this information to the host. If a PTF is not installed it may be changed on the host and re-transmitted to DPCX - see the Create PTF option following.

7 Create PTF .& ADD PTF ;

.& SEND PTF ;

SYSPTF option 7 allows manual entry of a PTF into the DPCX PTF dataset.

SYSINFOREF supports manual specification of a PTF at the host site. It is written to the host PTF library, and may be transmitted to the PTF dataset on a DPCX system.

8 Enable/Disable PTF .& UPDATE PTF ACTION(READY); and

.& UPDATE PTF ACTION(HOLD);

A PTF which is not installed may be held (disabled) by SYSPTF option 8. It cannot be installed until it is made ready (enabled).

SYSINFOREF performs the same function with the control statements shown.

This appendix contains working examples of programs employing techniques suggested elsewhere in this manual.

C.1 PROGRAM TO COPY A BDES TAPE BACK TO BQBLIBI

The sample PL/1 program which follows will read a BDES data-stream from tape or diskette and build a BQBLIBI dataset as output. They may then be copied into the SYSINFOREF BHDSSIL library in the normal way. The technique be used to make products distributed on diskettes in BDES format available for distribution via the network.

The sample program could readily be extended to handle DSCBs.

C.1.1 SOURCE OF SAMPLE PROGRAM.

```
BDEBQB:                    /* BDE TAPE TO BQBLIBI DATA SET */

/*********************************************************************
*                                                                    *
*  FUNCTION: READS BDE-FILE ESTABLISHED AT 8100 DPCX                 *
*            BY FDP 5798-DHX AND WRITES THE PROGRAMS                 *
*            AND PANELS INTO THE HOST-BQBLIBI.                       *
*            VERSION- AND LEVEL-SUFFIX FOR THE BQBLIBI               *
*            RECORDS IS DEFAULTED TO 00/00 IF IT IS                  *
*            NOT PROVIDED THRU JCL-PARM FOR THAT RUN.                *
*                                                                    *
*            PRINTS ACTIVITY-REPORT.                                 *
*  AUTHOR  : U.KAEMPFER, WTSC RALEIGH                                *
*  DATE    : JUNE 82                                                 *
*                                                                    *
*  MANUALS REFERENCED ARE:                                           *
*                                                                    *
*  - LY38-3036 HOST PREP LOGIC (FOR BQBLIBI RECORD                   *
*                                LAYOUT)                             *
*  - SC27-0533 INSTALLING HOST-PREP FOR USE WITH OS/VS               *
*              (FOR VSAM-DEF OF BQBLIBI)                             *
*                                                                    *
*********************************************************************/
```

```
PROC (JCLPARM) OPTIONS(MAIN) REORDER;

     DCL ABEND CHAR(50) INIT('THIS IS THE INITIAL ABEND MESSAGE TEXT');

     ON ERROR BEGIN;
              ON ERROR SYSTEM;
              PUT DATA;
              CALL PLIDUMP('TFSB',ABEND);
           END;

     DCL (LOW,SUBSTR,DATE,TIME,TRANSLATE) BUILTIN;

     DCL BDETAPE FILE RECORD INPUT;

     DCL EOF BIT(1) INIT('0');

     ON ENDFILE(BDETAPE) BEGIN;
                                 EOF = '1';
                               END;

     DCL P  POINTER;                          /* ALL INPUT STRUCTURES    */
                                                    /* BASED ON IT             */

     DCL K  POINTER;                          /* REC-ID BUILD-UP AREA    */

     DCL H  POINTER;                          /* LEVEL IN HEX            */

     DCL O  POINTER;                          /* 256-BYTE AREA           */

     DCL B  POINTER;                          /* 1024-BYTE AREA          */

     DCL PR POINTER;                          /* PRINT-LINE              */

     DCL S  POINTER;                          /* MH-SAVE AREA            */

     DCL BQBLIBI FILE RECORD OUTPUT SEQUENTIAL BUFFERED
                             ENVIRONMENT(VSAM);  /* IT'S A ESDS VSAM        */

     DCL JCLPARM CHAR(100) VARYING;           /* ONLY ACCEPTED FORMAT IS */
                                                    /* V..L.. WHERE .. = DIGIT */

     DCL VERSION CHAR(2) INIT('00');

     DCL LEVEL   CHAR(2) INIT('00');

     DCL 1 LEVELHEX BASED(H),
                 3 LEVELFILL1      BIT(8) UNALIGNED,
                 3 LEVELHEXONEBYTE BIT(8) UNALIGNED;

     DCL LEVELHEXHW BASED(H) BIN FIXED(15,0) UNALIGNED INIT(0);

     ALLOCATE LEVELHEXHW SET(H);
```

```
DCL 1 MH BASED (P),                    /* INPUT STRUCTURE        */
            3 MHID,                       /* FOR MESSAGE-HEADERS    */
              5 MHID01  BIT (16),         /* HEX '0281'             */
              5 MHID23  CHAR(02),         /* HEX '0000'             */
              5 MHID45  BIT (16),         /* PGM HEX '0000'         */
                                          /* PNL HEX '0404'         */
              5 MHFILL1 CHAR(24),         /* HEX NULL               */
            3 MHPID1    CHAR(04),         /* PGM/PNL-ID             */
            3 MHPID2    CHAR(04),         /* PGM/PNL-ID             */
            3 MHFILL2   CHAR(86),         /* HEX NULL               */
            3 MHCOUNT   CHAR(04);         /* EBCDIC COUNT OF 128 REC*/
                                          /* FOLLOWING THIS MH      */
                                          /* (USED TO CONTROL LOOP) */

DCL 1 PGM128IN BASED(P),               /* INPUT STRUCTURE        */
                                          /* FOR PGM-DATA-RECORDS   */
            3 PGMINP    CHAR(128);        /* 128 BYTES OF PGM-DATA  */

DCL 1 PGM112IN BASED (P),              /* INPUT STRUCTURE        */
            3 PGMINP112 CHAR(112),        /* 112 BYTES OF PGM-DATA  */
            3 PGMINPBLKCNT BIN FIXED(15,0) UNALIGNED;
                                          /* IN THE FIRST REC ONLY  */

DCL 1 PNL112IN BASED (P),              /* INPUT STRUCTURE        */
            3 PNLINP112 CHAR(112),        /* 112 BYTES OF PNL-DATA  */
            3 PNLINPFILL1  CHAR(001),     /*   1 BYTE  FILLER       */
            3 PNLINPHEXID BIN FIXED(15,0) UNALIGNED;

DCL 1 SMH BASED (S),                   /* STORE STRUCTURE        */
            3 SMHID,                      /* FOR THE MESSAGE-HEADER */
              5 SMHID01  BIT (16),        /* HEX '0281'             */
              5 SMHID23  CHAR(02),        /* HEX '0000'             */
              5 SMHID45  BIT (16),        /* PGM HEX '0000'         */
                                          /* PNL HEX '0404'         */
              5 SMHFILL1 CHAR(24),        /* HEX NULL               */
            3 SMHPID1    CHAR(04),        /* PGM/PNL-ID             */
            3 SMHPID2    CHAR(04),        /* PGM/PNL/ID             */
            3 SMHFILL2   CHAR(86),        /* HEX NULL               */
            3 SMHCOUNT   CHAR(04);        /* EBCDIC COUNT OF 128 REC*/
                                          /* FOLLOWING THIS  MH     */

ALLOCATE SMH SET (S);

DCL SMHSTRING CHAR(128) BASED (S);

DCL SMHCOUNTBIN BIN FIXED(15,0);       /* (USED TO CONTROL LOOP) */

DCL SMHPNLIDHEX BIN FIXED(15,0);       /* (USED FOR COMPARE    ) */
```

```
DCL 1 BQBKEY BASED (K),                  /*SEE MANUAL LY38-3036 */
            3 BQBKEYCC   CHAR(02) INIT('CC'), /*PAGE 8              */
            3 BQBKEYIN   CHAR(02) INIT('IN'), /* IN FACT, IT IS A    */
            3 BQBKEYFPPN CHAR(08),            /* REC-ID RATHER THAN  */
                                              /* A KEY, SINCE IT IS  */
                                              /* A ESDS              */
            3 BQBKEYV    BIN FIXED (15,0) UNALIGNED,
            3 BQBKEYFILL1 CHAR(1) INIT(LOW(1)),
            3 BQBKEYL    BIN FIXED (15,0) UNALIGNED,
            3 BQBKEYSEQ  BIN FIXED(15,0) UNALIGNED,
            3 BQBKEYFLAG,
              5 BQBKEYFLAG0 BIT(1) UNALIGNED,
              5 BQBKEYFLAG1 BIT(1) UNALIGNED,
              5 BQBKEYFILL  BIT(6) INIT('000000'B) UNALIGNED;

DCL 1 BQBKEYW BASED (K) CHAR(20);

DCL 1 BQBPRM,                                /* PARAMETER RECORD*/
            3 BQBPRMKEY     CHAR(20),              /* IN BQBLIBI      */
            3 BQBPRMTYPE    BIT(16),
            3 BQBPRMFILL1   CHAR(24) INIT(LOW(24)),
            3 BQBPRMPGPNID  CHAR(08),
            3 BQBPRMFILL2   CHAR(148) INIT(LOW(148)),
            3 BQBPRMCOUNT   BIN FIXED(15,0) UNALIGNED,
            3 BQBPRMFILL3   CHAR(4) INIT(LOW(4)),
            3 BQBPRMDATE    CHAR(8)
                            INIT(TRANSLATE('MO/DA/YE',DATE,'YEMODA')),
            3 BQBPRMFILL4   CHAR(56) INIT(LOW(56));

DCL 1 PGM256OUT BASED(O),            /* OUTPUT BUILD-UP AREA   */
            3 PGMFIRST128  CHAR(128),      /* FOR PROGRAM-RECORDS    */
            3 PGMBYTES128_240 CHAR(112),
            3 PGMBLKCNT BIN FIXED(15,0) UNALIGNED,
            3 PGMVERSION BIN FIXED(15,0) UNALIGNED,
            3 PGMLEVEL   BIT(8) UNALIGNED,
            3 PGMFILL1   CHAR(11) INIT(LOW(11));

DCL 1 PNL256OUT BASED(O),            /* OUTPUT BUILD-UP AREA   */
            3 PNLFIRST128  CHAR(128),      /* FPR PANEL-RECORDS      */
            3 PNLBYTES128_240 CHAR(112),
            3 PNLFILL1  CHAR (1),
            3 PNLIDHEX  BIN FIXED(15,0) UNALIGNED,
            3 PNLFILL2   CHAR(13);

DCL PGM256OUTSTRING BASED(O) CHAR(256);

ALLOCATE PGM256OUTSTRING SET(O);

/* THE FOLLOWING IS THE PROGRAM-CONTROLLED BLOCKING MECHANISM  */

DCL BLOCK1 CHAR(0276) BASED(B);      /*USED FOR PHYSICAL WRITE */

DCL BLOCK2 CHAR(0532) BASED(B);      /*USED FOR PHYSICAL WRITE */

DCL BLOCK3 CHAR(0788) BASED(B);      /*USED FOR PHYSICAL WRITE */

DCL BLOCK4 CHAR(1044) BASED(B);      /*USED FOR PHYSICAL WRITE */

DCL 1 BLOCKTAB BASED(B),
            3 BLOCKKEY     CHAR(20),
            3 TABBLOCK (4) CHAR(256);      /* USED TO LOGICAL WRITE IN*/
```

```
DCL I BIN FIXED(15,0);                  /* INDEX FOR BLOCKING     */

ALLOCATE BLOCK4 SET(B);

DCL CNTTOTBDE     BIN FIXED (15,0)      /* TOTAL BDE REC READ     */
                              INIT(-1);
      DCL CNTTOTBDEINV BIN FIXED (15,0)    /* TOTAL INVALID BDE REC  */
                              INIT(00);
      DCL CNTPGMBDE    BIN FIXED (15,0)    /* BDE REC READ PER PGM/PN*/
                              INIT(00);
      DCL CNTMHBDE     BIN FIXED (15,0)    /* TOTAL VALID MH REC READ*/
                              INIT(00);
      DCL CNTPGMBQB    BIN FIXED (15,0)    /* 240 REC WRITTEN  PGM/PN*/
                              INIT(00);
      DCL CNTTOTBQB    BIN FIXED (15,0)    /* 240 REC WRITTEN  TOTAL */
                              INIT(00);
      DCL CNTPRMBQB    BIN FIXED (15,0)    /* 272 PARM-REC WRITTEN   */
                              INIT(00);
      DCL CNTTOTBQBPH  BIN FIXED (15,0)    /* 1044 REC WRITTEN TOTAL */
                              INIT(00);
```

```
DCL MESSAGE CHAR(120) INIT('');

DCL 1 LINE BASED (PR),
            3 ASA           CHAR(1) INIT(' '),
            3 LINE120       CHAR(120) INIT('');
      ALLOCATE LINE  SET (PR);

DCL PCNTPGMBDE PIC'ZZZ9';
      DCL PCNTPGMBQB PIC'ZZZ9';

/* CHECKING OF JCL-PARM FOLLOWS :                                    */

IF   SUBSTR(JCLPARM,1,1) = 'V' &
          (SUBSTR(JCLPARM,2,1) >='0' & SUBSTR(JCLPARM,2,1) <='9') &
          (SUBSTR(JCLPARM,3,1) >='0' & SUBSTR(JCLPARM,3,1) <='9') &
           SUBSTR(JCLPARM,4,1) = 'L' &
          (SUBSTR(JCLPARM,5,1) >='0' & SUBSTR(JCLPARM,5,1) <='9') &
          (SUBSTR(JCLPARM,6,1) >='0' & SUBSTR(JCLPARM,6,1) <='9')
         THEN DO;
            VERSION = SUBSTR(JCLPARM,2,2);
            LEVEL   = SUBSTR(JCLPARM,5,2);
            LEVELHEXHW = LEVEL;
         END;
         ELSE DO;
            LINE120 = 'VERSION/LEVEL INVALID OR NOT ENTERED'||
                      ' THRU JCL-PARM. DEFAULTED TO 00/00';
            CALL PRINT(PR);
         END;
```

```
      READ FILE(BDETAPE) SET (P);     /* READ THE VERY FIRST TAPE REC */
           CNTTOTBDE = CNTTOTBDE +1;
           ALLOCATE BQBKEY SET (K);

MAINLOOP:
           DO WHILE (¬EOF);                /* AS LONG AS THERE ARE TAPE REC*/
              /* CHECK IF IT IS A MESSAGE-HEADER-RECORD FOR PROGRAMS      */
              IF MHID01  = '0000001010000001'B           /* HEX '0281'    */
               & MHID23  = LOW(2)
               & MHID45  = '0000000000000000'B
               & MHFILL1 = LOW(24)
               & MHPID1  = MHPID2                        /* PGMID TWICE   */
                                                         /* AS REQUIRED ? */
               & MHFILL2 = LOW(86)
                 THEN DO;                                /* IT IS A PGM-MH*/
                    SMH = MH;                            /* SAVE MH       */
                    ON CONVERSION BEGIN;
                       MESSAGE = 'INVALID BLOCK-COUNT IN MH.'||
                                 ' PROCESSING CONTINUES WITH NEXT MH';
                       GO TO PGMADMIT;
                    END;
                    SMHCOUNTBIN = SMHCOUNT;              /*AVOID CONVERSION*/
                    ON CONVERSION BEGIN;
                       ABEND = 'CONVERSION';
                       SIGNAL ERROR;
                    END;

              BQBKEYFPPN = 'INFP'||SMHPID1;         /*BUILD REC-ID   */
                    BQBKEYV    = VERSION;
                    BQBKEYL    = LEVEL;
                    BQBKEYSEQ  = 0;
                    BQBKEYFLAG0 = '1'B;                  /*NOT LAST RECORD*/
                    BQBKEYFLAG1 = '1'B;                  /*PARAMETER REC  */
                    BQBPRMTYPE  = '0000000000000000'B;   /* BUILD PARM REC*/
                    BQBPRMPGPNID = SMHPID1||SMHPID2;
                    BQBPRMCOUNT  = (SMHCOUNTBIN / 2);
                    IF SMHCOUNTBIN ¬= (BQBPRMCOUNT * 2)
                       THEN DO;
                          MESSAGE = 'BLOCK-COUNT IN MH NOT AN '||
                                    'EVEN NUMBER. PROCESSING '||
                                    'CONTINUES WITH NEXT MH';
                          GO TO PGMADMIT;
                       END;
                    BQBPRMKEY = BQBKEYW;

              WRITE FILE(BQBLIBI) FROM(BQBPRM);     /* WRITE PARM-REC*/
                    CNTMHBDE  = CNTMHBDE +1;
                    CNTPRMBQB = CNTPRMBQB +1;
                    READ FILE(BDETAPE) SET (P);      /* READ 1ST DATA REC */
                    CNTTOTBDE = CNTTOTBDE +1;
                    CNTPGMBDE             =1;
```

```
/* START BUILD-UP OF FIRST 1044-BYTE BLOCK            */

 BQBKEYSEQ = BQBKEYSEQ +1;    /* BUILD BLOCK-ID        */
       BQBKEYFLAG1 = '0'B;

 PGMFIRST128 = PGMINP;        /*BUILD FIRST HALF OF REC*/
       READ FILE(BDETAPE) SET (P);      /* READ 2ND DATA REC */
       CNTTOTBDE = CNTTOTBDE +1;
       CNTPGMBDE = CNTPGMBDE +1;
       PGMBYTES128_240 = PGMINP112; /*BUILD 2ND HALF OF REC */
       PGMBLKCNT = PGMINPBLKCNT;
       IF PGMBLKCNT ¬= BQBPRMCOUNT
          THEN DO;
             MESSAGE = 'MH-COUNT AND FIRST BLOCK COUNT'||
                       ' DONT MATCH';
             GO TO PGMADMIT;
          END;
       PGMVERSION = VERSION;
       PGMLEVEL = LEVELHEXONEBYTE;
       I = 1;

 TABBLOCK(I) = PGM256OUTSTRING;/* LOGICAL WRITE 1. REC */
       CNTPGMBQB              =1;     /* OF BLOCK            */

 READ FILE(BDETAPE) SET (P);   /* READ NEXT DATA REC   */
       CNTTOTBDE = CNTTOTBDE +1;
       CNTPGMBDE = CNTPGMBDE +1;
       DO WHILE(¬ EOF & (CNTPGMBDE ¬> SMHCOUNTBIN) & (I < 4));
         /* AS LONG AS THE REC PERTAINS TO THE SAME PROGRAM  */
         /* AND THE OUTPUT-BLOCK ISN'T FILLED YET            */
          PGMFIRST128 = PGMINP;     /*BUILD FIRST HALF OF REC*/
          READ FILE(BDETAPE) SET (P); /*READ NEXT DATA REC   */
          CNTTOTBDE = CNTTOTBDE +1;
          CNTPGMBDE = CNTPGMBDE +1;
          PGMBYTES128_240 = PGMINP112; /*BUILD 2ND HALF */
          PGMBLKCNT = 0;
          PGMVERSION = 0;
          PGMLEVEL = '00000000'B;
          I = I+1;

    TABBLOCK(I) = PGM256OUTSTRING;/* WRITE LOGICAL */
          CNTPGMBQB = CNTPGMBQB +1;

    READ FILE(BDETAPE) SET (P); /* READ NEXT DATAREC */
          CNTTOTBDE = CNTTOTBDE +1;
          CNTPGMBDE = CNTPGMBDE +1;
       END;
       IF (CNTPGMBDE > SMHCOUNTBIN) | EOF
          THEN DO;                    /* LAST REC OF THIS PGM*/
             BQBKEYFLAG0 = '0'B;
          END;
       BLOCKKEY = BQBKEYW;
       SELECT(I);                     /* WRITE PHYSICAL BLOCK */
          WHEN(1) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK1);
          END;
          WHEN(2) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK2);
          END;
          WHEN(3) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK3);
```

```
            END;
            WHEN(4) DO;
               WRITE FILE(BQBLIBI) FROM(BLOCK4);
            END;
         END;
         CNTTOTBQBPH = CNTTOTBQBPH +1;

/* END OF FIRST 1044-BYTE BLOCK                                    */
```

```
/* START FURTHER 1044-BYTE BLOCKS                          */

 DO WHILE(¬ EOF & (CNTPGMBDE ¬> SMHCOUNTBIN));
         /* AS LONG AS THE REC PERTAINS TO THE SAME PROGRAM  */

    BQBKEYSEQ = BQBKEYSEQ +1; /*BUILD BLOCK-ID          */
          I = 0;
          DO WHILE(¬EOF & (CNTPGMBDE ¬> SMHCOUNTBIN) & (I < 4));
           /* AS LONG AS THE REC PERTAINS TO THE SAME PROGRAM*/
           /* AND THE OUTPUT-BLOCK ISN'T FILLED YET          */
             PGMFIRST128 = PGMINP; /*BUILD FIRST HALF OF REC*/
             READ FILE(BDETAPE) SET (P); /* READ NEXT DATA REC*/
             CNTTOTBDE = CNTTOTBDE +1;
             CNTPGMBDE = CNTPGMBDE +1;
             PGMBYTES128_240 = PGMINP112; /*BUILD 2ND HALF */
             PGMBLKCNT = 0;
             PGMVERSION = 0;
             PGMLEVEL = '00000000'B;
             I = I+1;

       TABBLOCK(I) = PGM256OUTSTRING;/* WRITE LOGICAL */
             CNTPGMBQB = CNTPGMBQB +1;

       READ FILE(BDETAPE) SET (P); /* READ NEXT DATA REC*/
             CNTTOTBDE = CNTTOTBDE +1;
             CNTPGMBDE = CNTPGMBDE +1;
          END;
          IF (CNTPGMBDE > SMHCOUNTBIN) | EOF
             THEN DO;                    /* LAST REC OF THIS PGM*/
                BQBKEYFLAG0 = '0'B;
             END;
          BLOCKKEY = BQBKEYW;
          SELECT(I);                     /* WRITE PHYSICAL BLOCK */
             WHEN(1) DO;
              WRITE FILE(BQBLIBI) FROM(BLOCK1);
             END;
             WHEN(2) DO;
              WRITE FILE(BQBLIBI) FROM(BLOCK2);
             END;
             WHEN(3) DO;
              WRITE FILE(BQBLIBI) FROM(BLOCK3);
             END;
             WHEN(4) DO;
              WRITE FILE(BQBLIBI) FROM(BLOCK4);
             END;
          END;
          CNTTOTBQBPH = CNTTOTBQBPH +1;
       END;

/* END FURTHER 1044-BYTES BLOCKS */
```

```
LINE120 = 'RECORD TOTALS FOR PGM '||SMHPID1||' :';
      ASA = '0';
      CALL PRINT(PR);
      PCNTPGMBDE = CNTPGMBDE;
      PCNTPGMBQB = CNTPGMBQB;
      LINE120 = PCNTPGMBDE||' BDE-TAPE REC READ (128 BYTES)'||
       '  AND '||PCNTPGMBQB||' LOGICAL BQBLIBI REC WRITTEN '||
       '(240 BYTES)';
      CALL PRINT(PR);
      IF CNTPGMBQB ¬= (SMHCOUNTBIN / 2)
         THEN DO;      /* NUMBER OF 128-BYTE-REC IN MH HAS TO*/
                       /* MATCH ACTUAL NUMBER OF RECORDS     */
            LINE120 = '********** SOMETHING WENT WRONG *****';
            ASA     ='-';
            CALL PRINT(PR);
            PCNTPGMBDE = SMHCOUNTBIN;
            PCNTPGMBQB = CNTPGMBQB;
            LINE120 = 'SECTOR COUNT IN INPUT MH :'||PCNTPGMBDE
            ||'         RECORD COUNT IN OUTPUT   :'||PCNTPGMBQB;
            CALL PRINT(PR);
            LINE120 = 'PROCESSING   U N S U C C E S S F U L';
            CALL PRINT(PR);
            /* BACKOUT GOES HERE IF APPROPRIATE   **********/
         END;
         ELSE DO;
            LINE120 = 'PROCESSING FOR PGM '||SMHPID1||
                      ' SUCESSFULLY COMPLETED';
            CALL PRINT(PR);
            CNTTOTBQB = CNTTOTBQB + CNTPGMBQB;
         END;
```

```
            /* ADMIT SECTION                    */

             GO TO PGMADMITEND;                    /* SKIP AROUND IT */

PGMADMIT:    DO;      /* INVOKED IF INPUT ERRORS DETECTED            */
                      LINE120 = '************************************************';
                      ASA = '0';
                      CALL PRINT(PR);
                      LINE120 = MESSAGE;
                      CALL PRINT(PR);
                      LINE120 = 'INPUT MH(B 1-64): '||SUBSTR(SMHSTRING,1,64);
                      CALL PRINT(PR);
                      LINE120 ='      (B 65-128): '||SUBSTR(SMHSTRING,65,64);
                      CALL PRINT(PR);
                      LINE120 = '20 BYTES BQBLIBIKEY: '||BQBKEYW||
                                ' USED VERSION: '||VERSION||
                                ' USED LEVEL  : '||LEVEL;
                      CALL PRINT(PR);
                      LINE120 = '************************************************';
                      CALL PRINT(PR);
                      READ FILE(BDETAPE) SET (P);
                      CNTTOTBDEINV = CNTTOTBDEINV +1;

PGMADMITEND: END;
                    END;                          /* END IF-THEN  PGM-MH OK*/
```

```
ELSE DO;                                    /* SEE IF PANEL MH */
        /* CHECK IF IT IS A MESSAGE-HEADER-RECORD FOR PANELS */
        IF MHID01 = '0000001010000001'B      /* HEX '0281'    */
         & MHID23 = LOW(2)
         & MHID45 = '0000010000000100'B      /* HEX '0404'    */
         & MHFILL1 = LOW(24)
         & MHPID1 = MHPID2                   /* PNLID TWICE ? */
         & MHFILL2 = LOW(86)
           THEN DO;                          /* IT IS A PNL-MH*/
              SMH = MH;                      /* SAVE MH       */
              ON CONVERSION BEGIN;
                 MESSAGE = 'INVALID BLOCK-COUNT IN MH.'||
                           ' PROCESSING CONTINUES WITH NEXT MH';
                 GO TO PNLADMIT;
              END;
              SMHCOUNTBIN = SMHCOUNT;        /*AVOID CONVERSION*/
              ON CONVERSION BEGIN;
                 MESSAGE = 'INVALID PANEL-IDUNT IN MH.'||
                           ' PROCESSING CONTINUES WITH NEXT MH';
                 GO TO PNLADMIT;
              END;
              SMHPNLIDHEX = SMHPID1;         /*FOR COMPARE    */
              ON CONVERSION BEGIN;
                 ABEND = 'CONVERSION';
                 SIGNAL ERROR;
              END;

        BQBKEYFPPN = 'INPN'||SMHPID1;  /*BUILD REC-ID   */
              BQBKEYV = VERSION;
              BQBKEYL = LEVEL;
              BQBKEYSEQ = 0;
              BQBKEYFLAG0 = '1'B;            /*NOT LAST RECORD*/
              BQBKEYFLAG1 = '1'B;            /*PARAMETER REC  */

        BQBPRMTYPE = '0000010000000100'B;/*BUILD PARM REC*/
              BQBPRMPGPNID = SMHPID1||SMHPID2;
              BQBPRMCOUNT = (SMHCOUNTBIN / 2);
              IF SMHCOUNTBIN ¬= (BQBPRMCOUNT * 2)
                 THEN DO;
                    MESSAGE = 'BLOCK-COUNT IN MH NOT AN '||
                              'EVEN NUMBER. PROCESSING '||
                              'CONTINUES WITH NEXT MH';
                    GO TO PNLADMIT;
                 END;
              BQBPRMKEY = BQBKEYW;

        WRITE FILE(BQBLIBI) FROM(BQBPRM);    /* PARM-REC */
              CNTPRMBQB = CNTPRMBQB +1;
              CNTMHBDE = CNTMHBDE +1;
              READ FILE(BDETAPE) SET (P);
              CNTTOTBDE = CNTTOTBDE +1;
              CNTPGMBDE             =1;
```

```
/* START FIRST 1044-BYTE-BLOCK */

 BQBKEYSEQ = BQBKEYSEQ +1; /*BUILD REC-ID       */
       BQBKEYFLAG1 = '0'B;

 PNLFIRST128 = PGMINP; /*BUILD FIRST HALF OF REC*/
       READ FILE(BDETAPE) SET (P);
       CNTTOTBDE = CNTTOTBDE +1;
       CNTPGMBDE = CNTPGMBDE +1;
       IF PNLINPHEXID  ¬= SMHPNLIDHEX
          THEN DO;
             MESSAGE = 'MH-PANELID AND FIRST-BLOCK ID '||
                       ' DONT MATCH';
             GO TO PNLADMIT;
          END;
       PNLBYTES128_240 = PGMINP112; /*BUILD 2ND HALF REC*/
                                    /* INCL DATA LENGTH */
       PNLFILL1  = LOW(1);
       PNLIDHEX  = PNLINPHEXID;
       PNLFILL2  = LOW(13);
       I = 1;

 TABBLOCK(I) = PGM256OUTSTRING;/*WRITE LOGICAL 1ST*/
       CNTPGMBQB             =1;

 READ FILE(BDETAPE) SET (P);
       CNTTOTBDE = CNTTOTBDE +1;
       CNTPGMBDE = CNTPGMBDE +1;
       DO WHILE(¬ EOF & (CNTPGMBDE ¬> SMHCOUNTBIN)
                      & (I < 4));
          PNLFIRST128 = PGMINP;  /*BUILD 1ST HALF OF REC*/
          READ FILE(BDETAPE) SET (P);
          CNTTOTBDE = CNTTOTBDE +1;
          CNTPGMBDE = CNTPGMBDE +1;
          PNLBYTES128_240 = PGMINP112; /*BUILD 2ND HALF */
          PNLIDHEX  = 0;
          I = I+1;

    TABBLOCK(I) = PGM256OUTSTRING;/* WRITE LOGICAL*/
          CNTPGMBQB = CNTPGMBQB +1;

    READ FILE(BDETAPE) SET (P);
          CNTTOTBDE = CNTTOTBDE +1;
          CNTPGMBDE = CNTPGMBDE +1;
       END;
       IF (CNTPGMBDE > SMHCOUNTBIN) | EOF
          THEN DO;              /* LAST REC OF THIS PNL*/
             BQBKEYFLAG0 = '0'B;
          END;
       BLOCKKEY = BQBKEYW;
       SELECT(I);              /* WRITE PHYSICAL BLOCK */
          WHEN(1) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK1);
          END;
          WHEN(2) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK2);
          END;
          WHEN(3) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK3);
          END;
          WHEN(4) DO;
```

```
            WRITE FILE(BQBLIBI) FROM(BLOCK4);
         END;
      END;
      CNTTOTBQBPH = CNTTOTBQBPH +1;

/* END FIRST 1044-BYTE-BLOCK */
```

```
/* START FURTHER 1044-BYTE BLOCKS */

 DO WHILE(¬ EOF & (CNTPGMBDE ¬> SMHCOUNTBIN));
       /* AS LONG AS IT IS THE SAME PANEL            */

    BQBKEYSEQ = BQBKEYSEQ +1; /*BUILD REC-ID    */
          I = 0;
          DO WHILE(¬EOF & (CNTPGMBDE ¬> SMHCOUNTBIN)
                        & (I < 4));
           /* AS LONG AS IT IS THE SAME PANEL         */
           /* AND FITS IN THE SAME BLOCK              */
             PNLFIRST128 = PGMINP;
                                /*BUILD FIRST HALF OF REC*/
             READ FILE(BDETAPE) SET (P);
             CNTTOTBDE = CNTTOTBDE +1;
             CNTPGMBDE = CNTPGMBDE +1;
             PNLBYTES128_240 = PGMINP112;
                                          /*BUILD 2ND HALF */
             PNLIDHEX  = 0;
             I = I+1;

       TABBLOCK(I) = PGM256OUTSTRING;

                                  /* WRITE LOGICAL */
          CNTPGMBQB = CNTPGMBQB +1;

       READ FILE(BDETAPE) SET (P);
          CNTTOTBDE = CNTTOTBDE +1;
          CNTPGMBDE = CNTPGMBDE +1;
       END;
       IF (CNTPGMBDE > SMHCOUNTBIN) | EOF
          THEN DO;          /* LAST REC OF THIS PNL*/
             BQBKEYFLAG0 = '0'B;
          END;
       BLOCKKEY = BQBKEYW;
       SELECT(I);          /* WRITE PHYSICAL BLOCK */
          WHEN(1) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK1);
          END;
          WHEN(2) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK2);
          END;
          WHEN(3) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK3);
          END;
          WHEN(4) DO;
             WRITE FILE(BQBLIBI) FROM(BLOCK4);
          END;
       END;
       CNTTOTBQBPH = CNTTOTBQBPH +1;
    END;

/* END FURTHER 1044-BYTE-BLOCKS */
```

```
LINE120 = 'RECORD TOTALS FOR PNL '||SMHPID1||' :';
      ASA = '0';
      CALL PRINT(PR);
      PCNTPGMBDE = CNTPGMBDE;
      PCNTPGMBQB = CNTPGMBQB;
      LINE120 = PCNTPGMBDE||' BDE-TAPE R READ (128 B)'||
      '  AND '||PCNTPGMBQB||' LOGICAL BQBLIBI REC '||
      ' WRITTEN (240 BYTES)';
      CALL PRINT(PR);
      IF CNTPGMBQB ¬= (SMHCOUNTBIN / 2)
         THEN DO;
            LINE120 = '****** SOMETHING WENT WRONG ****';
            ASA ='-';
            CALL PRINT(PR);
            PCNTPGMBDE = SMHCOUNTBIN;
            PCNTPGMBQB = CNTPGMBQB;
            LINE120 = 'SEC CNT IN INPUT MH :'||PCNTPGMBDE
            ||'       REC CNT IN OUTPUT   :'||PCNTPGMBQB;
            CALL PRINT(PR);
            LINE120 = 'PROCESS  U N S U C C E S S F U L';
            CALL PRINT(PR);
            /* HERE SOME BACKOUT SHOULD BE DONE *********/
         END;
         ELSE DO;
            LINE120 = 'PROCESSING FOR PNL '||SMHPID1||
                      ' SUCESSFULLY COMPLETED';
            CALL PRINT(PR);
            CNTTOTBQB = CNTTOTBQB + CNTPGMBQB;
         END;
```

```
                /* ADMIT SECTION                */

                 GO TO PNLADMITEND;              /* SKIP AROUND IT */

PNLADMIT:        DO;        /*INVOKED IF INPUT ERRORS DETECTED   */
                          LINE120 = '*****************************************';
                          ASA = '0';
                          CALL PRINT(PR);
                          LINE120 = MESSAGE;
                          CALL PRINT(PR);
                          LINE120 = 'INPUT MH(B 1-64): '||
                                                SUBSTR(SMHSTRING,1,64);
                          CALL PRINT(PR);
                          LINE120 = '       (B 65-128): '||
                                               SUBSTR(SMHSTRING,65,64);
                          CALL PRINT(PR);
                          LINE120 = '20 BYTES BQBLIBIKEY: '||BQBKEYW||
                                    ' USED VERSION: '||VERSION||
                                    ' USED LEVEL  : '||LEVEL;
                          CALL PRINT(PR);
                          LINE120 = '*****************************************';
                          CALL PRINT(PR);
                          READ FILE(BDETAPE) SET (P);
                          CNTTOTBDEINV = CNTTOTBDEINV +1;

PNLADMITEND:     END;
                     END;                          /* END IF-THEN  PNL-MH OK*/
                     ELSE DO;                      /* INVALID INPUT (MH)    */
                        LINE120 = 'INVALID INPUT READ WHERE PGM OR'||
                                  ' PNL MH EXPECTED **********************';
                        CALL PRINT(PR);
                        LINE120 = 'INPUT (B 1-64): '||SUBSTR(PGMINP,1,64);
                        CALL PRINT(PR);
                        LINE120 ='    (B 65-128): '||SUBSTR(PGMINP,65,64);
                        CALL PRINT(PR);
                        LINE120 = 'READING CONTINUES UNTIL VALID'||
                                  ' MH FOUND OR EOF REACHED';
                        CALL PRINT(PR);
                        READ FILE(BDETAPE) SET (P);
                        CNTTOTBDEINV = CNTTOTBDEINV +1;
                     END;                          /* END INVALID INPUT     */
                END;                               /* END IF-ELSE  PGM-MH OK*/

    END MAINLOOP;
```

```
LINE120 = ' T O T A L S  FOR THIS RUN   :';
      ASA = '1';
      CALL PRINT(PR);
      PCNTPGMBDE = CNTTOTBDE;
      PCNTPGMBQB = CNTTOTBQB;
      LINE120 = PCNTPGMBDE||' TOTAL BDE-TAPE REC READ (128 BYTES)'||
       '  AND '||PCNTPGMBQB||' TOTAL LOGICAL BQBLIBI REC WRITTEN '||
       '(240 BYTES)';
      ASA = '0';
      CALL PRINT(PR);
      PCNTPGMBDE = CNTMHBDE;
      PCNTPGMBQB = CNTTOTBQBPH;
      LINE120 = PCNTPGMBDE||' TOTAL VALID MH REC READ'||
       '  AND '||PCNTPGMBQB||' TOTAL PHYSICAL BQBLIBI REC WRITTEN '||
       '(1044 BYTES)';
      ASA = '0';
      CALL PRINT(PR);
      PCNTPGMBQB = CNTPRMBQB;
      LINE120 = PCNTPGMBQB||' TOTAL BQBLIBI PARM REC WRITTEN (272 B)';
      ASA = '0';
      CALL PRINT(PR);
      PCNTPGMBDE = CNTTOTBDEINV;
      LINE120 = PCNTPGMBDE||' TOTAL   I N V A L I D   MH REC READ';
      ASA = '0';
      CALL PRINT(PR);
```

```
PRINT: PROC(POINTPR) REORDER;

          /*************************************************************************
          *                                                                        *
          *   THIS ROUTINE DOES ALL THE PRINTING WORK FOR THE FILE                 *
          *   CALLED *LISTING*. EACH TIME IT IS INVOKED, A 121 CHAR                *
          *   PRINT LINE (INCLUDING ASA CHARACTER) IS PASSED AS A                  *
          *   PARAMETER. IT KEEPS CONTROL OF OVERFLOW ETC.                         *
          *                                                                        *
          *************************************************************************/

     DCL LISTING FILE RECORD OUTPUT;

     DCL 1 OVFLL1 STATIC,
             3 OVFLASA     CHAR(1) INIT('1'),
             3 OVFLL1TXT   CHAR(112),
             3 OVFL1TXTP   CHAR(5) INIT('PAGE'),
             3 OVFL1PGE    PIC'ZZ9' INIT ('000');
           DCL 1 OVFLL2 STATIC CHAR(121) INIT('-');
           DCL LINECNT BIN FIXED(15,0) STATIC INIT(999);
           DCL POINTPR POINTER;
           DCL PRINTLINE BASED (POINTPR) CHAR(121);

     IF (LINECNT > 55) | (SUBSTR(PRINTLINE,1,1) = '1')
             THEN DO;
                OVFLL1TXT = 'BDE-FORMATTED TAPE TO HOST'||
                            '-BQBLIBI                   '||
                            TRANSLATE('MO/DA/YE',DATE,'YEMODA')||
                            '  '                        ||
                            TRANSLATE('HO.MI.SE',TIME,'HOMISE');
                OVFL1PGE = OVFL1PGE +1;
                LINECNT  = 3;
                WRITE FILE(LISTING) FROM (OVFLL1);
                WRITE FILE(LISTING) FROM (OVFLL2);
                SUBSTR(PRINTLINE,1,1) = ' ';
             END;

     WRITE FILE(LISTING) FROM (PRINTLINE);
           LINECNT = LINECNT +1;
           PRINTLINE = '';

END PRINT;

END BDEBQB;
```

C.1.2 RUNNING THE SAMPLE PROGRAM

The OS/VS JCL in Figure 66 on page 157 illustrates the DD statements needed to run the sample program. The example will read a tape, but the input could be direct from diskette if the host system has a diskette reader.

```
//BDEBQB   JOB BDEBQB,'BUILD BQBLIBI'
//GO      EXEC PGM=BDEBQB,PARM='ISASIZE(07K),R/V55L66'
//STEPLIB  DD  DSN=DPCX.BDE.PGM,DISP=SHR
//BDETAPE  DD  DSN=PNLBCKUP,VOL=SER=UKATST,UNIT=TAPE,
//             DCB=(RECFM=FB,LRECL=128,BLKSIZE=2048,DEN=3)
//BQBLIBI  DD  DSN=BQBLIBI,DISP=OLD
//LISTING  DD  SYSOUT=A
//SYSPRINT DD  SYSOUT=A
//PLIDUMP  DD  SYSOUT=A
//
```

Figure 66. JCL to run sample PL/1 program to build BQBLIBI: JCL statements such as these would be used to run the sample program which builds BQBLIBI on an OS/VS system.

C.2 PROGRAM FOR HOST-INITIATED SUBTASK

The program which is listed below may be initiated as a subtask by the host and will issue DPCX commands. Most commands will need to be contained in PROCs to allow their invocation from user number 251 under which the sub-task runs. A diagram illustrating the program flow may be found in Figure 67 on page 159.

In the picture, and in the source listing below, the copy of the program running as the host initiated subtask is designated as mode 'AAAA'. The commands are sent as messages from the host to a reserved user number. The host-initiated subtask initiates a second copy of itself running under the reserved user number. This is labelled mode 'BBBB' in Figure 67 on page 159 and in the program listing, and it retrieves the messages from the host and passes them in 4-byte notes to the original subtask initiated by the host (AAAA). This in turn initiates the SYSCTL3 System Service as a subtask as described in "Controlling the Execution Sequence" in the DPCX Programming: Guide to Program Structure manual.

The indication of success or failure returned by SYSCTL3 is reflected by the subtask (AAAA) in a message to the host, as are any errors encountered and messages issued by the invoked PROC or command.

The program may also be invoked from a PROC in an *CALL statement. When used in this manner, the program will write the passed data as a message to the host. It may be used exactly where an *CO statement would be used to inform a terminal operator. The copy of the program performing this function is referred to as mode 'CCCC' in Figure 67 on page 159 and the program listing.

Before deciding to use this technique, you should also read "Maintaining the Message Dataset" on page 168 for a precaution which will need to be taken to ensure that the message dataset does not start to fill with data which will not be retrieved.

Note: The use of this program with Host Prep SYSINFOREF gives the user access only to a subset of the function available to the user of Host Command Facility (HCF), and this method is not a generally viable alternative to the use of that product.

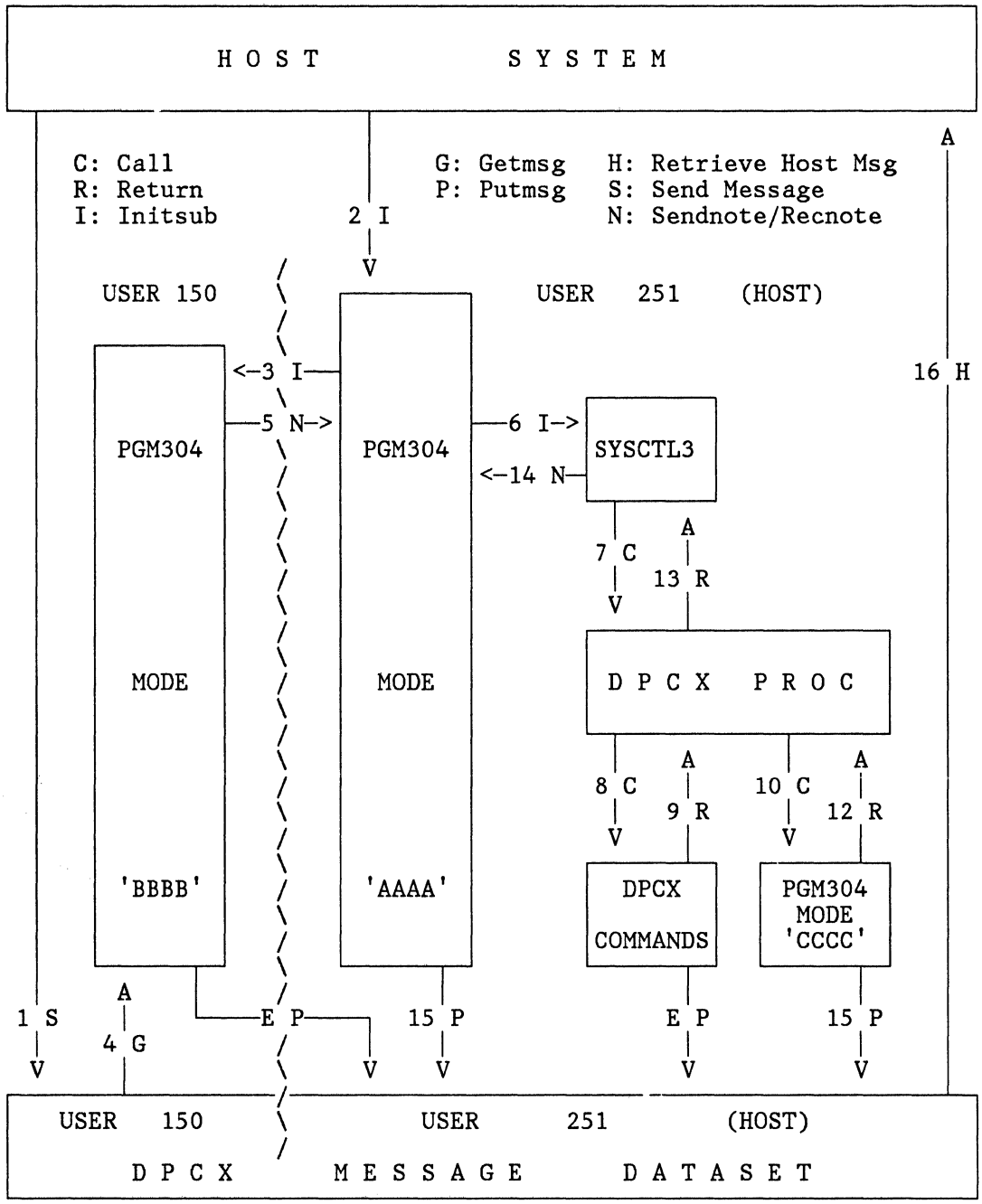


Figure 67. Schematic of the operation of sample program 304: how its different functions work to allow SYSINFOREF to execute commands at DPCX. Numbers depict the sequence of execution; 'E' an error flow.

C.2.1 SOURCE OF SAMPLE PROGRAM.

```
*********************************************************************************
*                                                                               *
*               ASSEMBLER VARIABLES                                             *
*                                                                               *
*********************************************************************************
         LCLA  &USER,&PGMID
         LCLA  &DEST1,&COMM1D,&COMM1E,&COMM1T,&DEST2,&COMM2
*
* *** REDEFINE THE FOLLOWING TO SUIT YOUR INSTALLATION'S CONVENTIONS
*
&USER    SETA  150        RESERVED USER NUMBER TO RECEIVE HOST MESSAGES
&PGMIL   SETA  304        PROGRAM NUMBER
*
&DEST1   SETA  81         FOR INTER-TASK COMMUNICATION BETWEEN
&COMM1D  SETA  91         THE TWO COPIES OF THE PROGRAM WHICH
&COMM1E  SETA  92         ARE RESPECTIVELY THE HOST-INITIATED
&COMM1T  SETA  93         AND MESSAGE RETRIEVAL SUBTASKS
*
&DEST2   SETA  82         FOR INTER-TASK COMMUNICATION BETWEEN THE
&COMM2   SETA  94         HOST INITIATED COPY OF THIS PROGRAM
*                         AND THE SYSCTL3 SUBTASK IT CREATES
```

```
*****************************************************************************
*                                                                           *
*              PROGRAM START, DATA AREAS AND FUNCTION DETERMINATION         *
*                                                                           *
*****************************************************************************
          STARTPGM PGMID=&PGMID,ENVIR=NOPRI,NAME='CALLCMD',OPSEL=NO,     *
                PRI=NONE
          SPACE 2
          DEFINE DATA
          DEFINE BUF=1                    HEADER FOR SYSTEM SERVICE
HDR       FLD    LVL=2,LNG=40
SMN       FLD    LVL=3,LNG=4,TYPE=AN
PGMID     FLD    LVL=3,LNG=4,TYPE=AN
OPID      FLD    LVL=3,LNG=3,TYPE=AN
TERMCD    FLD    LVL=3,LNG=1,TYPE=X
RESERV1   FLD    LVL=3,LNG=6
DESTID    FLD    LVL=3,LNG=3,TYPE=N
COMMID    FLD    LVL=3,LNG=3,TYPE=N
RESERV2   FLD    LVL=3,LNG=16
CTL3      FLD    LVL=2,LNG=4,TYPE=E,OCCURS=20
          DEFINE END
          SPACE 2
*
*               FIRST SEE IF WE HAVE BEEN CALLED FROM A PROC TO RETURN
*               A MESSAGE TO THE HOST, IN WHICH CASE BUFFER1 WILL BE
*               SET UP AND WILL HAVE '*CALL' AT POSITION 40.
*
          SRC    BUF(1),L=40,LNG=5
          COMP   '*CALL'
          GOTOC  EQ,ENQMSG
*
*               NOW DETERMINE WHETHER THE SUBTASK IS RUNNING UNDER THE
*               MSG USERID - IT WON'T BE WHEN THE HOST INITIATES IT.
*
          SRC    '&USER'
          DEST   REG(3)
          SRC    USERNO
          DEST   REG(2)
          COMP   REG(2),REG(3)
          GOTOC  EQ,COPY2
```

```
*****************************************************************************
*                                                                           *
*               PROCESSING FOR MODE 'AAAA'                                  *
*                                                                           *
*****************************************************************************
*
*              RE-INITIATE THE SUBTASK, THIS TIME FOR THE MSG USER.
*
         SRC   PGMID
         DEST  REG(2)
         INITSUB REG(2),USERNO='&USER'
         GOTOIF I(39),ON,NORESORC
*
*              LOOP ROUND ASSEMBLING MESSAGES.
*
LOOP     FREE  BUF(1)              CLEAN OUT BUFFER
         ZERO  REG(5)
*
*              WAIT FOR A BIT OF DATA.
*
LOOP1    ZERO  REG(6)
         INCR  REG(5)
         RECNOTE DESTID='&DEST1',TYPE=WAIT,COMMID=REG(6)
         GOTOIF I(38),ON,NODISK
         GOTOC X'01',RECERR
         GOTOC X'02',NODATA
         GOTOC X'04',DATA
         GOTO  RECERR
*
*              NOTE RECEIVED WITH DATA.
*
DATA     DEST  FLD=CTL3,I=REG(5)  PUT IT IN THE RIGHT PLACE
         SRC   REG(6)
         COMP  '&COMM1D'          IS IT EXPECTED?
         GOTOC NE,RECERR
         GOTO  LOOP1
*
*              NOTE RECEIVED WITH NO DATA - SHOULD BE END OF
*              MESSAGE OR END OF TASK.
*
NODATA   SRC   REG(6)
         COMP  '&COMM1T'          IS IT NOTIFICATION OF TERMINATION?
         GOTOC EQ,DONE
         COMP  '&COMM1E'          IS IT END OF MESSAGE?
         GOTOC NE,RECERR
*
*              BUILD THE SYSTEM SERVICE HEADER. THE MESSAGE RECEIVED
*              FROM THE HOST IS PASSED AS THE SYSCTL3 PARAMETER.
*
         SRC   '&DEST2'
         EDT   LNG=3,JS=R,FILL='0'
         DEST  FLD=DESTID
         SRC   '&COMM2'
         EDT   LNG=3,JS=R,FILL='0'
         DEST  FLD=COMMID
         SETMASK OFF,FLD=TERMCD,MASKVAL=11111111
         SETMASK ON,FLD=TERMCD,MASKVAL=00100000
*
*              INITIATE SYSCTL3 AS A SUBTASK.
*
         INITSUB PGMID=2616
```

```
         GOTOIF I(39),ON,NORSRC1
*
*              WAIT FOR NOTIFICATION OF COMPLETION.
*
         RECNOTE DESTID='&DEST2',TYPE=WAIT,COMMID='&COMM2'
         GOTOIF I(38),ON,NODISK
         GOTOC X'03',RECERR
*
*              BUILD STATUS MESSAGE FOR HOST.
*
         DEST  BUF(1),L=18,LNG=4
         COMP  'A00B'
         GOTOC EQ,GOOD
         COMP  '90F0'
         GOTOC EQ,BAD
         SRC   '      '
         GOTO JOIN
GOOD     SRC   ' GOOD'
         GOTO JOIN
BAD      SRC   ' BAD '
JOIN     DEST  BUF(1),L=22,LNG=5
         SRC   ' STATUS FROM '
         DEST  BUF(1),L=27,LNG=13
         SRC   '*** '
         DEST  BUF(1),L=14,LNG=4
PUTMSG   CALLSUB HOSTMSG
         GOTO  LOOP
*
*              NO MORE MESSAGES FROM HOST FROM COMPANION TASK.
*
DONE     EXIT  ,                    ALL DONE
```

```
*****************************************************************************
*                                                                           *
*               PROCESSING FOR MODE 'BBBB'                                  *
*                                                                           *
*****************************************************************************
*
*              GET A MESSAGE, UNKEYED, FOR THIS USER AND PROGRAM.
*
COPY2     ZERO  REG(4)              COUNTER OF HOST MESSAGES PROCESSED
LOOP2     GETMSG THISPGM
          GOTOIF I(38),ON,DONE2
          INCR  REG(4)              COUNT THIS MESSAGE
          ZERO  REG(5)
*
*              SEND MSG IN 4-BYTE NOTES TO THE HOST-INITIATED SUBTASK.
*
          CYCLE 20,ENDC
          INCR  REG(5)
          SRC   BUF(1),L=0,LNG=4,I=REG(5)
          SENDNOTE DESTID='&DEST1',DATA=YES,COMMID='&COMM1D'
          GOTOIF I(38),ON,NODISK2
          ENDCYCLE
*
*              INDICATE END OF MESSAGE.
*
ENDC      SENDNOTE DESTID='&DEST1',DATA=NO,COMMID='&COMM1E'
          GOTO  LOOP2
*
*              NOTIFY THE HOST INITIATED SUBTASK OF TERMINATION.
*
DONE2     SENDNOTE DESTID='&DEST1',DATA=NO,COMMID='&COMM1T'
          GOTOIF I(38),ON,NODISK2
EXIT2     EQU   *
          SRC   REG(4)              DID WE GET ANY MESSAGES FROM HOST?
          COMP  '0'
          GOTOC EQ,NOTHING
          EXIT  ,                   ALL DONE
```

```
*****************************************************************************
*                                                                           *
*               PROCESSING FOR MODE 'CCCC'                                  *
*                                                                           *
*****************************************************************************
*
*               ENQUEUE REQUESTOR'S MESSAGE TO HOST.
*
ENQMSG    SRC   '256'
          DEST  REG(1)
          MOVEBUF FROM=BUF(1),TO=BUF(2),LNG=REG(1) SAVE ENTIRE BUFFER
          SRC   '120'
          DEST  REG(2)
          SCAN  BUF(2),MOVE=NO,    ESTABLISH LENGTH OF COMMAND IMAGE     *
                FLD=CTL3,STOP=(X'00',EQ),CNT=REG(2),LNG='120'
          SCAN  BUF(2),MOVE=NO,    SCAN PAST *CALL                       *
                FLD=CTL3,STOP=(' ',EQ),CNT=REG(3),LNG=REG(2)
          SCAN  BUF(2),MOVE=NO,    SCAN PAST BLANKS                      *
                FLD=CTL3,D=REG(3),STOP=(' ',NE),CNT=REG(1),LNG=REG(2)
          ADD   REG(3),REG(1)
          SCAN  BUF(2),MOVE=NO,    SCAN PAST PROGRAM NUMBER              *
                FLD=CTL3,D=REG(3),STOP=(' ',EQ),CNT=REG(1),LNG=REG(2)
          ADD   REG(3),REG(1)
          SCAN  BUF(2),MOVE=NO,    SCAN PAST BLANKS                      *
                FLD=CTL3,D=REG(3),STOP=(' ',NE),CNT=REG(1),LNG=REG(2)
          ADD   REG(3),REG(1)
          SRC   BUF(2),FLD=CTL3,D=REG(3),LNG=REG(2)
          DEST  BUF(1),FLD='25',LNG=REG(2)
          SRC   'FROM PROC: '      BUILD MESSAGE
          DEST  BUF(1),L=14,LNG=11
          CALLSUB HOSTMSG          ENQUEUE TO HOST
          SRC   '256'
          DEST  REG(1)
          MOVEBUF FROM=BUF(2),TO=BUF(1),LNG=REG(1) RESTORE ENTIRE BUF
          RETURN ,                 SO AS NOT TO CHANGE THE PROC'S ARGS
```

```
*****************************************************************************************
*                                                                                       *
*               COMMON SUBROUTINE                                                       *
*                                                                                       *
*****************************************************************************************
*
*               SUBROUTINE TO ENQUEUE MESSAGES TO THE HOST.
*
*               INCLUDE DATE AND TIMESTAMP.
*
HOSTMSG  SRC    DATE
         TRNSLAT ALL,OLD=X'00',NEW='0'
         DEST   BUF(1),L=0,LNG=6
         SRC    TIME
         TRNSLAT ALL,OLD=X'00',NEW='0'
         DEST   BUF(1),L=7,LNG=6
         SRC    ' '
         DEST   BUF(1),L=6,LNG=1
         DEST   BUF(1),L=13,LNG=1
*
*              QUEUE THE MESSAGE FOR THE HOST.
*
         SRC    '251'
         DEST   REG(3)
         SRC    PGMID               NOT RELEVANT FOR MESSAGE TO HOST
         DEST   REG(2)              .... BUT MAKE SURE IT'S VALID
         PUTMSG REG(3),REG(2)
         RETURN
```

```
*****************************************************************************
*                                                                           *
*               ERROR ROUTINES                                              *
*                                                                           *
*****************************************************************************
*
*               EXCEPTION CONDITIONS.
*
NORESORC SRC    'INSUFFICIENT RESOURCES TO START SUBTASK FOR OPID &USER'
         DEST   BUF(1),L=14,LNG=200
         CALLSUB HOSTMSG
         EXIT
*
NORSRC1  SRC    'INSUFFICIENT RESOURCES TO START SYSCTL3'
DESTMSG  DEST   BUF(1),L=14,LNG=200
         GOTO   PUTMSG
*
NODISK   SRC    'INSUFFICIENT DISK FOR RECNOTE'
         GOTO   DESTMSG
*
RECERR   SRC    'UNEXPECTED CONDITION CODE FROM RECNOTE'
         GOTO   DESTMSG
*
NOTHING  SRC    'NO MESSAGES RECEIVED FROM THE HOST'
         DEST   BUF(1),L=14,LNG=200
         CALLSUB HOSTMSG
         EXIT
*
NODISK2  SRC    'INSUFFICIENT DISK FOR SENDNOTE'
         DEST   BUF(1),L=14,LNG=200
         CALLSUB HOSTMSG
         GOTO   EXIT2
*
*
         ENDPGM
         END
```

C.2.2 MAINTAINING THE MESSAGE DATASET

DPCX command processing routines invoked by SYSCTL3 use inter-task communication notes and messages to pass information back to other routines controlling the operator's terminal command session. Whilst the messages are retrieved to the host, the notes which are enqueued to Destid 236 and a Commid of the user number, in this case 251, are never received because no terminal session can occur for this user. In consequence, notes to destid 236 will gradually accumulate, and in time will start to degrade DPCX performance.

The simple program shown in Figure 68 will delete notes to destid 236 from the message dataset. It is best called from the IPL PROC with an *CALL statement, as there is then no risk of it disrupting processing for other operators.

```
         STARTPGM PGMID=305,ENVIR=NOPRI,PRI=NONE,OPSEL=YES,      *
               ACCESS='11111111',NAME='PURGNOTE'
         SRC   X'01EC'            KEY FOR NOTES TO DESTID 236
         DEST  DSINDEX
         DATASET DSID=6,CONTROL=YES
         DRL   KEY1,BLOCK=0
         EXIT
         ENDPGM
         END
```

Figure 68. Program to maintain the message dataset: this program will clean out the system-generated notes relating to user number 251 which are never retrieved.

D.0 SAMPLE FORMS

This chapter contains the examples of forms for the following purposes:

- Maintenance log of system software changes applied to a DPCX/DOSF system: Figure 69 on page 170.
- Reference application programs installed to SYSINFOREF versions and levels: Figure 70 on page 171.
- Maintenance log of application software installed on a DPCX/DOSF system: Figure 71 on page 172.
- Recording SYSIMOD option group 1 values for a DPCX system: Figure 72 on page 173.
- Recording SYSIMOD option group 2 values for a DPCX system: Figure 73 on page 174.
- Recording SYSIMOD option group 3 values for a DPCX system: Figure 74 on page 175.
- Recording SYSIMOD option group 6 values for a DPCX system: Figure 75 on page 176 and Figure 76 on page 177.
- Recording SYSIMOD option group 7 values for a DPCX system: Figure 77 on page 178.
- Recording SYSIMOD option group 8 values for a DPCX system: Figure 78 on page 179.
- Recording SYSIMOD option group C values for a DPCX system: Figure 79 on page 180.
- Recording RJE logon information for a DPCX system: Figure 80 on page 181.
- Recording RJE FCB definitions: Figure 81 on page 182.
- Recording DOSF values for a DPCX system: Figure 82 on page 183.
- Recording DTF session definitions for a DPCX system: Figure 83 on page 184 and Figure 84 on page 185.
- Recording device addresses, types, identities and locations for a DPCX system: Figure 85 on page 186.

System _____ at _____ Name _____
of Batch LU

Product Identifier	Update Level, Fix Package or PTF Id.	Date Installed

Figure 69. Form to record system software updates: this may be used to record the installation of service to system software on a DPCX/DOSF system.

Product or Module Identifier	Product, PTF or Apar Level	SYSINFOREF Version	Level

Figure 70. Form to record SYSINFOREF versions and levels: this may be used to record the version and level numbers assigned to application entities as they are loaded on the SYSINFOREF library.

System _____ at _____ Name _____
of Batch LU

Product or Module Identifier	SYSINFOREF Version Level		Date Installed

Figure 71. Form to record application software installation: this may be used to record the installation of application software on a DPCX/DOSF system.

SYSIMOD (1) VALUES System _____ at _____
(HOST OPTIONS)

Transaction Record Packing: _____

Transmit Record Blocking: _____

Connection Retry Counter: _____

Open Retry Counter: _____

Station Id: _____

Station Address: _____

Connect-Modem-to-Line Circuit: ___
(Used by Modem: Yes = 1, No = 2)

NRZI Encoding for Modem: ___
(NRZI = 1, not NRZI = 2)

Line Type: ___
(Leased = 1, Switched = 2)

Modem Generates Tone: ___
(Yes = 1, No = 2)

Permanent Request to Send: ___
(Modem can operate with: Yes = 1, No = 2)

Automatic Tests: ___
(Yes = 1, No = 2)

Read / Idle Timeout Counter: _____

Figure 72. Form to record SYSIMOD option group 1 values: this may be used to record the values for SYSIMOD option group 1 which describe the host connection.

SYSIMOD (2) VALUES System _____ at _____
(PRINT GROUP TABLE)

Enter number of sets of 256 records required for each print group:

1: _____ x256	2: _____ x256	3: _____ x256	4: _____ x256
5: _____ x256	6: _____ x256	7: _____ x256	8: _____ x256
9: _____ x256	10: _____ x256	11: _____ x256	12: _____ x256
13: _____ x256	14: _____ x256	15: _____ x256	16: _____ x256
17: _____ x256	18: _____ x256	19: _____ x256	20: _____ x256
21: _____ x256	22: _____ x256	23: _____ x256	24: _____ x256
25: _____ x256	26: _____ x256	27: _____ x256	28: _____ x256
29: _____ x256	30: _____ x256	31: _____ x256	32: _____ x256
33: _____ x256	34: _____ x256	35: _____ x256	36: _____ x256
37: _____ x256	38: _____ x256	39: _____ x256	40: _____ x256
41: _____ x256	42: _____ x256	43: _____ x256	44: _____ x256
45: _____ x256	46: _____ x256	47: _____ x256	48: _____ x256
49: _____ x256	50: _____ x256	51: _____ x256	52: _____ x256
53: _____ x256	54: _____ x256	55: _____ x256	56: _____ x256
57: _____ x256	58: _____ x256	59: _____ x256	60: _____ x256
61: _____ x256	62: _____ x256	63: _____ x256	64: _____ x256

Figure 73. Form to record SYSIMOD option group 2 values: this may be used to record the values for SYSIMOD option group 2 which define the print group table.

SYSIMOD (3) VALUES System _____ at _____
(TRANSACTION DATA SET GROUP TABLE)

Enter number of sets of 256 records required for each TDS group:

1: _____ x256	2: _____ x256	3: _____ x256
4: _____ x256	5: _____ x256	6: _____ x256
7: _____ x256	8: _____ x256	9: _____ x256
10: _____ x256	11: _____ x256	12: _____ x256
13: _____ x256	14: _____ x256	15: _____ x256
16: _____ x256	17: _____ x256	18: _____ x256
19: _____ x256	20: _____ x256	21: _____ x256
22: _____ x256	23: _____ x256	24: _____ x256
25: _____ x256	26: _____ x256	27: _____ x256
28: _____ x256	29: _____ x256	30: _____ x256
31: _____ x256	32: _____ x256	

Figure 74. Form to record SYSIMOD option group 3 values: this may be used to record the values for SYSIMOD option group 3 which define the transaction dataset group table

SYSIMOD (6) VALUES System _____ at _____
(LU - LA TABLE: PART 1)

Host LU Number	DPCX Terminal LA	Printer Group/Mode
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Figure 75. Form to record SYSIMOD option group 6 (part 1) values: this may be used to record the values for SYSIMOD option group 6 which define part 1 of the LU-LA table.

SYSIMOD (6) VALUES System _____ at _____
(LU - LA TABLE: PART 2)

Printer Group Number	Assigned LAs			
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____
_____	_____	_____	_____	_____

Figure 76. Form to record SYSIMOD option group 6 (part 2) values: this may be used to record the values for SYSIMOD option group 6 which define part 2 of the LU-LA table.

SYSIMOD (7) VALUES (RJE PARAMETERS) System _____ at _____

RJE Operator Number: _____

RJE Command Character: ____

RJE Transmit Chain Size: _____

RJE LU Numbers:
1: _____
2: _____
3: _____
4: _____
5: _____

Figure 77. Form to record SYSIMOD option group 7 values: this may be used to record the values for SYSIMOD option group 7 which set the RJE parameters.

SYSIMOD (8) VALUES (SYSTEM OPTIONS) System _____ at _____

1. ABEND Print Status:
(Enable = 1, Disable = 2) ___

2. DSC Mode Selection:
(1976 = 1, 1970 = 2) ___

3. Select Write Verify:
(Enable = 1, Disable = 2) ___

4. System Print Group Number: ____

5. User-Defined Categories:
(Select User-Defined = 1) ___

6. Disable Assist:
(Disable = 1) ___

Figure 78. Form to record SYSIMOD option group 8 values: this may be used to record the values for SYSIMOD option group 8 which define system options.

SYSIMOD (C) VALUES System _____ at _____
(PRINTER MATRIX FOR 8775S)

8775 LA	Associated Printer LAs		
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Figure 79. Form to record SYSIMOD option group C values: this may be used to record the values for SYSIMOD option group C which define the printer matrix for 8775 display terminals.

RJE LOGON DATA System _____ at _____

RJE LU NAMES: 1 _____ 2 _____

3 _____ 4 _____ 5 _____

DATA FOR LOGON-ID 1

Host Logmode Table Entry Name: _____

Host Application Name: _____

Logon User Data: _____

DATA FOR LOGON-ID 2

Host Logmode Table Entry Name: _____

Host Application Name: _____

Logon User Data: _____

DATA FOR LOGON-ID 3

Host Logmode Table Entry Name: _____

Host Application Name: _____

Logon User Data: _____

DATA FOR LOGON-ID 4

Host Logmode Table Entry Name: _____

Host Application Name: _____

Logon User Data: _____

Figure 80. Form to record RJE logon data: this may be used to record the RJE logon data for SYSRJE option 7.1.

RJE FCB NAME: _____

Maximum Page Length: _____

Top Margin: _____

Bottom Margin: _____

Vertical Stop 1 (Channel 2): _____

Vertical Stop 2 (Channel 3): _____

Vertical Stop 3 (Channel 4): _____

Vertical Stop 4 (Channel 5): _____

Vertical Stop 5 (Channel 6): _____

Vertical Stop 6 (Channel 7): _____

Vertical Stop 7 (Channel 8): _____

Vertical Stop 8 (Channel 9): _____

Vertical Stop 9 (Channel 10): _____

Vertical Stop 10 (Channel 11): _____

Figure 81. Form to record RJE FCB data: this may be used to record the RJE FCB data for SYSRJE option 7.1.

DOSF VALUES System _____ at _____

Adjust Mode (yes or no): _____

Line Width: _____

Lines per Page: _____

Tab Stops:

1: _____	2: _____	3: _____	4: _____	5: _____
6: _____	7: _____	8: _____	9: _____	10: _____
11: _____	12: _____	13: _____	14: _____	15: _____
16: _____	17: _____	18: _____	19: _____	20: _____

Overstrike Character: ___

Spaces text <---> line number: ___

Hyphenation Zone Width: ___

System Identification Code: ___

Date Representation: ___
(1 = MMDDYY, 2 = DDMMYY, 3 = YYMMDD)

Automated Text Operator: _____

Data Dictionary Dataset: _____

Figure 82. Form to record DOSF values: this may be used to record the DOSF values set with SYSXGEN or changed with SYSXMNT.

DTF SESSION System _____ at _____

Session: _____ LU Name: _____

Logmode: _____ LU Number: _____

LU Type: ___ Applname: _____
(1 = LU 1 single chain, 2 = LU 1 multiple chain,
3 = LU 0 no RU padding, 4 = LU 0 RU padding)

Session End Prog.: _____ Max. Doc. Size: _____

Default Operator No.: _____ Normal Running Mode: ___
(1 = Open, 2 = Closed)

Max. Chain Size: _____ Pad Character in Hex: ____

Terminator: _____ Session Initiation: ___
(1 both, 2 = DOSF, 3 = Host)

Autoshutdown: ___ Recoverable: ___
(1 = yes, 2 = no) (1 = yes, 2 = no)

Allow Unlisted Queues: ___ Action on Error: ___
(1 = yes, 2 = no) (1 = next queue, 2 = stop)

Figure 83. Form to record DTF session definitions: this may be used to record the values used with SYSXSDEF to define DTF sessions.

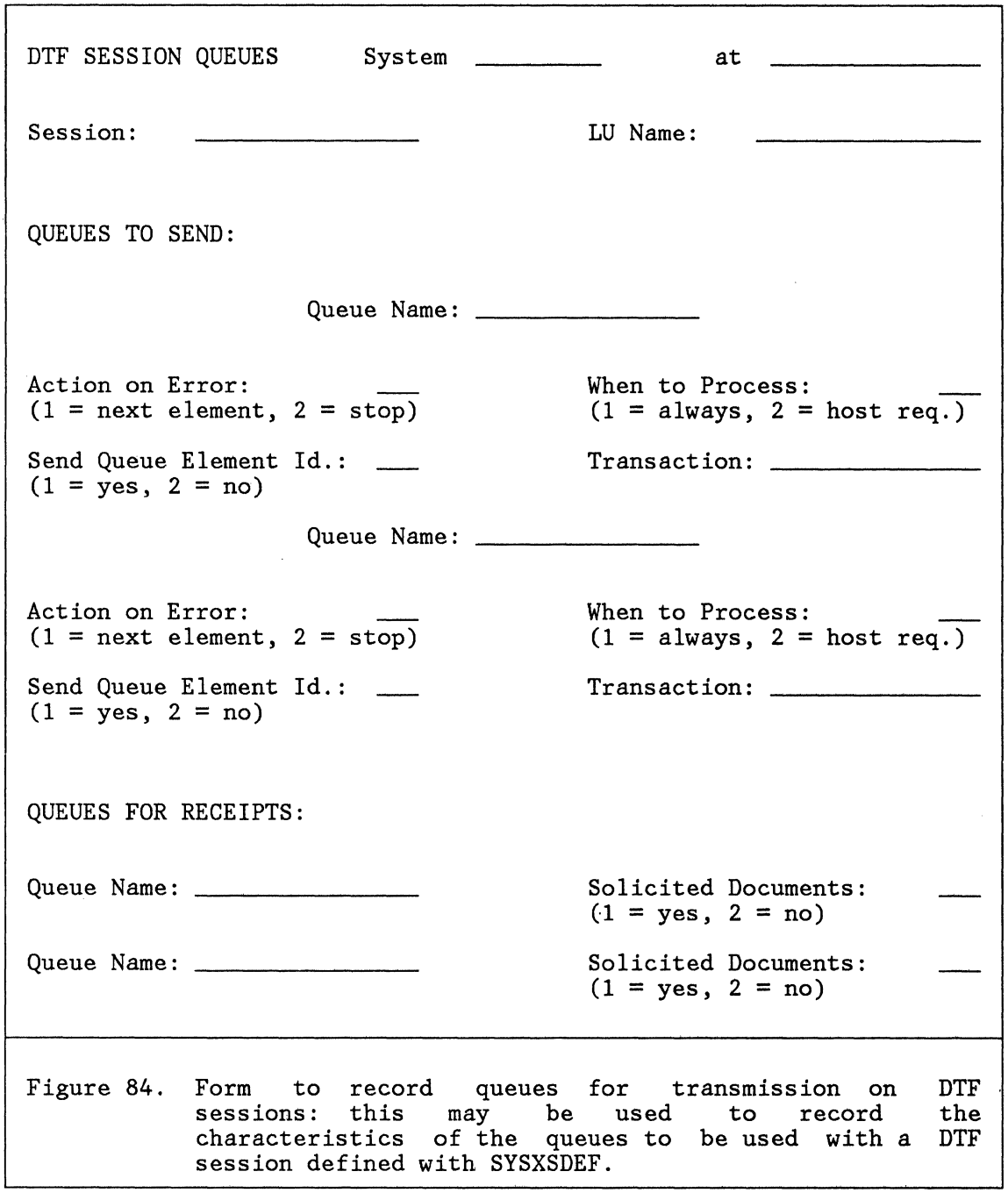


Figure 84. Form to record queues for transmission on DTF sessions: this may be used to record the characteristics of the queues to be used with a DTF session defined with SYSXSDEF.

DEVICES CONFIGURED System _____ at _____

LA	PA	Type	Serial	Location

Figure 85. Form to record the configuration of devices on a DPCX system: this may be used to record the addresses, types, identities and locations of devices attached to a DPCX/DOSF system.

This bibliography is organised in alphabetic sequence of title. A full list of all the system reference library manuals in the DPCX/DOSF family is contained in the DPCX/DOSF Guide to Books. The DPCX/DOSF Master Index and Glossary contains an index of topics.

SX27-0060 DOSF Installation Card for ASSIST. This card documents and illustrates the steps of installing DPCX, DOSF and IDTF with ASSIST.

SC27-0556 DOSF Planning.

This book explains how to plan the use of DOSF and write directions for its installation, configuration and subsequent operation.

SC27-0553 DOSF System Services.

This book is a reference for anyone who uses the DOSF system services.

SC27-0639 DPCX/DOSF Basic Operations.

This book introduces DPCX and describes basic operating procedures for the control operator to use when operating DPCX/DOSF.

SC27-0637 DPCX/DOSF Command Procedures.

This manual contains reference information about the command procedure control commands and arguments that are used within PROCs. Examples of PROCs are included to show some of the things a PROC can do.

SC27-0520 DPCX/DOSF Command Reference.

This book describes the commands that are used to operate a DPCX DOSF system and how to log the activities at an operator's terminal.

SC27-0537 DPCX/DOSF Diagnosis: Guide.

This publication discusses the DPCX and DOSF service aids that are available to help to identify problems within a DPCX system after it has been successfully installed.

GC27-0640 DPCX/DOSF Guide to Books.

This book lists the books needed to use DPCX and DOSF.

SC27-0484 DPCX/DOSF Installation.

This book sets out how to install and configure DPCX and DOSF, with information about PTFs and ASSIST installation.

GC22-9087 DPCX/DOSF Master Index and Glossary

This book contains information about topics and terms from the 8100/DPCX/DOSF system. The Master Index lists which topics are covered by which books, and the Glossary defines terms.

SC27-0523 DPCX/DOSF Messages and Codes

This book lists the messages and codes pro-

duced by DPCX and DOSF, with an explanation of each and suggested actions and responses.

SC27-0482 DPCX Planning.

This manual describes selected aspects of planning applications to use with DPCX.

SC27-0487 DPCX Programming: Guide to Host Communications for System Programmers.

This manual describes the facilities of DPCX systems which are used during communication sessions with the host system.

SC27-0490 DPCX Programming: Guide to Program Structure.

This book explains how to write programs for an 8100 running DPCX as the operating system, with reference to using program resources, performing arithmetic and logic operations and controlling the program sequence.

SC27-0486 DPCX Remote Job Entry: Installation and Operation.

This book tells how to install and operate the RJE facility of DPCX.

SC27-0492 DPCX System Services.

This book describes the System Services the control operator uses when operating an 8100/DPCX system.

SC27-0483 DPCX Terminal Operations: Program Execution Monitor Guide

This manual describes the use of the Program Execution Monitor (System Services SYSDEBUG and SYSTEST) to monitor and test DPCX programs.

SC27-0455 HCF Guide and Reference.

This manual presents a general introduction to HCF, provides information on planning for its installation and specific information for its operation.

SC27-0577 Host Prep: Guide to Host Services.

This book explains how to prepare programs, panels and datasets for end use at a DPCX system.

SC27-0580 Host Prep: SYSINFOREF Guide and Reference.

This book describes the SYSINFOREF portion of the Host Prep program products, with examples of its use and a complete reference of its control statements.

GA27-3341 IBM Multiuse Communication Loop Planning and Installation Guide.

This manual contains detailed information on designing and planning a loop configuration, and on installing and testing the loop cable, hardware and loop accessories.

GC34-2027 Information/System General and Preinstallation Information.

This document provides a general introduction to, and initial planning information for, the Information/System product and its Information/Management, Information/MVS and Information/Access features.

GC27-0429 NCCF General Information.

This book provides an overview of the Network Communications Control Facility.

GC30-3081 NLDM General Information.

This manual describes the functions and use of the Network Logical Data Manager.

GC34-2061 NPDA General Information.

This book provides an overview of the Network Problem Determination Application.

The glossary lists abbreviations and acronyms used in this manual, giving their meanings and a brief description of the concept in question.

ABEND Abnormal end: when a program or software component terminates abnormally.

ACF Advanced Communication Function: IBM program product versions of VTAM with the option to link multiple host domains.

AMS Access Method Services: programs for creating and maintaining VSAM datasets

APAR Authorised Program Analysis Request: the mechanism for resolving an error found in IBM software.

APPLID Application identifier: the name and characteristics with which a host program identifies itself to VTAM as an application.

ASSIST An aid for the installation of DPCX/DOSF 8100 systems which performs a standard customisation.

BDES Batch Data Exchange Services: a part of the Host Prep program products which allows application programs, panels and DSCBs to be written to tape or diskette for transmission to DPCX.

BOP Basic Operator Panel: the panel on the 8130 or 8140 processor containing the IPL switches and the four-digit display and keys used to communicate with the 8100.

CICS Customer Information Control System: a host DB/DC system IBM program product.

CIL Condition Incident Log: the DPCX dataset in which error conditions are recorded.

CLIST Command list: a list of commands for a command-driven program.

CNM Communications Network Management: the management of physically distinct, geographically separated computing resources connected by a TP network.

DASD Direct Access Storage Device: permanent auxiliary storage.

DB/DC Data Base/Data Communication: a system with structured methods of organising and transmitting data.

DCF Document Composition Facility: a host based text formatter IBM program product.

DD Data Definition: a host JCL statement describing a dataset.

DIF Document Interchange Facility: an IBM program product allowing the interchange of documents between DOSF on an 8100 and DLF on a host system.

DISOSS Distributed Office Support System: this IBM program product supports central filing

and retrieval of DOSF documents at a host system and electronic mail capabilities among users of a network of DPCX systems.

DITTO Data Interfile Transfer, Testing and Operations Utility: a DOS/VSE program product with data display and copy functions.

DLA Data Link Adaptor: an SDLC communications link on DPCX.

DLF Document Library Facility: an IBM program product providing a host system with a library facility for documents to be processed with DCF.

DOS/VSE Disk Operating System/Virtual Storage Extended: an IBM host operating system for smaller System/370 and 4300 processors.

DOSF Distributed Office Support Facility: an IBM product providing text processing to an 8100/DPCX system.

DPCX Distributed Processing Control Executive: an IBM operating system for 8100 processors supporting text (DOSF) and data applications.

DPPX Distributed Processing Programming Executive: another IBM operating system for 8100 processors supporting more sophisticated data applications.

DSC Data Stream Compatibility: a feature of DPCX providing a terminal attached to the 8100 access to host applications as if it were directly attached to the host.

DSCB Dataset Control Block: a record that describes the allocation and characteristics of a dataset.

DSX Distributed Systems Executive: this IBM program product resides on a host system and controls data constituting distributed systems.

DTF Document Transmission Facility: a feature of DOSF supporting the transmission of documents between an 8100 and a host system.

FCB Forms Control Buffer: a record defining carriage control for a printer.

FP Function Program: a DPCX user program or System Service.

HCF Host Command Facility: this IBM product allows a host-attached display screen to access DPCX as if the terminal were attached to the 8100.

Host Prep An IBM program product providing host management of DPCX 8100 systems.

IDTF Interactive Display Text Facility; an IBM program product consisting of microcode which gives an 8775 display terminal text entry and edit capabilities.

IMS Information Management System: a host DB/DC system.

IOCB Input/Output Control Block: in DPCX, a

record describing an attached device.

IPL Initial Program Load: the function of initialising a processor with an operating system, such as an 8100 with DPCX.

JCL Job Control Language: statements which instruct a host operating system in the processing of programs and data.

JES Job Entry Subsystem: a component of a host operating system responsible for the management of jobs.

LA Logical Address: the representation of a device within the architecture of DPCX addressing.

LU Logical Unit: the representation of some program within DPCX to VTAM and the NCP.

MVS See OS/VS2/MVS.

NCCF Network Communications Control Facility: an IBM program product which assists a host installation to control and operate an SNA network.

NCP Network Control Program: the software which resides in a 3705 communications controller.

NIM Network Installation Management: a capability of DPCX releases 3 and subsequent to install new releases of DPCX and DOSF across the network from the host.

NLDM Network Logical Data Manager: this IBM program product collects data relating to SNA sessions between network resources and gives a user online access to it.

NPDA Network Problem Determination Application: an IBM program product which assists a host installation in the isolation of network problems.

OS/VS1 Operating System/Virtual Storage 1: an IBM host operating system for medium System/370, 4300 and 303x processors.

OS/VS2/MVS Operating System/Virtual Storage 2/Multiple Virtual Storages: an IBM host operating system for large System/370, 4300, 303x and 308x processors.

PCA Primary Communications Attachment: a loop or link adaptor on an 8100.

PDS Partitioned Dataset: A library file organisation used by OS/VS operating systems.

PROC Procedure: a list of control statements and commands.

PSN Print Sequence Number: used by DPCX to identify a set of spooled print records.

PTF Program Temporary Fix: a package of replacement code issued by IBM to correct defects in a software component.

PU Physical Unit: the representation of the 8100 DPCX system to VTAM and the NCP.

PVS Program Validation Services: a component of Host Prep which prepares application programs for transmission to DPCX.

RJE Remote Job Entry: a facility of DPCX allowing jobs to be input an 8100 to a host system and printed output received back.

RSDS Relative Sequential Dataset: a dataset organisation employed by DPCX.

SASRAC Stand Alone Save Restore and Diskette Copy Program: the name by which the DASD dump restore utility is known in DPCX release 3.

SDLC Synchronous Data Link Control: a link-level transmission protocol employed in IBM SNA networks.

SNA Systems Network Architecture: an IBM architecture for data communications networks.

SSCP System Services Control Point: the program controlling an SNA network, for instance VTAM.

SSS Subsystem Support Services: IBM systems control programming supporting the management of data comprising distributed systems; largely superceded by later IBM program products.

SYSINFOREF Subsystem Information Retrieval Facility: a component of Host Prep which communicates programs, PTFs and other information between a host system and DPCX systems.

TAF Terminal Access Facility: a component of NCCF allowing an NCCF terminal access to other applications.

TP Tele-processing: descriptive of data communications techniques employed between geographically separated locations.

VSAM Virtual Storage Access Method: a dataset organisation and access method used by IBM host operating systems.

VTAM Virtual Telecommunications Access Method: an IBM SNA data communications access method.

GG24-1578-0

READER'S COMMENT FORM

OFFICE SYSTEMS INTERCONNECTION: GUIDE TO NETWORK MANAGEMENT FOR DPCX/DOSF SYSTEMS

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation.

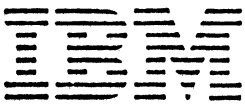
Reader's Comment Form

Cut or Fold Along Line

Fold

Raleigh International Systems Center
Department 985 / H594
Building 622-3
P.O. Box 12195
Research Triangle Park
Raleigh, North Carolina 27709
U.S.A.

Fold



GG24-1578-0