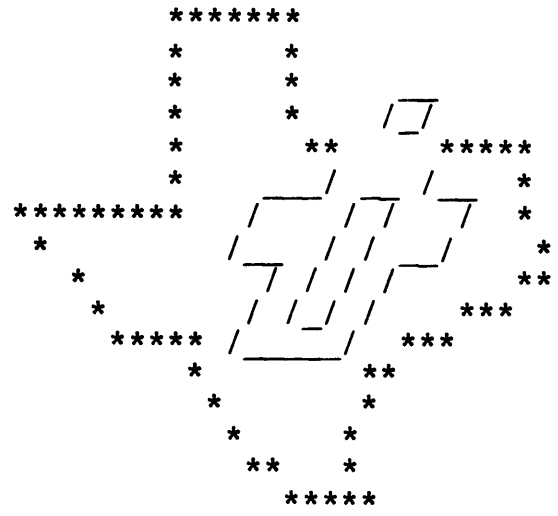
DATA SYSTEMS GROUP



D N O S

D N C S N U C L E U S

R E L E A S E I N F O R M A T I O N

Release 1.1.0

Part Number 2276805-9901*A

1 February, 1983

Texas Instruments
Incorporated

TABLE of CONTENTS

Paragraph Title

SECTION 1 GENERAL INFORMATION

1.1 INTRODUCTION
1.2 DNOS COMMUNICATIONS SUBSYSTEM
1.2.1 INTERFACE MODULES
1.2.1.1 FOUR CHANNEL COMMUNICATIONS CONTROLLER
1.2.1.2 BIT-ORIENTED/CHARACTER-ORIENTED ASYNCHRONOUS INTERFACE MODULE
1.2.1.3 X.21 BIT-ORIENTED/CHARACTER-ORIENTED ASYNCHRONOUS INTERFACE MODULE
1.2.2 USING A COMBINATION OF COMMUNICATIONS INTERFACE MODULES
1.2.3 USE WITH BELL 201B OR SYNTEK MODEMS
1.2.4 COMMUNICATION DOWNLOAD UTILITY
1.2.4.1 DOWNLOAD UTILITY LOG MESSAGES
1.2.4.2 DOWNLOAD UTILITY ERROR CODES AND EXPLANATION
1.2.5 FCCC PROBLEMS
1.2.6 COMMUNICATION SUBSYSTEM PROBLEMS
1.2.6.1 INITIATE EVENT COUNT PROBLEM
1.2.6.2 USE OF THE COMM DEVICE LIST MEMORY (CLM) COMMAND AND SYSTEM CRASHES
1.3 DNCS SYSTEM GENERATION
1.3.1 DNCS GENERATION UTILITY (XDGU)
1.3.2 VERIFY DEVICE CONFIGURATION (VDC)
1.4 MULTIPLE DNCS CONFIGURATIONS
1.4.1 INSTALLING MULTIPLE CONFIGURATIONS
1.4.2 EXECUTING A SELECTED CONFIGURATION
1.5 MEMORY REQUIREMENTS
1.6 DNOS COMMUNICATION SUBSYSTEM MEMORY REQUIREMENTS
1.7 DNOS SYSTEM TABLE REQUIREMENTS
1.8 DNCS BUFFER SEGMENT REQUIREMENTS
1.9 DISK UTILIZATION
1.10 HOW TO REPORT A SYSTEM CRASH
1.11 DNCS 1.0 TO DNCS 1.1 MIGRATION

SECTION 2 KNOWN PROBLEMS

2.1 SOFTWARE
2.2 DOCUMENTATION - DNCS OPERATIONS GUIDE

SECTION 3 PATCHES AND PATCH PROCEDURES

3.1 PATCH UPDATE PROCEDURE
3.2 APPLYING PATCHES RECEIVED AT A LATER DATE

SECTION 1

GENERAL INFORMATION

1.1 INTRODUCTION

This document contains information about the DNCS Nucleus product, release 1.1.0, that is not contained in the standard documentation associated with the object installation kit.

The subjects that are discussed in this document are special features or considerations that may be important for the proper installation and operation of the object package.

1.2 DNOS COMMUNICATIONS SUBSYSTEM

The following paragraphs describe communication subsystem features relating to the current release.

1.2.1 INTERFACE MODULES. The supported communication interface modules are FCCC, BCAIM, and X.21 BCAIM.

1.2.1.1 FOUR CHANNEL COMMUNICATIONS CONTROLLER. (FCCC or CP503). The FCCC occupies one full slot in the 990 chassis. Refer to the FCCC Installation and Operation Manual (part number 2263878-9701) for a detailed description of the board and its appropriate slot position and interrupt level. The recommended interrupt level and TILINE address is interrupt 8 and TILINE address >F900. The TILINE address is switch selected on the FCCC board and is independent of the slot it occupies in the chassis.

NOTE

Use communications interface cable, part number 946117-0001 or -0002, when connecting an FCCC channel to an external modem. If a full 24 pin cable is used, the FCCC's channel's receive circuitry will be disabled,

even with pin 24 cut.

1.2.1.2 BIT-ORIENTED/CHARACTER-ORIENTED ASYNCHRONOUS INTERFACE MODULE. (BCAIM or CP501). The BCAIM occupies one-half slot in the 990 chassis. Refer to the BCAIM Installation and Operation Manual (part number 2263883-9701) for a detailed description of the board and its appropriate slot position and interrupt level. The recommended interrupt level is 7.

NOTE

When ordering the BCAIM communications Interface Kit (part number 2303091-0002), the kit will include either cable 2303070-0002 or 2303079-0002.

1.2.1.3 X.21 BIT-ORIENTED/CHARACTER-ORIENTED ASYNCHRONOUS INTERFACE MODULE. (X.21 BCAIM or CP502). The X.21 BCAIM occupies one-half slot in the 990 chassis. Refer to the X.21 BCAIM Installation and Operation Manual (part number 2263886-9701) for a detailed description of the board and its appropriate slot position and interrupt level. The recommended interrupt level is 7.

NOTE

When ordering the X.21 BCAIM Communications Interface Kit (part number 2265184-0002), the kit will include either cable 2303070-0002 or 2303079-0002.

1.2.2 USING A COMBINATION OF COMMUNICATIONS INTERFACE MODULES. In systems where a combination of the above interface modules are present, the BCAIM or X.21 BCAIM should have a higher interrupt level than the FCCC. The higher the interrupt level, the lower its numerical value. In general, the load placed upon the CPU is directly related to the number of interrupts. Interrupt levels, interrupt rates, and the number of devices being supported are all factors to be considered when configuring an operating system.

1.2.3 USE WITH BELL 201B OR SYNTEK MODEMS. The BELL set interface (part number 946104-0002) does not operate properly with the BELL 201B modem or SYNTEK modems unless the cable (part number 946117-0001) is modified by cutting the wire connected to pin 24 of the male cable connector. Mark the cable to indicate this modification.

1.2.4 COMMUNICATION DOWNLOAD UTILITY. The DNCS Nucleus uses the Communications Download Utility (CDL) to download firmware patches to the FCCC.

1.2.4.1 DOWNLOAD UTILITY LOG MESSAGES. The system log records a message indicating the success of the download operation. If successful, a message appears in the following format:

```
ddd:tttt+      COMMDWNL(CMxx,80.354),L=yyyy,
       +                P=.S$COMM.S$COMMDN
               ,B=0000
```

Unsuccessful download completion messages appear in the following format:

```
ddd:tttt+      COMMDWNL(CMxx,80.354),E=xxxx,
       +                P=.S$COMM.S$COMMDN
               ,B=0000
ddd:tttt+      COMMDWNL(CMxx,80.354),R=yyyy,
       +                P=.S$COMM.S$COMMDN
               ,B=0000
```

Unsuccessful download completion messages may also appear in the following format if the communications device is broken. In this case, verify that the communications device has been included in the system generation correctly, that it is processing interrupts, and that it passes diagnostics.

```
ddd:tttt+      COMMDWNL(CMxx, ),E=xxxx,
       +                P=.S$COMM.S$COMMDN
               ,B=0000
```

Where:

ddd	is the julian day.
tttt	is the time that the message is logged.
CMxx	is the device name of the communication channel.
L=yyyy	is the load address.
E=xxxx	is the error returned during execution of the download task.
R=yyyy	is the starting address of the area that has been

released due to the error indicated by E=xxxx.
B=0000 is the BLWP address.
P= is the name of the program file that contains the code being downloaded.

1.2.4.2 DOWNLOAD UTILITY ERROR CODES AND EXPLANATION. The following is the set of errors that may be returned by the download utility and the action to be taken to correct the indicated problem.

ERROR CODE	ACTION/EXPLANATION
0001	An invalid BLWP address has been specified in the input control file used for a download or release function. Check the input control file specified to be sure that the BLWP vector field (columns 49-52) is not blank or that the address entered is a valid hex address.
0002 0100	An invalid address has been specified as the starting point of a release reserved block function in an input control file, or the control file specified has an invalid ordering of download and release functions. Check for valid address specifications for release functions in the control file being executed. When both download and release functions are specified in the same control file, all release functions must preceed any download functions.
0003	Error in format of Input Control File.
0004 0008 0010 0020 0040 1000	The download object code specified in a control file is invalid due to one of the following: -the object file specified contains an EOF before the normal end of object file record (':' record) -the ASCII object file contains a checksum error -the object file contains an invalid tag -the file specified is not ASCII or compressed object -the file specified is not a program file -the length field of the object file is less than or equal to zero

-the object file contains a bad address following a tag

Check to be sure the correct file has been specified as the download object or attempt to recover the object file.

0080 Download utility task execution error.

Reinstall the utility tasks.

02xx SVC error in assign luno, open program file, read record, release luno, or close program file. Where xx is the DNOS SVC error code.

The program file name or task specified in the Input Control File may be incorrect. See that the program file and/or specified task exists.

08xx An SVC error has occurred during the processing of opcode 8. Where xx is the DNOS SVC error code.

Check to be sure that the .S$ISBTCH batch stream does not contain two CDL (Comm Device Download Utility) requests that attempt to download the same code to the same communications device. In some cases, the error code received will be returned to the access name specified in response to the LISTING ACNM prompt in the CDL proc.

2000
4000 The synonyms 'COMM' and '$CMLU' have invalid values assigned to them.

Check to be sure that the utility tasks are being executed via the standard procs provided with the utility task installation procedure.

8000 The task specified as the download object in a control file has attached procedures or overlays.

Check to be sure the correct task and program file have been specified and that the task does not have any associated overlays or procedures.

1.2.5 FCCC PROBLEMS. The Comm Device List Memory (CLM) utility command may be used to test the FCCC. Enter the FCCC comm device

name and the starting address 09A. If the CLM command executes properly, the version of the firmware will be displayed. This version should be 80.354 or later. Display of this information proves that the FCCC is processing interrupts correctly, thus eliminating a mis-match between the address/interrupt on the board and the system configuration. If problems persist, the FCCC should be tested using diagnostics.

1.2.6 COMMUNICATION SUBSYSTEM PROBLEMS. Problems described here affect the communication subsystem. Patches to correct any problems in the subsystem will be furnished in the patch files: DNPCMON, DNPCSWS, and DNPCOMA. These patch files are applied during the Comm DSR generation process when the procedure, PCS, is executed.

NOTE

The Comm DSR patch files must be applied whenever the system is linked even if no changes have been made to any of the Comm DSR's.

1.2.6.1 INITIATE EVENT COUNT PROBLEM. Patch 1933 in the DNPCMON patch file eliminates problems with initiate event counts in the TSB when the communications PDT is opened in the SVC >4E special mode. Without this patch the communications package that has opened the PDT in SVC >4E mode cannot be killed.

1.2.6.2 USE OF THE COMM DEVICE LIST MEMORY (CLM) COMMAND AND SYSTEM CRASHES. When a communications package executing on the FCCC has opened the communications PDT in SVC >4E special mode, use of the Comm Device List Memory (CLM) command can cause system crash >65 or a hang of the CLM task. Patch 1934 in the DNPCMON patch file must be applied to eliminate this problem. DNOS 1.1 kernel patch 1938 is also required to eliminate this problem.

1.3 DNCS SYSTEM GENERATION

The following paragraphs describe DNCS System Generation (DNCSGEN) features and limitations relating to the current release.

1.3.1 DNCS GENERATION UTILITY (XDGU). XDGU presents a user friendly interface for entering required configuration parameters. The following information may be helpful in preventing operational problems that may occur when trying to use XDGU.

1. There should be no gaps in LU specification. Omitted LUs are implied to be available and may be used in a pooled fashion. This may allow devices to log on with wrong SLU types.

2. The DNCSGEN configuration cannot be modified by text editing the TEXTCONF file. Unpredictable and erroneous configurations may result. Change the configuration only thru XDGU.

3. LU names in DNCS 1.1 are generated by DNCS using the PU name and the LU number (1 - >FF). They are created by a software algorithm and are not user specifiable beyond this extent. For example, if the PU name is CDAANS, then the LU name for LU 01 would be LDAANS01. Note that the LU name always begins with L. For DNCS communication, it does not matter that the LU names in the host Network Control Program (NCP) are different; however for human communication it may be desirable to keep a list that maps the corresponding names, if they are different. Another way of accurately talking about a DNCS LU is to identify its PU, its number, and its circuit name. The most desirable alternative is for the user to conform his LU names in his NCP to DNCS LU names.

4. While in XDGU, it is possible to modify a circuit which supports resources, such as CIPC circuits, to a circuit which does not support resources, such as SDLC circuits. This is an error condition. A circuit so defined should be deleted and re-added.

5. Pressing the CMD key when entering data for an entity while in XDGU causes the cursor to appear for possible entry of a new command. If it is desired to then continue entering data for the same entity, enter HELP, then press RETURN twice.

6. The 'print' key under XDGU works if TIFORM is installed on the user's system. If TIFORM is not installed, the 'print' key does not work and the following message appears: 'INTERNAL ERROR CODE >0041 xxxxx'.

1.3.2 VERIFY DEVICE CONFIGURATION (VDC). VDC is not intended to find all possible errors that might be entered when creating a particular DNCSGEN configuration. In particular, the following is a list of errors you might make during DNCSGEN that VDC will not catch:

1. Duplicate use of synonym names for application definitions under multiple VPUs.

2. More than one circuit definition referencing the same port name.

3. Pooled LU assignments for resources of type SVQ or KSR. These resource types require a dedicated LU.

4. A circuit referencing a port which is defined on the wrong type board to support the required function, i.e., an IPC circuit attached to a port which is defined on a FCCC board, or a SDLC circuit attached to a port which is defined on a VIRTUAL board.

1.4 MULTIPLE DNCS CONFIGURATIONS

The following information is presented as an aid to the network administrator who may have the requirement for keeping multiple versions of DNCS installed on a single system. Execution of the following steps will allow you to install these systems and to successfully bring into execution the selected configuration.

1.4.1 INSTALLING MULTIPLE CONFIGURATIONS. In order to install multiple configurations, you must perform the following steps. This example assumes that CONFIG1 is the selected configuration.

1. Create a unique working installation directory for each DNCS configuration to be installed. For example, you could use the configuration directory that was created during the XDGU process. For the purposes of this discussion, assume that this directory is SYS.SDGU.CONFIG1, where SYS is the system volume. Under this directory create a program file called .S$UTIL and a command directory called .D$CMDS to hold the installed parts.

2. When the INSDC prompt is entered, answer the last 3 prompts as follows:

```
DNCS SYSTEM VOLUME:  SYS.S$DGU$.CONFIG1
DNOS SYSTEM VOLUME:  SYS.S$DGU$.CONFIG1
DNCS COMMAND DIRECTORY:  SYS.S$DGU$.CONFIG1.D$CMDS
```

Note that multiple configurations can be installed by entering the appropriate configuration name, for example, CONFIG2, CONFIG3, etc., provided the program file and command directory are created as indicated. Each configuration must be patched via the PATDC procedure.

1.4.2 EXECUTING A SELECTED CONFIGURATION. In order to put a selected configuration into execution, you must perform the following steps. This example assumes that CONFIG1 is the selected configuration.

1. Terminate DNCS and all its associated packages.
2. Execute the following Copy Directory:

   CD INPUT=SYS.SDGU.CONFIG1.S$UTIL, OUTPUT=SYS, OPTIONS=REP

3. IPL the system.
4. Enter the following command to access the CONFIG1 command library and to allow normal operations to begin:

   .USE SYS.SDGU.CONFIG1.D$CMDS, .S$CMDS

1.5 MEMORY REQUIREMENTS

The DNCS Nucleus job consists of at most nine task segments, one procedure segment, and ten buffer segments. The following tables describe the memory requirements for the DNCS job. All variables are specified during the generation of DNCS as described in the DNCS System Generation Reference Manual.

name	size(bytes)	program file	memory resid?	proc/ task
DNCSCOMM	7668	<dnos volume>.S$UTIL	YES	proc
DNCSPDCT	15256 + (a)	<dnos volume>.S$UTIL	YES	task
DNCSSCT	14380 + (b)	<dnos volume>.S$UTIL	NO	task
DNCSCLOK	1370	<dnos volume>.S$UTIL	NO	task
DNCSCI	41784	<dnos volume>.S$UTIL	NO	task
Buffer seg	(c)		YES	seg
Total	80458 + (d)			

Where:

(a) = 28*(no. of CIRCUITs)
+ 58*(no. of RESOURCEs on all CIRCUITs)
+ 58*(no. of CIRCUITs with PROTOCOL of SDLC)
+ 58*(no. of CIRCUITs with PROTOCOL of LAP)
+ 14*(no. of BOARDs) + 30*(no. of PORTs on all BOARDs)
+ 92*(sum of all MAXREADS for all CIRCUITs)
+ 58*(no. of CONCURRENT XFERS for SUBSYSTEM of TYPE RFT)

(b) = 6*(no. of PUs) + 8*(no. of BOARDs) + 8*(no. of CIRCUITs)
+ 8*(no. of PORTs on all BOARDs)
+ 8*(no. of RESOURCEs on all CIRCUITs)
+ 126 [if CIRCUIT defined with PROTOCOL of LAP]
+ 10 [if CIRCUIT defined with PROTOCOL of RFT]
+ 10 [if PU defined with ACCESS METHOD of PSN]
+ 8*(no. of leased line NETWORK NAMEs on all DTEs)

(c) = 15780*(no. of buffer segments defined in optional patch no. 1695)

(d) = 6*(no. of PUs) + 36*(no. of CIRCUITs)
+ 66*(no. of RESOURCEs on all CIRCUITs)
+ 58*(no. of CIRCUITs with PROTOCOL of SDLC)
+ 58*(no. of CIRCUITs with PROTOCOL of LAP)
+ 92*(sum of all MAXREADS for all CIRCUITs)
+ 22*(no. of BOARDs) + 38*(no. of PORTs on all BOARDs)
+ 58*(no. of CONCURRENT XFERS for SUBSYSTEM of TYPE RFT)
+ 8*(no. of leased line NETWORK NAMEs on all DTEs)
+ 15780*(no. of buffer segments defined in optional patch no. 1695)
+ 126 [if CIRCUIT defined with PROTOCOL of LAP]
+ 10 [if CIRCUIT defined with PROTOCOL of RFT]
+ 10 [if PU defined with ACCESS METHOD of PSN]

The DNCS Nucleus job includes three additional tasks for support of SNA. The following table describes the memory requirements for these tasks.

name	size(bytes)	program file	memory resid?	proc/ task
DNCSTSR	21308 + (a)	<dnos volume>.S$UTIL	NO	task
DNCSPC	9884 + (b)	<dnos volume>.S$UTIL	NO	task
DNCSDC	17288 + (c)	<dnos volume>.S$UTIL	NO	task
Total	48486 + (d)			

Where:

(a) = 22*(no. of VPUs for all PUs) + 152*(no. of PUs)
+ 20*(no. of APPLICATIONs for all VPUs)
+ 14*(no. of APPLICATIONs for all VPUs with SYNs defined)
+ 20*(no. of RESOURCEs on all CIRCUITs)

(b) = 26*(no. of PUs) + 2*(no. of LUs for all PUs)
+ 10 [if PU defined with ACCESS METHOD of PSNI

(c) = 32*(no. of RESOURCEs on all CIRCUITs)
+ 10*(sum of all MODEs with APPLICATION defined)
+ 10*(no. of RESOURCEs on all CIRCUITs or no. of pooled LUs, whichever is greater)

(d) = 178*(no. of PUs) + 2*(no. of LUs for all PUs)
+ 22*(no. of VPUs for all PUs)
+ 72*(no. of RESOURCEs on all CIRCUITs)
+ 20*(no. of APPLICATIONs for all VPUs)
+ 14*(no. of APPLICATIONs for all VPUs with SYNs defined)
+ 10*(sum of all MODEs with APPLICATION defined)
+ 10*(no. of RESOURCEs on all CIRCUITs or no. of pooled LUs, whichever is greater)
+ 10 [if PU defined with ACCESS METHOD of PSNI

The DNCS Nucleus job includes one additional task for the support of SNA using X.25. The following table describes the memory requirements for this task.

name	size(bytes)	program file	memory resid?	proc/ task
DNCSNIA	8364 + (a)	<dnos volume>.S$UTIL	NO	task

Where:

(a) = 856*(no. of PUs with ACCESS METHOD of PSN)

The DNCS Nucleus job includes one additional task for the support of X.25. The following table describes the memory requirements for this task.

name	size(bytes)	program file	memory resid?	proc/ task
DNCSNAP	20770 + (a)	<dnos volume>.S$UTIL	NO	task

Where:

(a) = 76*(no. of DTEs) + 98*(no. of VIRTUAL CIRCUITS on all DTEs)
+ 16 [if PU defined with ACCESS METHOD of PSN]
+ 2*(no. of PUs with ACCESS METHOD of PSN)
+ 16 [if SUBSYSTEM defined with TYPE of RFT]
+ 2*(no. of VIRTUAL CIRCUITS of TYPE SR, SO, RO)
+ 2*(no. of VIRTUAL CIRCUITS of TYPE PE and CIRCUIT USER of RFT)

The DNCS Nucleus Utilities consist of twelve tasks. The following table describes the memory requirements for the DNCS Nucleus Utilities.

name	size(bytes)	program file	memory resid?	proc/ task
DNCSINIT	2588	<dncs volume>.S$DNCS.PGMTASK	NO	task
DNCSSTOP	2060	<dncs volume>.S$DNCS.PGMTASK	NO	task
EMSCI	11040	<dncs volume>.S$DNCS.PGMTASK	NO	task
SVQHST	14774	<dncs volume>.S$DNCS.PGMTASK	NO	task
SVQSCI	6970	<dncs volume>.S$DNCS.PGMTASK	NO	task
SVQQUE	6666	<dncs volume>.S$DNCS.PGMTASK	NO	task
SVQCLR	5398	<dncs volume>.S$DNCS.PGMTASK	NO	task
SVQSET	6442	<dncs volume>.S$DNCS.PGMTASK	NO	task
DNCSGEN	47568	<dcfwo>.S$DCGEN.DNCSGEN	NO	task
BLDDEF	43866	<dcfwo>.S$DCGEN.DNCSGEN	NO	task
BLDPROC	19488	<dcfwo>.S$DCGEN.DNCSGEN	NO	task
VDC	30856	<dcfwo>.S$DCGEN.DNCSGEN	NO	task

The DNCS Nucleus job downloads code to the FCCC for support of SDLC and LAP circuits. The following table describes memory requirements for the DNCS Nucleus download code.

name	size(bytes)	program file
DLSDLC	4374 + (a)	<dncs volume>.S$DNCS.PGMTASK
LAP	10606 + (b)	<dncs volume>.S$DNCS.PGMTASK

Where:

(a) = 328*(no. of writes outstanding on board)
+ 328*(sum of all MAXREADS for all CIRCUITS with PROTOCOL of SDLC)

(b) = 278*(sum of all MAXREADS for all CIRCUITS with PROTOCOL of LAP)

1.6 DNOS COMMUNICATION SUBSYSTEM MEMORY REQUIREMENTS

The DNOS communication subsystem is defined during DNOS system generation and installed during DNCS object installation. The subsystem is made-up of three parts: communication software level scheduler (COMSWS), DSR overlays, and communication SVC processor (IOCSVC). COMSWS is configurable and is linked into the DNOS system kernel during ALGS. DSRCMNS, DSRSDLC, and DSRCOMA are the overlays which support DNCS/IPC, SDLC, and FCCC, respectively, and are loaded at system IPL. IOCSVC is linked into the DNOS system SVC overlay. The following table describes the memory requirements for the DNOS communication subsystem. All variables are specified during the generation of DNOS as described in the DNCS Nucleus Object Installation Guide.

name	size(bytes)	program file	memory resid?	proc/ task
COMSWS	1026 + (a)	<dnos volume>.<system name>	YES	proc
DSRCOMA	3788	<dnos volume>.<system name>	YES	ovly
DSRCMNS	5320	<dnos volume>.<system name>	YES	ovly
DSRSDLC	8128	<dnos volume>.<system name>	YES	ovly
IOCSVC	356	<dnos volume>.<system name>	YES	ovly

Where:

(a) = 98*(no. of DNCS IPC devices)
+ 8*(sum of all DNCS IPC SESSIONs for all DNCS IPC devices)
+ 104*(no. of FCCC devices) + 162*(no. of BCAIM devices)
+ 36*(no. of device channels with PROTOCOL of COMA)
+ 252*(no. of device channels with PROTOCOL of SDLC)

1.7 DNOS SYSTEM TABLE REQUIREMENTS

The DNOS system table area is impacted both directly and indirectly by the DNOS communication subsystem and DNCS communication requirements. Directly by inclusion of the Physical Device Tables (PDTs) for each communication channel. Indirectly by increasing the DNOS system root, which limits the maximum allowed system table size. The following equation may be used to calculate the impact on system table area (in bytes) for support of DNCS communications. All Variables are specified during the generation of DNOS with DNCS communication support as described in the DNCS Nucleus Object Installation Guide.

```
1026 + 98*(no. of DNCS IPC devices)
+ 8*(sum of all DNCS IPC SESSIONs for all DNCS IPC devices)
+ 104*(no. of FCCC devices) + 162*(no. of BCAIM devices)
+ 214*(no. of device channels with PROTOCOL of COMA)
+ 178*(no. of device channels with PROTOCOL of CMNS)
+ 430*(no. of device channels with PROTOCOL of SDLC)
```

1.8 DNCS BUFFER SEGMENT REQUIREMENTS

The DNCS Nucleus creates buffer segments at execution time for storage of I/O buffers. These buffer segments are then accessed by each DNCS Nucleus task via mapping. Depending on the configuration, multiple buffer segments may be needed. DNCS can handle multiple buffer segments and must be patched in order to create more than three at execution time.

The number of buffer segments required for a particular configuration may be calculated by using the following equation. All variables are specified during the generation of DNCS as described in the DNCS System Generation Reference Manual.

```
no. of buffer segments = [2*(sum of all MAXREADs for all CIRCUITs)
                         + (no. of RESOURCEs on all CIRCUITs)
                         + (no. of CIRCUITs with PROTOCOL of SDLC)
                         + (no. of CIRCUITs with PROTOCOL of LAP)
                         + (no. on CONCURRENT XFERS for SUBSYSTEM
                            of TYPE RFT)]
                         / 54
```

The result should be rounded to the next whole number and can not exceed 10. Procedures for patching DNCS to allocate multiple buffer segments are documented in paragraph 2.1.

1.9 DISK UTILIZATION

The following table summarizes the disk requirements for the DNCS Nucleus. The figures are estimates and will vary depending on the number of sectors/ADU of the disk and configuration parameters. Generally, the larger the sectors/ADU the more disk space required, due to disk allocation on an ADU basis. Also, the more configurable resources defined in DNCS the more disk space required. The ADU size in the following table is based on 256 bytes/ADU.

name	disk resident space (adus)
DNCS NUCLEUS DIRECTORY (DCFWO)	12500-14500
DNOS SYSGEN DATA DIRECTORY (S$OSLINK)	750
DNCS GENERATION DIRECTORY (SDGU)	1650
DNOS SYSTEM DIRECTORY (S$UTIL)	950
DNCS SYSTEM DIRECTORY (S$DNCS)	550
DNCS COMMAND DIRECTORY	450

1.10 HOW TO REPORT A SYSTEM CRASH

When a system crash occurs, refer to the DNOS Messages and Codes Reference Manual for an explanation of the crash code shown on the programmer panel of the CPU. To conduct a crash analysis, refer to the DNOS Systems Programmer's Guide. If the crash is not understandable, or if the crash continues to occur, or if the crash was forced, you may want to send information to your customer representative for analysis. Before sending the crash file, be sure that it was created large enough to contain the entire system image.

If your software was supplied to you by Texas Instruments and you are sending the information to Texas Instruments for analysis, please send the following information on either magnetic tape, diskettes, or some other disk media:

1. The .S$CRASH file from the system crash saved according to procedures described in the DNOS Messages and Codes Reference Manual.
2. The DNOS system linkmaps (found in files SYSMAP, IOUMAP, DMMAP), system configuration (found in file CONFIG), communication subsystem linkmaps (found in

files DMAPCOMA, DMAPCMNS, DMAPSDLC, CMAPCSWS), and communication subsystem configuration (found in file LSTCFDSR) for the system in use at the time of the crash. If these linkmaps are on the running system, they can be found in the directory .SSGU.<system name>.

3. The DNCS Nucleus linkmaps found in the directory <dncs generation volume>.SDGU.<configuration>.LINKMAP for the system in use at the time of the crash.

4. The DNCS system configuration found in the file <dncs generation volume>.SDGU.<configuration>.TEXTCONF for the system in use at the time of the crash.

5. The DNCS configurable task definition files found in the directory <dncs generation volume>.SDGU.<configuration>.LIST.

6. A text file with information about the system activity at the time of the crash.

7. A current listing of your software configuration, the output from the List Software Configuration (LSC) command.

Once the information has been copied, the medium will be returned to you. As soon as the problem is resolved, you will be informed.

1.11 DNCS 1.0 TO DNCS 1.1 MIGRATION

It is recommended that you install DNCS 1.1 on a clean DNOS 1.1 system volume. However, if you are installing DNCS 1.1 on a system volume that has been used to run DNCS 1.0, be sure that the system volume contains DNOS 1.1, that the DNCS common procedure DNCSCOMM has been deleted from the system program file .S$SHARED, and that the system initialization batch stream, .S$ISBTCH, has been modified to delete the CDL procedures for DNCS devices.

The DNCS configurations created by the DNCS 1.0 generation utility are not totally compatable with DNCS 1.1, so they must be either regenerated or updated before they may be used to create a DNCS 1.1 system. If you recreate the systems, then it is recommended that you delete the directory .SDGU before

beginning the generation process. If you wish to update your existing configurations, then the following steps must be executed:

1. Enter the name of the existing configuration in response to the 'INPUT CONFIGURATION' prompt of the XDGU proc.

2. Modify each PU defined in the existing configuration. This may be done by entering the XDGU command MPU, then entering the name of the PU. Depressing 'ENTER' will then fill in the previously defined data for the PU, as well as meaningless data for the new fields. Enter the correct data for the new fields XID, REMOTE DTE, SLU1 LOGMODE, SLU2 LOGMODE, AND SLU3 LOGMODE.

3. Modify each BOARD defined in the existing configuration. This may be done by entering the XDGU command MB, then entering the name of the board as previously defined. Enter the correct board type, either FCCC or VIRTUAL, to complete the screen.

4. Modify each CIRCUIT that has a protocol type of SDLC. This may be done by entering the XDGU command MC, then entering the name of the circuit. Depressing 'ENTER' will fill in the protocol prompt (SDLC). Depressing 'ENTER' again will display the remaining prompts, and the existing values for those previously defined. Enter the required responses to the NRZI and AUTO ANSWER prompts.

5. Exit the XDGU utility by entering the QUIT command, specifying YES to the prompt 'Are You Sure?'.

This concludes the modifications required to update a DNCS 1.0 configuration so that it may by used for DNCS 1.1. The resulting configuration is used in the continuation of the installation process, including verifying the configuration (VDC), building the new configurable tables (BDCT), etc.

SECTION 2

KNOWN PROBLEMS

This section documents known problems that may be encountered in installing and operating the DNCS Nucleus object package.

2.1 SOFTWARE

1. Installation of DNOS KERNEL patch 1770 is required before DNCS will operate correctly. Since this patch is shipped as an optional patch, you must text edit the KERNEL patch file and remove the asterisks from patch 1770 before it will apply.

2. When certain Communication Request Blocks (CRBs) are not completed within 10 seconds, DNCS automatically (without operator intervention) issues a master reset to the FCCC. This mechanism is used to force completion of the CRBs so that DNCS does not have to crash. This FCCC reset, however, will require a restart of other communication products, such as 3780, that are sharing the board with DNCS.

3. To avoid system table area availability problems, modify the system table area size to include the DNCS memory requirements as described in paragraph 1.7. It is recommended that the XSCU and MST SCI procedures be used to set the system table area size to MAX after the DNCS Nucleus installation is complete.

4. The DNCS CI command !RESTART FCCC does not download the FCCC firmware patches. Although the successful operation of DNCS does not require these patches, other comm products that may be sharing the board with DNCS, such as 3780, do require these patches. Do not try to run 3780 on a board that has been reset via !RESTART FCCC until TDNCS/XDNCS has been executed. XDNCS causes the FCCC to be reset and the firmware patches to be re-downloaded.

5. XDNCS issues a reset to the FCCC which clears all downloaded code on the board. It then downloads

patches to the board via CDL. This procedure may cause problems for other comm products sharing this FCCC unless you observe the following conventions.

a. Always start DNCS before starting any other comm package. This means that XDNCS must be executed and the DNCS job started. After DNCS is started, then execute the CDL and start up procedures for the other comm packages.

b. If you include XDNCS in .S$ISBTCH , the DNCS XBJ procedure may reset the board after CDL completion for other comm packages. The recovery procedure in this case is to re-execute the CDL procedures prior to starting the other comm packages.

6. The TDNCS procedure issues a reset to the FCCC which clears all downloaded code on the board and therefore may cause problems for other comm products sharing this FCCC. It is recommended that TDNCS be issued only after you have decided that other shared packages may also be terminated.

7. STR 10376 - Password protection of DNCS commands is available only thru a DNCS CI command. However, this specification is not permanent and must be re-entered whenever the system is rebooted. Optional patch 1750 in PATDC allows permanent enabling of password protection and specification of the password value.

8. STR 11645 - The number of DNCS buffer segments may need to be larger than 3 for some configurations. Optional patch 1695 in PATDC allows modifying the number of buffer segments.

9. The DNCS SCI command HOSTQST may be used to start queue servers for more than one application. If two or more queue servers are running simultaneously, a conflict may result preventing proper dequeue of a serviced request due to improper syncronization with a shared procedure. STR 14342 addresses this problem and post-release patch 2578 in PATDC is available through customer support. Unless you have this patch installed on your system, you cannot run concurrent host application queue servers.

10. When the DNCS SCI command QHOST is used to enter a host request, the related transaction is recorded in the host queue trace file. It has been observed on

occasions that when a transaction is entered, the send acknowledgement appears in the trace file but the expected response does not occur due to SVQ losing its LU connection. To correct this condition, enter the DNCS CI command !D <termname>. If under the display header BINDINFO the LU name is filled with asterisks, then it will be necessary to restart your SVQ terminal with the CI command !RESTART <termname>. After successful restart, your QHOST trace file should indicate terminal connection and it may be necessary to reenter your QHOST request. If you do not know your SVQ terminal name, issue the CI command !D TERMS. Find device type SVQ under the DVCTYP header. The corresponding terminal name can be found under the TERMNAME

11. When in the VDT2 emulator and executing DNCS CI commands, line 24 may be appended to the command if the ENTER key is pressed when the cursor is on line 24. This may cause the following message to be returned: DNCS0053 E INVALID USE OF _____. If this condition occurs, enter ESC+S and reenter the command.

12. STR14551. When the Service Queue is initialized and a Host Queue started, the Host Queue task may get hung in a loop if a STOP command is issued on the SVQ resource. Patch 2647 in PATDC fixes this problem.

13. STR14224. Patch 2607 in PATDC fixes the following incorrect responses from CI commands:

    a. DISPLAY of RFT circuit produces 'SYSGEN ERROR'.

    b. ASSIGN to SDLC circuit produces 'SYNTAX ERROR' even though the function was performed.

    c. START/STOP of SDLC circuit produces 'UNSUPPORTED FUNCTION' even though the function was performed.

2.2 DOCUMENTATION - DNCS OPERATIONS GUIDE

1. Paragraph 3.2.34: The Stop Port command cannot stop a port in the disconnected (DISCN) state. There are two ways to get a port out of DISCN state: 1)connect the physical circuit to the port, or 2) issue a Stop Board command and assign the circuit to the port to which it is physically connected.

SECTION 3

PATCHES AND PATCH PROCEDURES

3.1 PATCH UPDATE PROCEDURE

Patches are maintained by Texas Instruments and are available to customers from two sources - Customer Support Line and Patch Update Service. The Customer Support Line is able to provide patches on an as needed basis over the telephone or by communications link. Call (512)-250-7407 to get the latest patch files. Periodically, Texas Instruments will ship all current patches for the DNOS system family software to customers on the subscription service. Refer to the DNOS Products Patch Update Service Release Information for a list of the latest patches. In both cases, a detailed explanation will be provided on how to apply the patches to your system.

It is recommended that you call the Customer Support Line to get the latest patches prior to installation of the product.

3.2 APPLYING PATCHES RECEIVED AT A LATER DATE

When you receive patch updates, instructions are included in the update package which describe how to copy the patch files to your master media.

Prior to applying these patches to your DNCS system, it is necessary that DNCS and all supported packages be terminated. This includes the following:

1. DNCS System Generation Utility
2. DNCS Job
3. DNCS/SNA Emulators (XVDT2, XPTR1, XPTR3, XKSR, PSC) at all stations
4. DNCS Service Queue
5. DNCS X.25 RFT
6. DNCS 914A Terminals

An alternative approach to terminating DNCS and its packages is to IPL the system after applying the patches.