DNPG0.WS4 (= "DR Net Programmer's Guide", foreword)

DR Net
Programmer's Guide

First Edition: March 1984

(Retyped by Emmanuel ROCHE.)

Foreword

DR Net is a network operating system for the CP/M family of operating systems. Developed for use on any 8086- or 8088-based computer, DR Net allows application programs, without modification, to access remote disk drives, list devices, and queues, as though they were resident facilities. DR Net is also a functional extension of the CP/NET Version 1.2 network operating system for 8080-, 8085-, or Z-80-based systems. Because the two share many system functions, communication between CP/NET Version 1.2 requesters and DR Net servers is supported.

How to use this manual

This manual describes DR Net's interface with programs and processes. Before you use this manual, you should have a thorough understanding of the CP/M-86 and Concurrent CP/M system calls, and their calling conventions. It is not necessary, however, to understand DR Net's interface with the physical network hardware. The hardware interface is transparent to the program.

Section 1 describes how DR Net interacts with the local operating system to create network requesters and servers. Section 1 also explains how communication is managed between these two types of nodes.

Section 2 describes the functions that can be implemented on a remote basis, and presents the implementation notes for functions affected by the network environment.

Section 3 describes DR Net's treatment of local, remote, and network errors.

Section 4 describes the system calls unique to DR Net.

Appendix A lists all of the system calls, and summarize their input parameters and returned values. Appendix B describes the DR Net utilities. Appendix C lists the contents of each network message.

DR Net documentation set

The "DR Net User's Guide" explains the use of the DR Net utilities. This manual also includes an introduction to the computer networks, and a description of the network error messages and what they mean.

The "DR Net System Manager's Guide" describes DR Net configuration considerations, and the system installation utilities.

The "DR Net System Guide" describes DR Net's proprietary and hardware-dependent modules, and their interfaces to local processes, the resident operating system, and the network I/O hardware.

For information on the CP/NET Version 1.2 network operating system, see the "CP/NET Reference Manual".

Conventions used in this manual

The following conventions are used in this manual.

- DR Net utilities are spelled in uppercase. The CMD filetype is assumed after the first textual appearance of a filename.

- Throughout this manual, frequent mention is made to long and short pointers. A long pointer is a double-word value where the first word is the offset, and the second word is the segment address. Short pointers specify an offset.

- The word "map" indicates that a local disk drive, list device, or queue, is replaced by a remote, equivalent resource. For example, "drive B is mapped to drive C" means all system calls to drive B are actually implemented on drive C.

- The word "host" refers to the resident operating system in a network node, as in the expression "host operating system".

Table of Contents

Tables, Figures, and Listings

Tables

Figures

Listing

EOF

DNPG1.WS4 (= "DR Net Programmer's Guide", section 1)

DR Net
Programmer's Guide

First Edition: March 1984

(Retyped by Emmanuel ROCHE.)

Section 1: DR Net overview

DR Net consists of an invariant module and hardware-dependent Network I/O System (NIOS). The invariant module contains DR Net's interface to its host operating system, and the supporting code for the DR Net system calls. The NIOS contains the interface between the invariant module and the physical network hardware. The two modules are merged together with a system generation utility to create a single DR Net system module. After DR Net has been integrated with its host operating system, it does not affect how system calls are executed, only where they are executed. Furthermore, DR Net is transparent to the calling process.

DR Net's interface with its host operating system can be described as one of two functional roles: a network requester, or a network server. In a requester, the DR Net module is a system-call screener that intercepts disk drive, list device, and queue-related system calls, and initiates transactions over the network. The DR Net server is a process manager that oversees the creation and maintenance of shadow processes for every requester process logged on that server.

The section describes how DR Net screens system calls, and manages shadow processes.

Section 1.1, "DR Net nodes and messages", presents the fundamental information on DR Net node types, network topology, and messages.

Section 1.2, "Requester and server functions", describes the characteristics of, and functions provided by, these different types of network nodes.

Section 1.3, "Network environments", describes DR Net's system-call routing mechanism, and how it is manipulated.

1.1 DR Net nodes and messages

A DR Net network node is any computer system with at least one floppy-disk drive, the CP/M-86 or Concurrent CP/M operating system, and the DR Net network operating system. Network nodes can communicate with each other when the host operating system and DR Net are loaded, and DR Net is attached. DR Net defines two types of network nodes: requesters and servers.

Network transactions are conducted on a resource swapping basis. Each network requester has a resource map that defines which local disk drives, list devices, and queues are mapped to a server. When an application makes a system call to a mapped resource, DR Net traps the call, and sends it to the server indicated in the resource map table. The server then implements the call on the designated resource, and returns the response.

Internode communication is conducted by means of a fixed format message. All data required to execute the function remotely is contained in this message. DR Net uses the same message format to return the response to the calling application.

1.1.1 DR Net node types

DR Net distinguishes between two types of network nodes: requesters and servers. Under DR Net, only requesters can initiate a network transaction. Servers can only respond to requester's inquiries. Axiomatically, servers never surprise requesters; every message received by a requester is always a response to an earlier request.

Requester and server network functions are available fom the different Digital Research operating systems as follows:

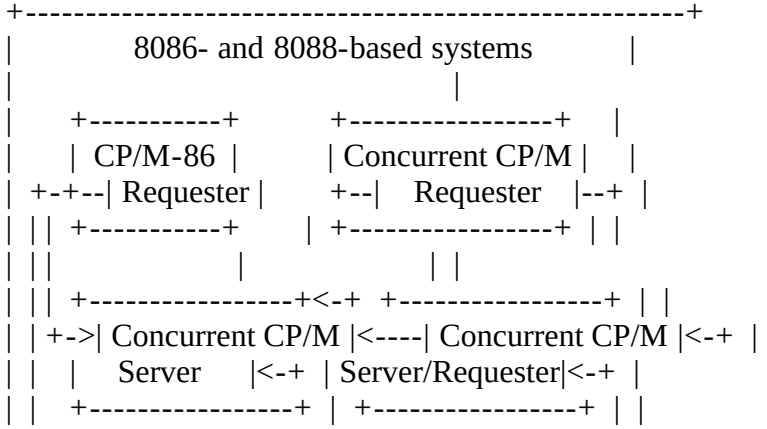
- Systems based on on CP/M-86 can function only as requesters on the network. They can talk to servers based on DR Net or MP/M II (CP/NET Version 1.2).

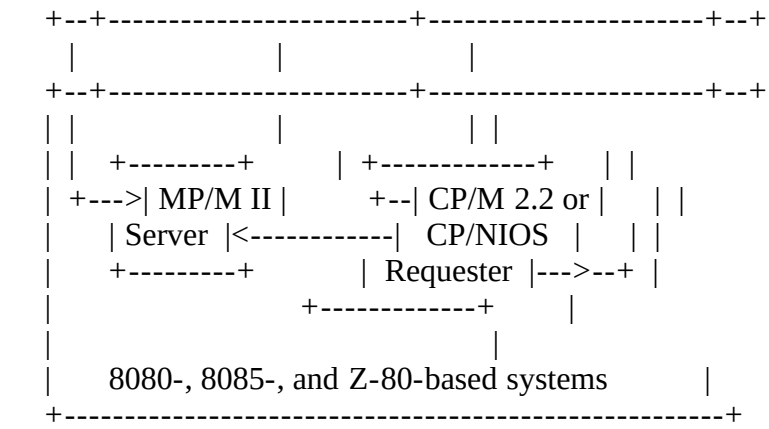
- Systems based on Concurrent CP/M can function as a network requester, a node server, or as both a network requester and server.

- Requesters based on CP/NET Version 1.2 can talk to servers based on DR Net.

Basic differences in these operating systems restrict the system calls supported over the network. Section 2 describes the limitations in detail.

Figure 1-1 illustrates the possible communication links between nodes running these operating systems. The arrows indicate who can send a request to whom.





(Arrows indicate who can send a request to whom.)

Figure 1-1. DR Net configuration options

Up to 255 computers can communicate to each other on a DR Net network. Each one is uniquely identified by a hexadecimal number in the range of 00 through FEh. Any number of computers can be configured as a network server or a network requester. For example, a network of computers under DR Net can consist of 254 requesters and one server, or 255 combination server/requester nodes.

There are limits, however, to the number of servers or requesters a computer can talk to at one time. A single requester node can support a maximum of 90 processes attached to the network simultaneously. Each of these processes can log on up to 16 server nodes at a time. Each server node can support up to 83 requester processes at a time.

1.1.2 DR Net message

Requester processes communicate with server processes by means of DR Net messages. Figure 1-2 illustrates its contents. See Appendix C for a detailed description of each field.

Field Length:

1 (*)	1 or 2	1 or 2	1	1 or 2	1 -- 65,536
Message Format Code	Destination Node & Process ID Number	Source Node & Process Number	System Call Number	Length of Data Field	Data Field: All information required to execute system call remotely

(*) Number indicates length of field. Where two lengths are shown, the first is for format 00 and 01 messages, and the second is for format 06 and 07 messages. (See Appendix C for a the description of the different DR Net message formats.)

Figure 1-2. Components of the DR Net message

The data field in the DR Net message is used for two purposes. For all requests, the messages contains the conformation required for the server to perform the function. For all responses, it contains the return value and related data. For example, the data field in a F_READ request contains the current user number and the File Control Block (FCB) of the file to be read. The data field in a F_READ response message contains the return code, the updated FCB, the data that was read, and network related information.

1.2 Requester and server functions

DR Net's interface with its host operating system can be described as either a system-call screener, or a process controller. Call screening is the requester function that determines if the disk drive, list device, or queue referenced by a system call is local or mapped to a remote system. Calls that reference a local resource are passed to the host operating system for execution. Calls that reference a remote resource are withheld and sent to the specified server node.

Process control is a server function. The DR Net server module creates and maintains a separate process for each requester process logged on. This "shadow" process exists only as long as the requester is logged on to the server, and idles when not in use. When a message is received from a requester, DR Net signals the associated shadow process to read the message, decode it, and present the function to the host operating system. From the perspective of the server's host operating system, shadow processes are like any other application process.

After the call is executed, the shadow process prepares the response message, and sends it back to the requester. DR Net's requester module then decodes the message, and returns the response to the calling process or program, according to local return convention.

Although systems based upon CP/M-86 and Concurrent CP/M can both function as DR Net requesters, DR Net's interface to a Concurrent CP/M host is different than its interface to a CP/M-86 host. This difference should not affect your application software. However, it may impact upon any custom applications you develop for use under DR Net.

1.2.1 Requester nodes based on CP/M-86

In a requester with a CP/M-86 node, the DR Net system image is contained in the file DRNET.CMD. This file has two logical parts: the Network I/O System (NIOS) and the Network Disk Operating System (NDOS). The NIOS is DR Net's interface to the network hardware. It is developed by a system implementer, and integrated with the NDOS for a CP/M-86 requester by the DR Net GENRQR.CMD system generation utility. The NDOS is DR Net's invariant module (that is, suitable for any network hardware) that interfaces to application programs and the host operating system. It is the NDOS that performs the system-call

screening and provides the support for the DR Net functions.

In systems based on CP/M-86, DR Net is loaded with the NETLDR.CMD utility. The DRNET.CMD system image file must be present on the disk when NETLDR is invoked. The following events occur when NETLDR is run:

- DRNET.CMD is read, and its constituent NIOS and NDOS modules are loaded into the top of available system memory.

- The memory region occupied by the NIOS and NDOS is removed from the MRT table.

- The pointer for INT 224 is changed from the BDOS entry point to the NDOS entry point.

- The network I/O line drivers are initialized.

Figure 1-3 illustrates system memory configuration after NETLDR has run. At this point, DR Net is attached, and the system-call screening function is implemented. However, the requester is not logged on any servers. The operator invokes the LOGON.CMD utility to log on servers. See Appendix A for a description of the NETLDR and LOGON utilities.

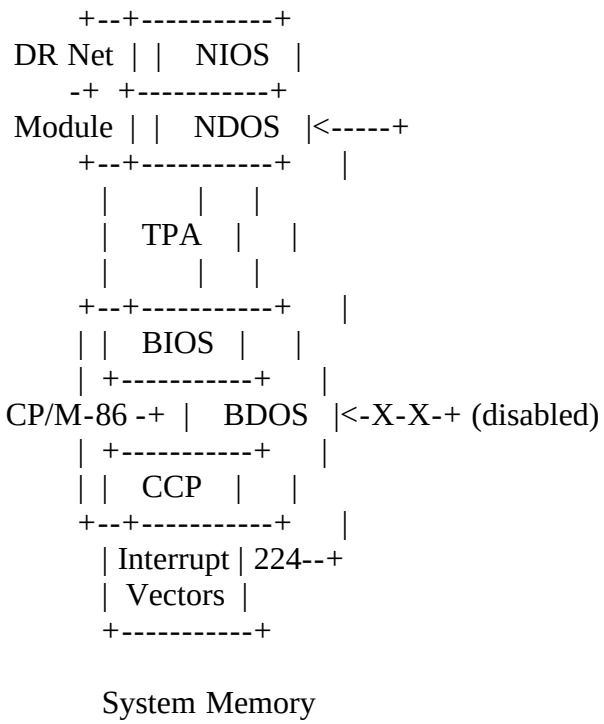


Figure 1-3. CP/M-86 requester memory organization

The NDOS traps each system call made by a program to determine who should execute the function -- the host system or a server. The first round of screening considers the resource referenced by the call. Unless a disk drive, list device, or queue is referenced, the NDOS passes the call on the local BDOS for execution.

When the NDOS encounters a call referencing a disk drive, list device, or queue, it looks to an internal data structure called the Requester

Configuration Table (RCT) to determine whether or not the resource is mapped to another system. All calls that reference local resources are passed on to the BDOS for execution. All calls to a resource mapped to a server are withheld from the BDOS, and sent to the server specified by an entry in the RCT. The Requester Configuration Table is described in Section 1.3, "Network environments".

Note: Normally, queue calls cannot be made by programs running under CP/M-86. However, the NDOS adds to the functions supported by CP/M-86 several functions otherwise only available from Concurrent CP/M. See "CP/M-86 functions extensions" below for the description of the additional functions supported.

All NDOS functions, from call screening to message output, are transparent to the calling program and the BDOS. The same is true for the NDOS's return of control to the calling program. When the response is received, the NDOS mimics the Concurrent CP/M return conventions for register contents and data storage before it relinquishes the CPU to the original calling process. Failures, whether resulting from a remote BDOS error or a network error, are also returned according to Concurrent CP/M conventions. See Section 3, "DR Net error handling", for the description of DR Net error reporting protocol.

CP/M-86 function extensions

Besides the network functions, the NDOS adds to a CP/M-86 requester the ability to perform several Concurrent CP/M system calls locally. This allows many programs written to run under Concurrent CP/M to run on requesters based on CP/M-86. The additional calls are as follows:

44: F_MULTISEC
45: F_ERRMODE
59: P_LOAD
152: F_PARSE
153: C_GET
154: S_SYSDAT
156: P_PDADR
164: L_GET

When the NDOS traps these calls, it executes them instead of passing them to the BDOS. (If they went to the BDOS, an error would result.) These additional functions are supported because the NDOS maintains an internal buffer that contains the requisite data from the process desciptor, User Data Area (UDA), and System Data Page.

The NDOS supports other functions not allowed under CP/M-86. In all cases, these are functions that reference files, drives, list devices, and queues (system calls with the prefixes F_, DRV_, L_, and Q_, respectively.) However, the resource referenced by these calls must be remote rather than local. For instance, an application can request the full range of queue functions, as long as the queue reference is resident on a remote server. See Table 2-4, "Available CP/M-86 functions", in Section 2 for the list of all calls supported locally and remotely by CP/M-86 requesters.

1.2.2 Nodes based on Concurrent CP/M

A computer running Concurrent CP/M can function as either a network requester or a network server, or as both a network requester and server. Which function is implemented is determined at DR Net system generation time, when you specify the number of shadow and requester processes. The DR Net GENNET system generation utility then selects the corresponding DR Net server, requester, or simultaneous requester/server resource file; reads the NIOS.CMD file; and constructs the DR Net system module. This module is stored on the disk under the filename DRNET.CMD.

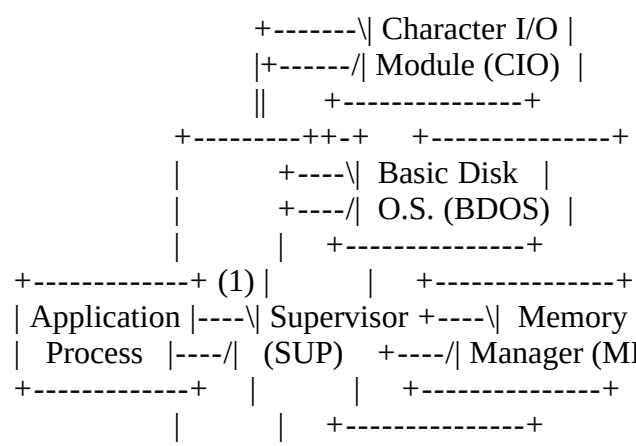
Unlike nodes with a CP/M-86 host, the DRNET.CMD file is merged with the Concurrent CP/M's CCPM.SYS system image file, rather than stored for use as a separate file. The DR Net ADDNET utility performs the integration of the two files, and records the results under the filename CCPM.SYS.

In server and simultaneous requester/server nodes, loading the CCPM.SYS also enables DR Net's server function. That is, the network I/O line drivers are initialized and the shadow process control mechanism is loaded. The local resources are immediately available to remote requesters.

In requester and requester/server nodes, loading the CCPM.SYS file does not in itself enable the requester call screening function. All calls made by a process are routed to the local SUP, BDOS, RTM, CIO, or MEM module. DR Net must be attached before the operator can use the computer as a network requester node. There are two ways to attach DR Net. An application can call DR Net's N_ATTACH function, or the operator can invoke the NETON.CMD utility. NETON makes a series of N_ATTACH calls.

Concurrent CP/M requester

The DR Net N_ATTACH function attaches a process to the network, and gives it a network environment. In addition, all child processes created by an attached process are automatically attached. Once attached, all system calls made by the process are subjected to DR Net's call-screening mechanism. System-call screening under Concurrent CP/M is implemented as shown in Figure 1-4.



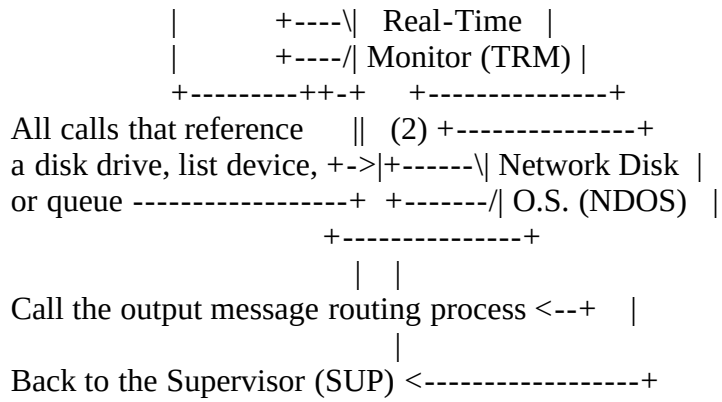


Figure 1-4. Concurrent CP/M function screening procedure

The two numbered locations ["(1)" and "(2)"] in Figure 1-4 indicate the decision points in the procedure that decides whether a system call references a remote or local resource.

1. The supervisor (SUP) determines whether the resource referenced is always local, or possibly mapped. Calls in the first category (for example, all memory (MC_) and flag (DEV_) related functions) are forwarded to the appropriate local module for execution. Calls in the second category (that is, all disk (DRV_), file (F_), list device (L_), and queue (Q_) related calls) are sent to the NDOS for further evaluation.

2. The NDOS checks the Requester Configuration Table (RCT) to determine whether or not the local resource referenced in the call is mapped to a remote system. If the resource referenced is not mapped, the call is sent back to the SUP for forwarding to the resident module. If the resource referenced is mapped, the NDOS makes the DR Net message, and initiates the network transaction.

The Requester Configuration Table (RCT) is the same for both CP/M-86 and Concurrent CP/M requesters. See Section 1.3, "Network environments", for the description of this data structure.

After the NDOS initiates a network transaction, it waits for the response message. When this message is received, the NDOS mimics the standard return conventions for register contents, data storage, and error handling before returning control to the calling process.

Concurrent CP/M server

The DR Net server is a process manager that creates local processes for each requester process that logs on. This is a shadow process because it remains allocated to a single requester process, and serves as its representative for presenting system calls to the server's host operating system. There is a limit to the number of shadow processes that can be supported in a server. This maximum value is a parameter elicited by the GENNET.CMD system generation program.

When a message is received, DR Net selects the shadow process assigned to the calling requester. The shadow process then decodes the DR Net message, and calls its Concurrent CP/M host. Before its call is made, however, all memory allocation, temporary data storage, and register initialization is performed according to the local calling convention. The SUP implements the call no differently than it would a call generated by a local process.

After the function is executed, the shadow process assembles the response message. This message includes the return value and all associated data, such as a data record, an FCB, or queue contents. Finally, the response is output to the network, to complet the transaction.

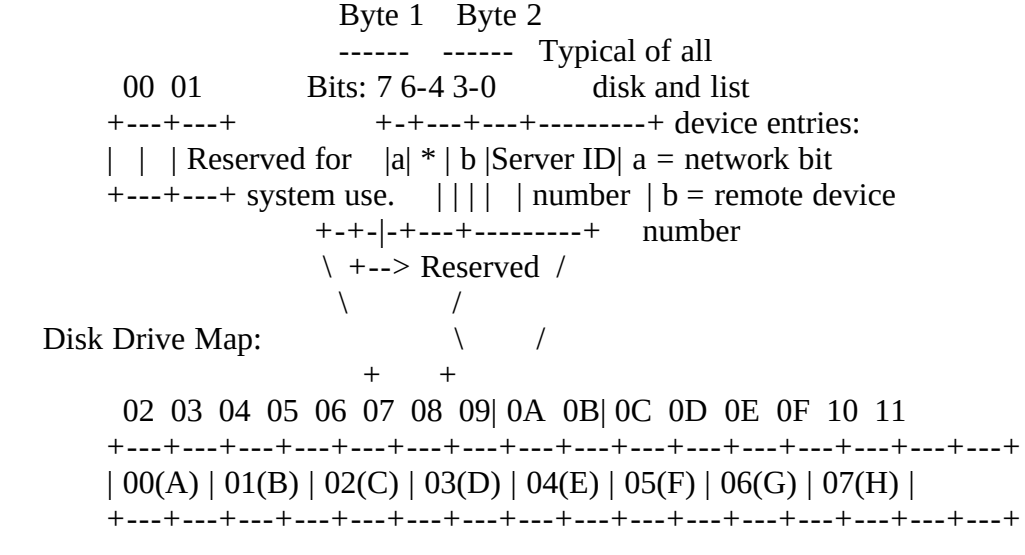
Shadow processes idle between messages. They remain assigned to their requester process until a log-off is received from the requester, or a network error occurs. See Section 3, "DR Net error handling", for the description of the errors that cause shadow processes to be terminated in a server.

1.3 Network environments

Each requester process attached to the network has a network environment. The environment consists of a requester Configuration Table (RCT) and a Log-on Table. The RCT indicates which local disk drives, list devices, and queues are mapped, and the server to which the remote resource is attached. The Log-on Table indicates which servers are currently logged on. These two data structures are illustrated in Figure 1-5.

Function 69, N_RCT, is used to get the Requester Configuration Table and the Log-on Table. Figure 1-5 illustrates the appearance of the two tables when this function is called. The Requester Configuration Table and Log-on Table are not contiguous in memory, however. They are maintained in separate portions of memory, and only assembled into a single structure for the sake of function 69.

(Numbers indicate hexadecimal offset from first byte in table.)



```
 12  13  14  15  16  17  18  19  1A  1B  1C  1D  1E  1F  20  21
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 08(I) | 09(J) | 0A(K) | 0B(L) | 0C(M) | 0D(N) | 0E(O) | 0F(P) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

 22  23  24  25  26  27  28  29  2A  2B  2C  2D  2E  2F  30  31
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                 Reserved                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

 32  33  34  35  36  37  38  39  3A  3B  3C  3D  3E  3F  40  41
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                 Reserved                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

List Device Map:

```
 42  43  44  45  46  47  48  49  4A  4B  4C  4D  4E  4F  50  51
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 00  | 01  | 02  | 03  | 04  | 05  | 06  | 07  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

 52  53  54  55  56  57  58  59  5A  5B  5C  5D  5E  5F  60  61
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 08  | 09  | 0A  | 0B  | 0C  | 0D  | OE  | 0F  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Queue Map: (First 16 separate entries)

```
 62  63  64  65  66  67  68  69  6A  6B  6C  6D  6E  6F  70  71
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| * | Local Queue Name            | Remote Queue Name       :
+-|-+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
  +--> Type Flag

 72  73  74  75
+---+---+---+---+
:   |   | Remote|
: * | * | Queue |
: | | | |   ID  |
+-|-+-|-+---+---+
  |   +--> Server ID
  +------> (Last character of Remote Queue Name.)
```

Log-on Table:

```
                    +---------+---------+
                    | Server  | Reserved| Typical
                    | Node ID | for sys-| Log-on Table
                    |         | tem use | entry
                    +---------+---------+
                     \            /
                      \          /
                       \        /
                        +      +
```

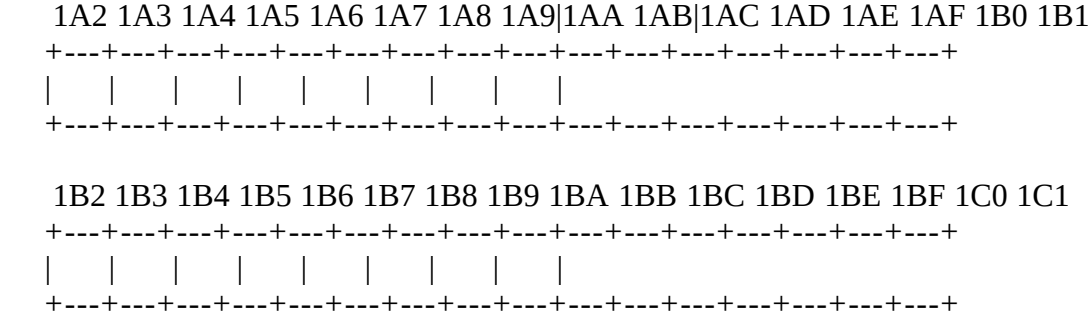


Figure 1-5. Requester Configuration Table

1.3.1 Requester configuration table

The Requester Configuration Table (RCT) is organized into the three resource maps and Log-on Table shown in Figure 1-5. Table 1-1 below describes the fields in each Requester Configuration Table map segment. The initial values for each field in the Requester Configuration Table are entered at DR Net system generation as responses to prompts from the system generation utility. This series of prompts creates the master Master Requester Configuration Table. Processes acquire copies of the master when they attach to the network. For more information on the network attach function and how it affects a process's network environment, see Section 1.3.3, "Network environments and process families".

Table 1-1. Requester Configuration Table field descriptions

Format: Map segment
Field description

Disk Drive Map
Table bytes 02h through 21h contain the mapping for local drives 00 (A) through 0F (P). Each map entry consists of two bytes, where the first entry contains the map for drive A, the second entry contains the map for drive B, and so forth through drive P.

The first byte of each entry has two significant bit fields.

- Bit 7 indicates whether the corresponding drive is mapped or local, as follows:

  bit 7 = 0: local
  bit 7 = 1: mapped

- Bits 3 through 0 contains the hex number of the remote drive.

The second byte of each field contains the hexadecimal ID number of the server node.

List Device Map
Table bytes 42 through 61 contains the mapping for local list devices LST0: through LST15:. This portion of the table is organized in the same fashion as

the drive map, where each entry consists of two bytes, and the position of entry within the range is significant. The two bytes are used in the same way as they are in the Disk Drive Map.

Queue Map
Table bytes 62 through 1A1 contain the map assignments for up to 16 local queues. Because queues are not sequentially labelled like drives and printers, the fields are more descriptive, as follows:

- The Type Flag byte replaces the network bit, to indicate whether the queue is remote or local. The corresponding values are as follows:

    00h: local
    01h: remote

- The Local Queue Name contains the eight-character ASCII name of the local queue to be mapped.

- The Remote Queue Name contains the eight-character ASCII name of the server-based, replacement queue.

- Server ID contains the hex ID number of the server node with the mapped queue.

- Remote Queue ID contains the short pointer into the System Data Page of the server-based, replacement queue.

Note: The queue length cannot exceed the maximum message buffer size when a queue is mapped over the network.

1.3.2 Log-on Table

Each process attached to the network has a Log-on Table to indicate which server nodes are accessible. The list consists of 10h (16) separate, two-byte fields, where the first byte contains the server ID number, and the second is reserved for system use. A null entry is indicated by an 0FFh in the first byte.

A process's Log-on Table is created when the process is attached to the network. If neither the process nor its parent was attached previously, the Log-on Table is blank. If the process was created by an already attached process, the child's Log-on Table is copied from its parent's current version. The child process does not need to log on all servers listed in the parent's table.

To register a server in the Log-on Table, a process calls function 64, N_LOGON. A log-on message is then sent to the specified server. The server's ID number is written only when the server's response indicates success. To erase a server from the table, the process calls the DR Net function 65, N_LOGOFF.

1.3.3 Network environments and process families

Process families and network environments are closely affiliated under DR Net. For example, once a process is attached to the network, every child it creates is also automatically attached. Unless specifically specified otherwise, the entire family shares a single copy of the requester Configuration Table.

A single copy of the Requester Configuration Table (RCT) is enough for most applications. However, individual processes or process subfamilies can obtain a separate RCT with the DR Net N_ATTACH function. This allows a process to make changes to its copy of the Requester Configuration Table without affecting its relatives' copy. The use of N_ATTACH to create new network environments is described below. Following this, the effect of the DR Net N_DETACH function upon the environment is described.

DR Net NETON and LOGON utilities

For many applications, the DR Net NETON and LOGON utilities suffice to attach and log-on processes. When NETON is invoked, it makes a series of N_ATTACH calls to attach the current process and all ancestral processes, until it finds a process that is already attached. All child processes created after NETON has been run are automatically attached by DR Net, and inherit the parent's environment.

The LOGON utility works in a similar fashion. When it is invoked, it calls the N_LOGON function for the current process and each ancestral process. It stops calling N_LOGON only when the process's parent pointer is null.

Creating separate environments with the N_ATTACH function

A process's network environment is created when the process is attached to the network. A process is attached to the network in one of two ways:

1) The process calls DR Net function 73: N_ATTACH, and specifies a null process descriptor input parameter. DR Net responds by attaching the calling process.

2) A process calls N_ATTACH, but specifies another process descriptor as an input parameter. DR Net does not attach the calling process in this case; instead, DR Net attaches the specified process.

Attaching a requester process to the network gives it a network environment consisting of a copy of the Requester Configuration Table and a Log-on Table. The contents of these tables are determined by the following rules:

- If the process is not attached and its parent is not attached (or there is no parent), it gets a copy of the master Requester Configuration Table and a blank Log-on Table.

- If the process is not attached but its parent is attached, it gets its parent's Requester Configuration Table and a copy of its parent's Log-on Table.

- If the process is already attached, it gets a new requester Configuration Table, which is a copy of its previous RCT, but its Log-on Table is unaffected.

Note: There is a limit to the number of Requester Configuration Tables that can be created with the N_ATTACH function. The value is set at system generation, and is stored in the Parameter Table. See the description of the DR Net N_PARATAB function in Section 4 for the description of the Parameter Table.

Effect of N_DETACH upon a process's network environment

The N_DETACH function DOES NOT necessarily terminate the calling process's conncetion to the network. This is only true when the calling process has a first generation copy of the master requester Configuration Table. Fro example, when a process references a second or third generation copy, N_DETACH replaces the current Requester Configuration Table with the previous generation copy. The process remains attached to the network, and its Log-on Table unaltered.

When the process has a first generation Requester Configuration Table, a N_DETACH does detach the network. Detaching involves the following actions:

- A N_LOGOFF is sent to each server listed in the Log-on Table.

- The Log-on Table is deallocated.

- The process's link to its Requester Configuration Table is cut. All references to disk drives, list devices, and queues are not screened, and go directly to the local system.

When a process detaches, its former copy of the Requester Configuration Table is not necessarily deallocated. Because other processes might be using this copy, it remains in existence until there is no chance that an active process could reference it.

Modifying a Requester Configuration Table

A process changes its environment using DR Net function 69: N_RCT. This functions serves a dual purpose. First, it allows you to retrieve the calling process's Requester Configuration Table and Log-on Table. Second, it allows you to change the map of disk drives, list devices, and queues. Changes made affect all processes that share that table.

Network environments example

The sequence of frames in Figure 1-6 illustrates what happens when processes attach and detach from the network, and how network environments are created and purged. The following symbols are used in this figure:

- Ln, where "n" is a number, indicates the copy number of the Log-on Table. Notice that each attached process has its own Log-on Table.

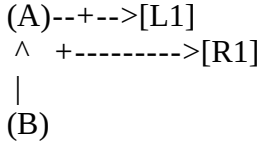
- Rn, where "n" is a number, indicates the copy number of the Requester Configuration Table.

In both cases, the copy is made from the parent's current table.

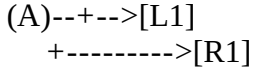
1) Process A, detached from network

2) Process A creates process B

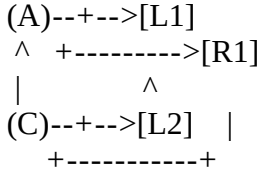
3) Process B attaches process A



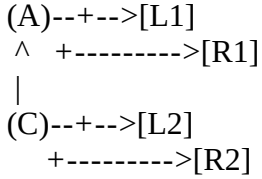
4) Process B terminates



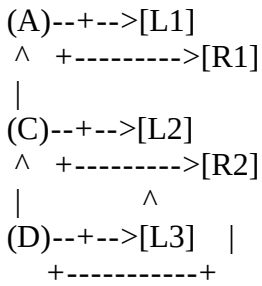
5) Process A creates process C



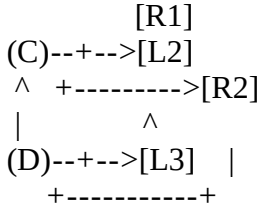
6) Process C attaches itself



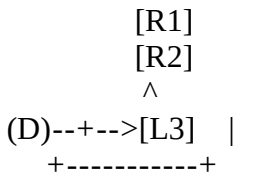
7) Process C creates process D



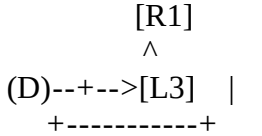
8) Process A terminates



9) Process C terminates



10) Process D detaches



11) Process D detaches a second time

Figure 1-6. Process attach and detach sequence

1. Process A has been created, but it is not attached to the network.

2. Process A creates process B. Neither is attached to the network.

3. Process B calls N_ATTACH to attach process A. Process B is NOT attached, but A now has a first generation compy of the master Requester Configuration Table (R1) and a Log-on Table (L1).

4. Process B terminates. This has no effect upon process A.

5. Process A creates process C. Process C is allocated a Log-on Table (L2

-- a direct copy of process A's L1), and shares process A's Requester Configuration Table (R1).

6. Process C calls N_ATTACH to attach itself. Process C's Log-on Table (L2) is unaffected. However, it gets a second generation copy of the Requester Configuration Table (R2 is a direct copy of process A's R1).

7. Process C creates process D. Process D gets its own Log-on Table (L3 is a direct copy of L2), and shares process C's Requester Configuration Table (R2).

8. Process A terminates. Its Log-on Table (L1) is deallocated, but the first generation of the Requester Configuration Table (R1) is not.

9. Process C terminates. Its Log-on Table (L2) is deallocated, but the second generation copy of the Requester Configuration Table (R2) is not.

10. Process D calls N_DETACH. Its log-on Table (L3) is unaffected, but the second generation Requester Configuration Table (R2) is deallocated. Process D is still attached to the network, and the first generation Requester Configuration Table (R1) defines its environment.

11. Process D makes a second call to N_DETACH. Because there is no previous generation Requester Configuration Table, the process is detached from the network, and its Log-on Table deallocated. Because there are no processes referencing copy R1 of the Requester Configuration Table, DR Net deallocates it.

EOF

DNPG2.WS4 (= "DR Net Programmer's Guide", section 2)

DR Net
Programmer's Guide

First Edition: March 1984

(Retyped by Emmanuel ROCHE.)

Section 2: System call support

The support of system calls by DR Net connotes two things. First, it means that the requester NDOS recognizes these functions and traps the call rather than pass it to the host operating system, forms the call into a message, and decodes the response message. Second, it means that the server module decodes the request and recognize the system call, presents the call to its host operating system according to the local system call calling convention, and assembles the response message.

Generally, the functions supported by DR Net for implementation over the network are all calls related to Concurrent CP/M disk drive, list device, and queue functions. In most cases, programs making calls to these resources can expect transparent execution of the function. The qualification is necessary, because DR Net's ability to transport the call does not guarantee that the remote system can execute it.

When all nodes on the network use Concurrent CP/M, all concerns for system call compatibility are irrelevant. However, compatibility considerations are very relevant in a hybrid network consisting of nodes with different operating systems. Tables 2-2 through 2-6 list all the Concurrent CP/M system calls, so you can determine beforehand whether a system call is supported locally or over the network. Each table is associated with a specific operating system, and answers the following questions for each function:

- Does this operating system support this function when the system is a server?

- Does the NDOS issue a request for this function under this operating system?

- Does the local system support this function when the system is a requester?

- If it does not, is the function harmless when it is called?

The last two questions are included because DR Net and CP/NET NDOS extend the functions available on the local operating system. For example, the DR Net NDOS, without its host operating system, supports functions F_MULTISEC, F_PARSE, C_GET, S_SYSDAT, P_PDADR, and L_GET.

2.1 System calls serviced

All Concurrent CP/M Release 3.1 functions are listed in Tables 2-2 through 2-6, as follows. Use these tables to determine whether the system calls made by your applications are serviced on requester or server nodes running on the different Digital Research operating systems.

- Table 2-2 describes calls available on CP/M Release 2.2 nodes.
- Table 2-3 describes calls available on CP/NOS nodes.
- Table 2-4 describes calls available on MP/M II nodes.
- Table 2-5 describes calls available on CP/M-86 nodes.
- Table 2-6 describes calls available on Concurrent CP/M Release 3.1 nodes.

Each table has four columns of "YES", "NO", and "NO*" responses to the fundamental questions described in the introduction to this section. The column headers represent the questions according to the definition shown in Table 2-1.

Table 2-1. Column header definitions

Format: Column header
Definitions

S -- Server
Can this operating system perform this function when the function is received over the network?

R -- Requester
Can the NDOS build a message from this function under this operating system?

L -- Local
Does this operating system support this call locally?

H -- Harmless
If the operating system does not support the function, does the NDOS intercept it and render the function harmless? The answers in this column are applicable only after the network has been attached.

The "Harmless" column is relevant only when the answer in the "Local" column is "NO". Otherwise, it is blank. In addition, the "Server" or "Requester" columns are eliminated when the host operating system does not support that network function.

These tables indicate whether or not a system call can be executed remotely. Confirm the calls made by your applications by referencing the functions in the table of your requester operating system, and in the table of your server

operating system. The requester table indicates the functions trapped and transported by the NDOS. The server table indicates the functions supported. A "YES" in both tables indicates that transparent, remote execution is provided.

"NO" in the "Requester" column indicates that the call cannot be made over the network. This does not mean programs cannot use this function. Instead, it indicates that the function is either meaningless in a network context, or the resource referenced is not mapped over the network. P_DELAY is an example of a function that is meaningless if executed remotely. All the memory allocation functions are examples of calls that reference resources that are always local.

A "NO" in the "Server" column indicates that the call is not supported by that host operating system. If the server receives a system call that it does not support, it returns the host system's error message, indicating an unsupported function was called.

A "NO*" in the "Requester" and "Server" columns indicates that the call is not supported, but the function is. In a requester, this means the Concurrent CP/M SUP module, or the NDOS, decomposes the call into a series of other calls. A "NO*" in the "Server" column indicates that the call is never received, but the function is available. Each call with a "NO*" is described in Section 2.2, "Implementation notes". See these descriptions for a listing of the system calls invoked for decomposed functions.

Tables 2-2 through 2-6 also indicate the effect of DR Net upon the functions available from the local operating system. As described in Section 1.2.1, "CP/M-86 requester nodes", the NDOS adds system calls that are not otherwise supported. For this purpose, reference the "Local" and "Harmless" columns in the table for the node's host operating system to determine if it is safe to make the call. "Harmless" indicates that the call results in a NOP.

Finally, these tables flag calls that have network repercussions. For all calls marked by an asterisk ("*"), there is a description in Section 2.2, "Implementation notes", of the effect the network has on the call, or the effect the call has on the network.

Table 2-2. CP/M Release 2.2 functions

Dec	Hex	Mnemonic	Description	R	L	H
* 0	00	P_TERMCPM	System Reset	NO	YES	
1	01	C_READ	Console Input	YES	YES	
2	02	C_WRITE	Console Output	YES	YES	
3	03	AUX_READ	Auxiliary Input	NO	YES	
4	04	AUX_WRITE	Auxiliary Output	NO	YES	
* 5	05	L_WRITE	List Output	YES	YES	
6	06	C_RAWIO	Direct Console I/O	YES	YES	
7	07	AXI_STAT	Auxiliary Input Status	NO	YES	
8	08	AXO_STAT	Auxiliary Output Status	NO	YES	
* 9	09	C_WRITESTR	Print String	NO*	YES	
* 10	0A	C_READSTR	Read Console Buffer	NO*	YES	
* 11	0B	C_STAT	Get Console Status	YES	YES	

```
* 12  0C  S_BDOSVER     Return BDOS Version Number  NO   YES
* 13  0D  DRV_ALLRESET  Reset Disk System          NO*  YES
  14  0E  DRV_SET       Select Disk               YES  YES
  15  0F  F_OPEN        Open File                 YES  YES
  16  10  F_CLOSE       Close File                YES  YES
  17  11  F_SFIRST      Search for First          YES  YES
  18  12  F_SNEXT       Search for Next           YES  YES
  19  13  F_DELETE      Delete File               YES  YES
  20  14  F_READ        Read Sequential           YES  YES
  21  15  F_WRITE       Write Sequential          YES  YES
  22  16  F_MAKE        Make File                 YES  YES
  23  17  F_RENAME      Rename File               YES  YES
  24  18  DRV_LOGINVEC  Return Log-in Vector      YES  YES
  25  19  DRV_GET       Return Current Disk       YES  YES
  26  1A  F_DMASET      Set DMA Offset            NO   YES
  27  1B  DRV_ALLOCVEC  Get Allocation Vector     YES  YES
  28  1C  DRV_SETRO     Write Protect Disk        YES  YES
  29  1D  DRV_ROVEC     Get Read/Only Vector      YES  YES
  30  1E  F_ATTRIB      Set File Attributes       YES  YES
  31  1F  DRV_DPB       Get Disk Param. Block Addr. YES  YES
* 32  20  F_USERNUM     Get/Set User Code         NO   YES
  33  21  F_READRAND    Read Random               YES  YES
  34  22  F_WRITERAND   Write Random              YES  YES
  35  23  F_SIZE        Compute File Size         YES  YES
  36  24  F_RANDREC     Set Random Record         YES  YES
* 37  25  DRV_RESET     Reset Drive               YES  YES
* 38  26  DRV_ACCESS    Access Drive              YES  NO   NO
  39  27  DRV_FREE      Free Drive                YES  NO   NO
  40  28  F_WRITEZF     Write Random with Zero Fill YES  YES
  41  29  F_TWREC       Test and Write Record     NO   NO   NO
  42  2A  F_LOCK        Lock record               YES  NO   NO
  43  2B  F_UNLOCK      Unlock Record             YES  NO   NO
  44  2C  F_MULTISEC    Set Multisector Count     NO   NO   NO
* 45  2D  F_ERRMODE     Set BDOS Error Mode       YES  NO   NO
  46  2E  DRV_SPACE     Get Free Disk Space       NO   NO   NO
  47  2F  P_CHAIN       Chain to Program          NO   NO   NO
  48  30  DRV_FLUSH     Flush Buffers             NO   NO   NO
  49  31  SCB_GETSET    Get/set CP/M Plus SCB     NO   NO   NO
  50  32  S_BIOS        Direct BIOS Call          NO   NO   NO
  51  33  F_DMAOFF      Set DMA Base              NO   NO   NO
  52  34  F_DMAGET      Get DMA Base              NO   NO   NO
  53  35  MC_MAX        Get Maximum Memory        NO   NO   NO
  54  36  MC_ABSMAX     Get Absolute Memory       NO   NO   NO
  55  37  MC_ALLOC      Allocate Memory           NO   NO   NO
  56  38  MC_ALLOCABS   Allocate Absolute Memory  NO   NO   NO
  57  39  MC_FREE       Free memory               NO   NO   NO
  58  3A  MC_ALLFREE    Free All Memory           NO   NO   NO
  59  3B  P_LOAD        Program Load              NO   NO   NO
  60  3C  RSX_CALL      Call Resident System Ext.  NO   NO   NO
  61  3D  ?
  62  3E  ?
  63  3F  ?
  64  40  N_LOGON       Log On a Server           YES  NO   NO
  65  41  N_LOGOFF      Log Off a Server          YES  NO   NO
```

66	42	N_SENDMSG	Send a Message	YES	NO	NO
67	43	N_GETMSG	Get a Message	YES	NO	NO
68	44	N_STAT	Get Network Status	YES	NO	NO
69	45	N_RCT	Get Req. Config. Table	YES	NO	NO
70	46	N_ATTRIB	Set Comp. Attributes	YES	NO	NO
71	47	N_SCT	Get Ser. Config. Table	YES	NO	NO
72	48	N_ERRMODE	Set Network Error Mode	NO	NO	NO
73	49	N_ATTACH	Attach Network	NO	NO	NO
74	4A	N_DETACH	Detach Network	NO	NO	NO
75	4B	N_BUFSIZ	Set Message Buffer Size	?		
76	4C	?				
77	4D	N_PARATAB	Get Parameter Table	NO	NO	NO
99	63	F_TRUNCATE	Truncate File	NO	NO	NO
100	64	DRV_SETLABEL	Set Directory Label	NO	NO	NO
101	65	DRV_GETLABEL	Return Directory Label	NO	NO	NO
102	66	F_TIMEDATE	Read File XFCB	NO	NO	NO
103	67	F_WRITEXFCB	Write File XFCB	NO	NO	NO
104	68	T_SET	Set Date and Time	NO	NO	NO
105	69	T_GET	Get Date and Time	NO	NO	NO
*106	6A	F_PASSWD	Set Default Password	YES	NO	NO
107	6B	S_SERIAL	Return Serial Number	NO	NO	NO
108	6C	?				
109	6D	C_MODE	Get/Set Console Mode	NO	NO	NO
110	6E	C_DELIMIT	Get/Set Delimiter	NO	NO	NO
111	6F	C_WRITEBLK	Write Console Block	NO	NO	NO
112	70	L_WRITEBLK	Write List Block	NO	NO	NO
128	80	M_ALLOC	Allocate Memory Segment	NO	NO	NO
129	81	M_ALLOC	Allocate Memory Segment	NO	NO	NO
130	82	M_FREE	Free Memory Segment	NO	NO	NO
131	83	DEV_POLL	Poll Device	NO	NO	NO
132	84	DEV_WAITFLAG	Flag Wait	NO	NO	NO
133	85	DEV_SETFLAG	Flag Set	NO	NO	NO
134	86	Q_MAKE	Make Queue	NO	NO	NO
135	87	Q_OPEN	Open Queue	NO	NO	NO
136	88	Q_DELETE	Delete Queue	NO	NO	NO
137	89	Q_READ	Read Queue	NO	NO	NO
138	8A	Q_CREAD	Conditionally Read Queue	NO	NO	NO
139	8B	Q_WRITE	Write Queue	NO	NO	NO
140	8C	Q_CWRITE	Conditionally Write Queue	NO	NO	NO
141	8D	P_DELAY	Delay Process	NO	NO	NO
142	8E	P_DISPATCH	Dispatch Process	NO	NO	NO
143	8F	P_TERM	Terminate Calling Process	NO	NO	NO
144	90	P_CREATE	Create Process	NO	NO	NO
145	91	P_PRIORITY	Set Priority of Process	NO	NO	NO
146	92	C_ATTACH	Attach Console	NO	NO	NO
147	93	C_DETACH	Detach Console	NO	NO	NO
148	94	C_SET	Set Console	NO	NO	NO
149	95	C_ASSIGN	Assign Console	NO	NO	NO
150	96	P_CLI	Call Command Line Interp.	NO	NO	NO
151	97	P_RPL	Call Resident Procedure Lib.	NO	NO	NO
152	98	F_PARSE	Parse Filename	NO	NO	NO
153	99	C_GET	Get Console	NO	NO	NO

154	9A	S_SYSDAT	Get SYSDAT Address	NO	NO	NO
155	9B	T_SECONDS	Get Date and Time	NO	NO	NO
156	9C	P_PDADR	Return Proc. Desc. Address	NO	NO	NO
157	9D	P_ABORT	Abort Specified Process	NO	NO	NO
158	9E	L_ATTACH	Attach List Device	NO	NO	NO
159	9F	L_DETACH	Detach List Device	NO	NO	NO
160	A0	L_SET	Set List Device	NO	NO	NO
161	A1	L_CATTACH	Conditionally Attach List	NO	NO	NO
162	A2	C_CATTACH	Conditionally Attach Con.	NO	NO	NO
163	A3	S_OSVER	Return Version Number	NO	NO	NO
164	A4	L_GET	Get List Device Number	NO	NO	NO

Table 2-3. CP/NOS functions

	Dec	Hex	Mnemonic	Description	R	L	H
*	0	00	P_TERMCPM	System Reset	NO	YES	
	1	01	C_READ	Console Input	YES	YES	
	2	02	C_WRITE	Console Output	YES	YES	
	3	03	AUX_READ	Auxiliary Input	NO	YES	
	4	04	AUX_WRITE	Auxiliary Output	NO	YES	
*	5	05	L_WRITE	List Output	YES	YES	
	6	06	C_RAWIO	Direct Console I/O	YES	YES	
	7	07	AXI_STAT	Auxiliary Input Status	NO	YES	
	8	08	AXO_STAT	Auxiliary Output Status	NO	YES	
*	9	09	C_WRITESTR	Print String	NO*	YES	
*	10	0A	C_READSTR	Read Console Buffer	NO*	YES	
*	11	0B	C_STAT	Get Console Status	YES	YES	
*	12	0C	S_BDOSVER	Return BDOS Version Number	NO	YES	
*	13	0D	DRV_ALLRESET	Reset Disk System	NO*	YES	
	14	0E	DRV_SET	Select Disk	YES	NO	NO
	15	0F	F_OPEN	Open File	YES	NO	NO
	16	10	F_CLOSE	Close File	YES	NO	NO
	17	11	F_SFIRST	Search for First	YES	NO	NO
	18	12	F_SNEXT	Search for Next	YES	NO	NO
	19	13	F_DELETE	Delete File	YES	NO	NO
	20	14	F_READ	Read Sequential	YES	NO	NO
	21	15	F_WRITE	Write Sequential	YES	NO	NO
	22	16	F_MAKE	Make File	YES	NO	NO
	23	17	F_RENAME	Rename File	YES	NO	NO
	24	18	DRV_LOGINVEC	Return Log-in Vector	YES	NO	NO
	25	19	DRV_GET	Return Current Disk	NO	NO	NO
	26	1A	F_DMASET	Set DMA Offset	NO	YES	
	27	1B	DRV_ALLOCVEC	Get Allocation Vector	YES	NO	NO
	28	1C	DRV_SETRO	Write Protect Disk	YES	NO	NO
	29	1D	DRV_ROVEC	Get Read/Only Vector	YES	NO	NO
	30	1E	F_ATTRIB	Set File Attributes	YES	NO	NO
	31	1F	DRV_DPB	Get Disk Param. Block Addr.	YES	NO	NO
*	32	20	F_USERNUM	Get/Set User Code	NO	YES	
	33	21	F_READRAND	Read Random	YES	NO	NO
	34	22	F_WRITERAND	Write Random	YES	NO	NO
	35	23	F_SIZE	Compute File Size	YES	NO	NO
	36	24	F_RANDREC	Set Random Record	YES	NO	NO

```
* 37  25  DRV_RESET     Reset Drive               YES  NO   YES
* 38  26  DRV_ACCESS    Access Drive              YES  NO   NO
  39  27  DRV_FREE      Free Drive                YES  NO   YES
  40  28  F_WRITEZF     Write Random with Zero Fill YES  NO   NO
  41  29  F_TWREC       Test and Write Record      NO   NO   NO
  42  2A  F_LOCK        Lock record               YES  NO   NO
  43  2B  F_UNLOCK      Unlock Record             YES  NO   NO
  44  2C  F_MULTISEC    Set Multisector Count      NO   NO   NO
* 45  2D  F_ERRMODE     Set BDOS Error Mode       YES  NO   NO
  46  2E  DRV_SPACE     Get Free Disk Space        NO   NO   NO
  47  2F  P_CHAIN       Chain to Program           NO   NO   NO
  48  30  DRV_FLUSH     Flush Buffers              NO   NO   NO
  49  31  SCB_GETSET    Get/set CP/M Plus SCB      NO   NO   NO
  50  32  S_BIOS        Direct BIOS Call           NO   NO   NO
  51  33  F_DMAOFF      Set DMA Base               NO   NO   NO
  52  34  F_DMAGET      Get DMA Base               NO   NO   NO
  53  35  MC_MAX        Get Maximum Memory         NO   NO   NO
  54  36  MC_ABSMAX     Get Absolute Memory        NO   NO   NO
  55  37  MC_ALLOC      Allocate Memory            NO   NO   NO
  56  38  MC_ALLOCABS   Allocate Absolute Memory   NO   NO   NO
  57  39  MC_FREE       Free memory                NO   NO   NO
  58  3A  MC_ALLFREE    Free All Memory            NO   NO   NO
  59  3B  P_LOAD        Program Load               NO   NO   NO
  60  3C  RSX_CALL      Call Resident System Ext.  NO   NO   NO
  61  3D  ?
  62  3E  ?
  63  3F  ?
  64  40  N_LOGON       Log On a Server            YES  NO   NO
  65  41  N_LOGOFF      Log Off a Server           YES  NO   NO
  66  42  N_SENDMSG     Send a Message             YES  NO   NO
  67  43  N_GETMSG      Get a Message              YES  NO   NO
  68  44  N_STAT        Get Network Status         YES  NO   NO
  69  45  N_RCT         Get Req. Config. Table     YES  NO   NO
  70  46  N_ATTRIB      Set Comp. Attributes       YES  NO   NO
  71  47  N_SCT         Get Ser. Config. Table     YES  NO   NO
  72  48  N_ERRMODE     Set Network Error Mode     NO   NO   NO
  73  49  N_ATTACH      Attach Network             NO   NO   NO
  74  4A  N_DETACH      Detach Network             NO   NO   NO
  75  4B  N_BUFSIZ      Set Message Buffer Size    ?
  76  4C  ?
  77  4D  N_PARATAB     Get Parameter Table        NO   NO   NO

  99  63  F_TRUNCATE    Truncate File              NO   NO   NO
 100  64  DRV_SETLABEL  Set Directory Label        NO   NO   NO
 101  65  DRV_GETLABEL  Return Directory Label     NO   NO   NO
 102  66  F_TIMEDATE    Read File XFCB             NO   NO   NO
 103  67  F_WRITEXFCB   Write File XFCB            NO   NO   NO
 104  68  T_SET         Set Date and Time          NO   NO   NO
 105  69  T_GET         Get Date and Time          NO   NO   NO
*106  6A  F_PASSWD      Set Default Password       YES  NO   NO
 107  6B  S_SERIAL      Return Serial Number       NO   NO   NO
 108  6C  ?
 109  6D  C_MODE        Get/Set Console Mode       NO   NO   NO
 110  6E  C_DELIMIT     Get/Set Delimiter          NO   NO   NO
```

```
111  6F  C_WRITEBLK    Write Console Block        NO   NO   NO
112  70  L_WRITEBLK    Write List Block           NO   NO   NO

128  80  M_ALLOC       Allocate Memory Segment    NO   NO   NO
129  81  M_ALLOC       Allocate Memory Segment    NO   NO   NO
130  82  M_FREE        Free Memory Segment        NO   NO   NO
131  83  DEV_POLL      Poll Device                NO   NO   NO
132  84  DEV_WAITFLAG  Flag Wait                  NO   NO   NO
133  85  DEV_SETFLAG   Flag Set                   NO   NO   NO
134  86  Q_MAKE        Make Queue                 NO   NO   NO
135  87  Q_OPEN        Open Queue                 NO   NO   NO
136  88  Q_DELETE      Delete Queue               NO   NO   NO
137  89  Q_READ        Read Queue                 NO   NO   NO
138  8A  Q_CREAD       Conditionally Read Queue   NO   NO   NO
139  8B  Q_WRITE       Write Queue                NO   NO   NO
140  8C  Q_CWRITE      Conditionally Write Queue  NO   NO   NO
141  8D  P_DELAY       Delay Process              NO   NO   NO
142  8E  P_DISPATCH    Dispatch Process           NO   NO   NO
143  8F  P_TERM        Terminate Calling Process  NO   NO   NO
144  90  P_CREATE      Create Process             NO   NO   NO
145  91  P_PRIORITY    Set Priority of Process    NO   NO   NO
146  92  C_ATTACH      Attach Console             NO   NO   NO
147  93  C_DETACH      Detach Console             NO   NO   NO
148  94  C_SET         Set Console                NO   NO   NO
149  95  C_ASSIGN      Assign Console             NO   NO   NO
150  96  P_CLI         Call Command Line Interp.  NO   NO   NO
151  97  P_RPL         Call Resident Procedure Lib.NO  NO   NO
152  98  F_PARSE       Parse Filename             NO   NO   NO
153  99  C_GET         Get Console                NO   NO   NO
154  9A  S_SYSDAT      Get SYSDAT Address         NO   NO   NO
155  9B  T_SECONDS     Get Date and Time          NO   NO   NO
156  9C  P_PDADR       Return Proc. Desc. Address NO   NO   NO
157  9D  P_ABORT       Abort Specified Process    NO   NO   NO
158  9E  L_ATTACH      Attach List Device         NO   NO   NO
159  9F  L_DETACH      Detach List Device         NO   NO   NO
160  A0  L_SET         Set List Device            NO   NO   NO
161  A1  L_CATTACH     Conditionally Attach List  NO   NO   NO
162  A2  C_CATTACH     Conditionally Attach Con.  NO   NO   NO
163  A3  S_OSVER       Return Version Number      NO   NO   NO
164  A4  L_GET         Get List Device Number     NO   NO   NO
```

Table 2-4. MP/M II functions

```
Dec Hex  Mnemonic     Description              S    L    H
--- ---  ---------    -----------              ---  ---  ---
  0  00  P_TERMCPM    System Reset              NO   YES
  1  01  C_READ       Console Input             NO   YES
  2  02  C_WRITE      Console Output            NO   YES
  3  03  AUX_READ     Auxiliary Input           YES  YES
  4  04  AUX_WRITE    Auxiliary Output          YES  YES
* 5  05  L_WRITE      List Output               YES  YES
  6  06  C_RAWIO      Direct Console I/O        NO   NO   YES
  7  07  AXI_STAT     Auxiliary Input Status    NO   NO   YES
```

```
  8  08  AXO_STAT     Auxiliary Output Status     NO   YES
* 9  09  C_WRITESTR    Print String              NO*  YES
* 10  0A  C_READSTR     Read Console Buffer       NO*  YES
* 11  0B  C_STAT       Get Console Status        YES  YES
* 12  0C  S_BDOSVER     Return BDOS Version Number  NO   YES
* 13  0D  DRV_ALLRESET  Reset Disk System         NO*  YES
  14  0E  DRV_SET      Select Disk               YES  YES
  15  0F  F_OPEN       Open File                 YES  YES
  16  10  F_CLOSE      Close File                YES  YES
  17  11  F_SFIRST     Search for First          YES  YES
  18  12  F_SNEXT      Search for Next           YES  YES
  19  13  F_DELETE     Delete File               YES  YES
  20  14  F_READ       Read Sequential           YES  YES
  21  15  F_WRITE      Write Sequential          YES  YES
  22  16  F_MAKE       Make File                 YES  YES
  23  17  F_RENAME     Rename File               YES  YES
  24  18  DRV_LOGINVEC  Return Log-in Vector      YES  YES
  25  19  DRV_GET      Return Current Disk       YES  YES
  26  1A  F_DMASET     Set DMA Offset            NO   YES
  27  1B  DRV_ALLOCVEC  Get Allocation Vector     YES  YES
  28  1C  DRV_SETRO     Write Protect Disk        YES  YES
  29  1D  DRV_ROVEC     Get Read/Only Vector      YES  YES
  30  1E  F_ATTRIB      Set File Attributes       YES  YES
  31  1F  DRV_DPB       Get Disk Param. Block Addr. YES  YES
* 32  20  F_USERNUM     Get/Set User Code         NO   YES
  33  21  F_READRAND    Read Random               YES  YES
  34  22  F_WRITERAND   Write Random              YES  YES
  35  23  F_SIZE       Compute File Size         YES  YES
  36  24  F_RANDREC     Set Random Record         YES  YES
* 37  25  DRV_RESET     Reset Drive               YES  YES
* 38  26  DRV_ACCESS    Access Drive              YES  YES
  39  27  DRV_FREE      Free Drive                YES  YES
  40  28  F_WRITEZF     Write Random with Zero Fill YES  YES
  41  29  F_TWREC       Test and Write Record     NO   NO   NO
  42  2A  F_LOCK       Lock record               YES  YES
  43  2B  F_UNLOCK      Unlock Record             YES  YES
  44  2C  F_MULTISEC    Set Multisector Count     NO   YES
* 45  2D  F_ERRMODE     Set BDOS Error Mode       NO   YES
  46  2E  DRV_SPACE     Get Free Disk Space       NO   YES
  47  2F  P_CHAIN       Chain to Program          NO   YES
  48  30  DRV_FLUSH     Flush Buffers             NO   YES
  49  31  SCB_GETSET    Get/set CP/M Plus SCB     NO   NO   NO
  50  32  S_BIOS       Direct BIOS Call          NO   NO   NO
  51  33  F_DMAOFF      Set DMA Base              NO   NO   NO
  52  34  F_DMAGET      Get DMA Base              NO   NO   NO
  53  35  MC_MAX       Get Maximum Memory        NO   NO   NO
  54  36  MC_ABSMAX     Get Absolute Memory       NO   NO   NO
  55  37  MC_ALLOC      Allocate Memory           NO   NO   NO
  56  38  MC_ALLOCABS   Allocate Absolute Memory   NO   NO   NO
  57  39  MC_FREE       Free memory               NO   NO   NO
  58  3A  MC_ALLFREE    Free All Memory           NO   NO   NO
* 59  3B  P_LOAD       Program Load              NO*  NO   NO
  60  3C  RSX_CALL      Call Resident System Ext.  NO   NO   NO
  61  3D  ?
```

62	3E	?			
63	3F	?			
64	40	N_LOGON	Log On a Server	YES	NO NO
65	41	N_LOGOFF	Log Off a Server	YES	NO NO
66	42	N_SENDMSG	Send a Message	YES	NO NO
67	43	N_GETMSG	Get a Message	YES	NO NO
68	44	N_STAT	Get Network Status	NO	NO NO
69	45	N_RCT	Get Req. Config. Table	NO	NO NO
70	46	N_ATTRIB	Set Comp. Attributes	YES	NO NO
71	47	N_SCT	Get Ser. Config. Table	YES	NO NO
72	48	N_ERRMODE	Set Network Error Mode	NO	NO NO
73	49	N_ATTACH	Attach Network	NO	NO NO
74	4A	N_DETACH	Detach Network	NO	NO NO
75	4B	N_BUFSIZ	Set Message Buffer Size	?	
76	4C	?			
77	4D	N_PARATAB	Get Parameter Table	NO	NO NO
99	63	F_TRUNCATE	Truncate File	NO	NO NO
100	64	DRV_SETLABEL	Set Directory Label	NO	YES
101	65	DRV_GETLABEL	Return Directory Label	NO	YES
102	66	F_TIMEDATE	Read File XFCB	NO	YES
103	67	F_WRITEXFCB	Write File XFCB	NO	YES
104	68	T_SET	Set Date and Time	NO	YES
105	69	T_GET	Get Date and Time	NO	YES
106	6A	F_PASSWD	Set Default Password	YES	YES
107	6B	S_SERIAL	Return Serial Number	NO	YES
108	6C	?			
109	6D	C_MODE	Get/Set Console Mode	NO	NO NO
110	6E	C_DELIMIT	Get/Set Delimiter	NO	NO NO
111	6F	C_WRITEBLK	Write Console Block	NO	NO NO
112	70	L_WRITEBLK	Write List Block	NO	NO NO
128	80	M_ALLOC	Allocate Memory Segment	NO	YES
129	81	M_ALLOC	Allocate Memory Segment	NO	YES
130	82	M_FREE	Free Memory Segment	NO	YES
131	83	DEV_POLL	Poll Device	NO	YES
132	84	DEV_WAITFLAG	Flag Wait	NO	YES
133	85	DEV_SETFLAG	Flag Set	NO	YES
134	86	Q_MAKE	Make Queue	NO	YES
135	87	Q_OPEN	Open Queue	NO	YES
136	88	Q_DELETE	Delete Queue	NO	YES
137	89	Q_READ	Read Queue	NO	YES
138	8A	Q_CREAD	Conditionally Read Queue	NO	YES
139	8B	Q_WRITE	Write Queue	NO	YES
140	8C	Q_CWRITE	Conditionally Write Queue	NO	YES
141	8D	P_DELAY	Delay Process	NO	YES
142	8E	P_DISPATCH	Dispatch Process	NO	YES
143	8F	P_TERM	Terminate Calling Process	NO	YES
144	90	P_CREATE	Create Process	NO	YES
145	91	P_PRIORITY	Set Priority of Process	NO	YES
146	92	C_ATTACH	Attach Console	NO	YES
147	93	C_DETACH	Detach Console	NO	YES
148	94	C_SET	Set Console	NO	YES
149	95	C_ASSIGN	Assign Console	NO	YES

```
 150  96  P_CLI        Call Command Line Interp.   NO   YES
 151  97  P_RPL        Call Resident Procedure Lib.NO   YES
 152  98  F_PARSE      Parse Filename              NO   YES
 153  99  C_GET        Get Console                 NO   YES
 154  9A  S_SYSDAT     Get SYSDAT Address          NO   YES
 155  9B  T_SECONDS    Get Date and Time           NO   YES
 156  9C  P_PDADR      Return Proc. Desc. Address  NO   YES
 157  9D  P_ABORT      Abort Specified Process     NO   YES
 158  9E  L_ATTACH     Attach List Device          NO   YES
 159  9F  L_DETACH     Detach List Device          NO   YES
 160  A0  L_SET        Set List Device             NO   YES
 161  A1  L_CATTACH    Conditionally Attach List   NO   YES
 162  A2  C_CATTACH    Conditionally Attach Con.   NO   YES
 163  A3  S_OSVER      Return Version Number       NO   YES
 164  A4  L_GET        Get List Device Number      NO   YES
```

Table 2-5. CP/M-86 functions

```
 Dec Hex  Mnemonic     Description                R    L    H
 --- ---  ---------    -----------                --- --- ---
*  0  00  P_TERMCPM    System Reset                NO   YES
   1  01  C_READ       Console Input               NO   YES
   2  02  C_WRITE      Console Output              NO   YES
   3  03  AUX_READ     Auxiliary Input             NO   NO   NO
   4  04  AUX_WRITE    Auxiliary Output            NO   NO   NO
*  5  05  L_WRITE      List Output                 YES  YES
   6  06  C_RAWIO      Direct Console I/O          NO   YES
   7  07  AXI_STAT     Auxiliary Input Status      NO   NO   NO
   8  08  AXO_STAT     Auxiliary Output Status     NO   NO   NO
   9  09  C_WRITESTR   Print String                NO   YES
  10  0A  C_READSTR    Read Console Buffer         NO   YES
  11  0B  C_STAT       Get Console Status          NO   YES
* 12  0C  S_BDOSVER    Return BDOS Version Number  NO   YES
* 13  0D  DRV_ALLRESET Reset Disk System           NO*  YES
  14  0E  DRV_SET      Select Disk                 YES  YES
  15  0F  F_OPEN       Open File                   YES  YES
  16  10  F_CLOSE      Close File                  YES  YES
  17  11  F_SFIRST     Search for First            YES  YES
  18  12  F_SNEXT      Search for Next             YES  YES
  19  13  F_DELETE     Delete File                 YES  YES
  20  14  F_READ       Read Sequential             YES  YES
  21  15  F_WRITE      Write Sequential            YES  YES
  22  16  F_MAKE       Make File                   YES  YES
  23  17  F_RENAME     Rename File                 YES  YES
  24  18  DRV_LOGINVEC Return Log-in Vector        YES  YES
  25  19  DRV_GET      Return Current Disk         YES  YES
  26  1A  F_DMASET     Set DMA Offset              NO   YES
  27  1B  DRV_ALLOCVEC Get Allocation Vector       YES  YES
  28  1C  DRV_SETRO    Write Protect Disk          YES  YES
  29  1D  DRV_ROVEC    Get Read/Only Vector        YES  YES
  30  1E  F_ATTRIB     Set File Attributes         YES  YES
  31  1F  DRV_DPB      Get Disk Param. Block Addr. YES  YES
* 32  20  F_USERNUM    Get/Set User Code           NO   YES
```

```
  33  21  F_READRAND    Read Random              YES  YES
  34  22  F_WRITERAND   Write Random             YES  YES
  35  23  F_SIZE        Compute File Size        YES  YES
  36  24  F_RANDREC     Set Random Record        YES  YES
* 37  25  DRV_RESET     Reset Drive              YES  YES
* 38  26  DRV_ACCESS    Access Drive             YES  YES
  39  27  DRV_FREE      Free Drive               YES  YES
  40  28  F_WRITEZF     Write Random with Zero Fill YES  YES
  41  29  F_TWREC       Test and Write Record    NO   NO   NO
  42  2A  F_LOCK        Lock record              YES  NO   NO
  43  2B  F_UNLOCK      Unlock Record            YES  NO   NO
* 44  2C  F_MULTISEC    Set Multisector Count    YES  YES
* 45  2D  F_ERRMODE     Set BDOS Error Mode      NO   YES
  46  2E  DRV_SPACE     Get Free Disk Space      YES  NO   NO
  47  2F  P_CHAIN       Chain to Program         NO   NO
  48  30  DRV_FLUSH     Flush Buffers            YES  NO   NO
  49  31  SCB_GETSET    Get/set CP/M Plus SCB    NO   NO   NO
* 50  32  S_BIOS        Direct BIOS Call         NO*  YES
  51  33  F_DMAOFF      Set DMA Base             NO   YES
  52  34  F_DMAGET      Get DMA Base             NO   YES
  53  35  MC_MAX        Get Maximum Memory       NO   YES
  54  36  MC_ABSMAX     Get Absolute Memory      NO   YES
  55  37  MC_ALLOC      Allocate Memory          NO   YES
  56  38  MC_ALLOCABS   Allocate Absolute Memory NO   YES
  57  39  MC_FREE       Free memory              NO   YES
  58  3A  MC_ALLFREE    Free All Memory          NO   YES
* 59  3B  P_LOAD        Program Load             NO*  YES
  60  3C  RSX_CALL      Call Resident System Ext.  NO   NO   NO
  61  3D  ?
  62  3E  ?
  63  3F  ?
  64  40  N_LOGON       Log On a Server          YES  NO   NO
  65  41  N_LOGOFF      Log Off a Server         YES  NO   NO
  66  42  N_SENDMSG     Send a Message           NO   NO   NO
  67  43  N_GETMSG      Get a Message            NO   NO   NO
  68  44  N_STAT        Get Network Status       YES  NO   NO
  69  45  N_RCT         Get Req. Config. Table   YES  NO   NO
  70  46  N_ATTRIB      Set Comp. Attributes     YES  NO   NO
  71  47  N_SCT         Get Ser. Config. Table   YES  NO   NO
  72  48  N_ERRMODE     Set Network Error Mode   NO   NO   NO
  73  49  N_ATTACH      Attach Network           NO   NO   NO
  74  4A  N_DETACH      Detach Network           NO   NO   NO
  75  4B  N_BUFSIZ      Set Message Buffer Size  ?
  76  4C  ?
  77  4D  N_PARATAB     Get Parameter Table      YES  NO

  99  63  F_TRUNCATE    Truncate File            YES  NO   NO
 100  64  DRV_SETLABEL  Set Directory Label      YES  NO   NO
 101  65  DRV_GETLABEL  Return Directory Label   YES  NO   NO
 102  66  F_TIMEDATE    Read File XFCB           YES  NO   NO
 103  67  F_WRITEXFCB   Write File XFCB          YES  NO   YES
 104  68  T_SET         Set Date and Time        NO   NO   YES
*105  69  T_GET         Get Date and Time        YES  NO   NO
*106  6A  F_PASSWD      Set Default Password     YES  NO   YES
```

```
 107  6B  S_SERIAL     Return Serial Number       NO   NO   NO
 108  6C  ?
 109  6D  C_MODE       Get/Set Console Mode       NO   NO   NO
 110  6E  C_DELIMIT    Get/Set Delimiter          NO   NO   NO
 111  6F  C_WRITEBLK   Write Console Block        NO   NO   NO
*112  70  L_WRITEBLK   Write List Block           NO*  NO   NO

 128  80  M_ALLOC      Allocate Memory Segment    NO   NO   NO
 129  81  M_ALLOC      Allocate Memory Segment    NO   NO   NO
 130  82  M_FREE       Free Memory Segment        NO   NO   NO
 131  83  DEV_POLL     Poll Device                NO   NO   NO
 132  84  DEV_WAITFLAG Flag Wait                  NO   NO   NO
 133  85  DEV_SETFLAG  Flag Set                   NO   NO   NO
*134  86  Q_MAKE       Make Queue                 YES  NO   NO
 135  87  Q_OPEN       Open Queue                 YES  NO   NO
 136  88  Q_DELETE     Delete Queue               YES  NO   NO
 137  89  Q_READ       Read Queue                 YES  NO   NO
 138  8A  Q_CREAD      Conditionally Read Queue   YES  NO   NO
 139  8B  Q_WRITE      Write Queue                YES  NO   NO
 140  8C  Q_CWRITE     Conditionally Write Queue  YES  NO   NO
 141  8D  P_DELAY      Delay Process              NO   NO   YES
 142  8E  P_DISPATCH   Dispatch Process           NO   NO   YES
*143  8F  P_TERM       Terminate Calling Process  NO   YES
*144  90  P_CREATE     Create Process             NO   NO   NO
 145  91  P_PRIORITY   Set Priority of Process    NO   NO   YES
 146  92  C_ATTACH     Attach Console             NO   NO   YES
 147  93  C_DETACH     Detach Console             NO   NO   YES
 148  94  C_SET        Set Console                NO   NO   NO
 149  95  C_ASSIGN     Assign Console             NO   NO   YES
 150  96  P_CLI        Call Command Line Interp.  NO   NO   NO
 151  97  P_RPL        Call Resident Procedure Lib.NO  NO   NO
 152  98  F_PARSE      Parse Filename             NO   YES
 153  99  C_GET        Get Console                NO   YES
 154  9A  S_SYSDAT     Get SYSDAT Address         NO   YES
*155  9B  T_SECONDS    Get Date and Time          YES  NO   NO
 156  9C  P_PDADR      Return Proc. Desc. Address NO   YES
 157  9D  P_ABORT      Abort Specified Process    NO   NO   NO
 158  9E  L_ATTACH     Attach List Device         YES  NO   NO
 159  9F  L_DETACH     Detach List Device         YES  NO   YES
 160  A0  L_SET        Set List Device            YES  NO   YES
 161  A1  L_CATTACH    Conditionally Attach List  YES  NO   NO
 162  A2  C_CATTACH    Conditionally Attach Con.  NO   NO   YES
 163  A3  S_OSVER      Return Version Number      NO   NO   YES
 164  A4  L_GET        Get List Device Number     YES  NO   NO
```

Table 2-6. Concurrent CP/M functions

```
 Dec Hex  Mnemonic     Description              S    R    L    H
 --- ---  ---------    -----------              ---  ---  ---  ---
*  0  00  P_TERMCPM    System Reset               NO   NO   YES
   1  01  C_READ       Console Input              YES  NO   YES
   2  02  C_WRITE      Console Output             YES  NO   YES
*  3  03  AUX_READ     Auxiliary Input            YES  NO   NO   YES
```

*	4	04	AUX_WRITE	Auxiliary Output	YES	NO NO YES
*	5	05	L_WRITE	List Output	YES	YES YES
	6	06	C_RAWIO	Direct Console I/O	YES	NO YES
	7	07	AXI_STAT	Auxiliary Input Status	NO	NO NO YES
	8	08	AXO_STAT	Auxiliary Output Status	NO	NO NO YES
*	9	09	C_WRITESTR	Print String	NO*	NO YES
*	10	0A	C_READSTR	Read Console Buffer	NO*	NO YES
*	11	0B	C_STAT	Get Console Status	YES	NO YES
*	12	0C	S_BDOSVER	Return BDOS Version Number	NO	NO YES
*	13	0D	DRV_ALLRESET	Reset Disk System	NO*	NO* YES
	14	0E	DRV_SET	Select Disk	YES	YES YES
	15	0F	F_OPEN	Open File	YES	YES YES
	16	10	F_CLOSE	Close File	YES	YES YES
	17	11	F_SFIRST	Search for First	YES	YES YES
	18	12	F_SNEXT	Search for Next	YES	YES YES
	19	13	F_DELETE	Delete File	YES	YES YES
	20	14	F_READ	Read Sequential	YES	YES YES
	21	15	F_WRITE	Write Sequential	YES	YES YES
	22	16	F_MAKE	Make File	YES	YES YES
	23	17	F_RENAME	Rename File	YES	YES YES
	24	18	DRV_LOGINVEC	Return Log-in Vector	YES	YES YES
	25	19	DRV_GET	Return Current Disk	YES	YES YES
	26	1A	F_DMASET	Set DMA Offset	NO	NO YES
	27	1B	DRV_ALLOCVEC	Get Allocation Vector	YES	YES YES
	28	1C	DRV_SETRO	Write Protect Disk	YES	YES YES
	29	1D	DRV_ROVEC	Get Read/Only Vector	YES	YES YES
	30	1E	F_ATTRIB	Set File Attributes	YES	YES YES
	31	1F	DRV_DPB	Get Disk Param. Block Addr.	YES	YES YES
*	32	20	F_USERNUM	Get/Set User Code	NO	NO YES
	33	21	F_READRAND	Read Random	YES	YES YES
	34	22	F_WRITERAND	Write Random	YES	YES YES
	35	23	F_SIZE	Compute File Size	YES	YES YES
	36	24	F_RANDREC	Set Random Record	YES	YES YES
*	37	25	DRV_RESET	Reset Drive	YES	YES YES
*	38	26	DRV_ACCESS	Access Drive	YES	YES YES
	39	27	DRV_FREE	Free Drive	YES	YES YES
	40	28	F_WRITEZF	Write Random with Zero Fill	YES	YES YES
	41	29	F_TWREC	Test and Write Record	NO	NO NO NO
	42	2A	F_LOCK	Lock record	YES	YES YES
	43	2B	F_UNLOCK	Unlock Record	YES	YES YES
*	44	2C	F_MULTISEC	Set Multisector Count	YES	YES YES
*	45	2D	F_ERRMODE	Set BDOS Error Mode	NO	NO YES
	46	2E	DRV_SPACE	Get Free Disk Space	YES	YES YES
*	47	2F	P_CHAIN	Chain to Program	NO*	NO* YES
	48	30	DRV_FLUSH	Flush Buffers	NO	NO YES
	49	31	SCB_GETSET	Get/set CP/M Plus SCB	NO	NO NO NO
*	50	32	S_BIOS	Direct BIOS Call	NO*	NO* YES
	51	33	F_DMAOFF	Set DMA Base	NO	NO YES
	52	34	F_DMAGET	Get DMA Base	NO	NO YES
	53	35	MC_MAX	Get Maximum Memory	NO	NO YES
	54	36	MC_ABSMAX	Get Absolute Memory	NO	NO YES
	55	37	MC_ALLOC	Allocate Memory	NO	NO YES
	56	38	MC_ALLOCABS	Allocate Absolute Memory	NO	NO YES
	57	39	MC_FREE	Free memory	NO	NO YES

```
  58  3A  MC_ALLFREE    Free All Memory          NO   NO   YES
* 59  3B  P_LOAD       Program Load             NO*  NO*  YES
  60  3C  RSX_CALL     Call Resident System Ext.  NO   NO   NO
  61  3D  ?
  62  3E  ?
  63  3F  ?
  64  40  N_LOGON      Log On a Server          YES  YES  NO   NO
  65  41  N_LOGOFF     Log Off a Server         YES  YES  NO   NO
  66  42  N_SENDMSG    Send a Message           NO   NO   NO   NO
  67  43  N_GETMSG     Get a Message            NO   NO   NO   NO
  68  44  N_STAT       Get Network Status       NO   YES  YES
  69  45  N_RCT        Get Req. Config. Table   NO   YES  YES
  70  46  N_ATTRIB     Set Comp. Attributes     YES  NO   NO   NO
  71  47  N_SCT        Get Ser. Config. Table   YES  YES  NO   NO
  72  48  N_ERRMODE    Set Network Error Mode   NO   YES  NO   NO
  73  49  N_ATTACH     Attach Network           NO   YES  NO   NO
  74  4A  N_DETACH     Detach Network           NO   YES  NO   NO
  75  4B  N_BUFSIZ     Set Message Buffer Size  ?
  76  4C  ?
  77  4D  N_PARATAB    Get Parameter Table      NO   YES  NO   NO

  99  63  F_TRUNCATE   Truncate File            YES  YES  YES
 100  64  DRV_SETLABEL  Set Directory Label      YES  YES  YES
 101  65  DRV_GETLABEL  Return Directory Label   YES  YES  YES
 102  66  F_TIMEDATE   Read File XFCB           YES  YES  YES
 103  67  F_WRITEXFCB  Write File XFCB          YES  YES  YES
 104  68  T_SET        Set Date and Time        NO   NO   YES
*105  69  T_GET        Get Date and Time        YES  NO   YES
*106  6A  F_PASSWD     Set Default Password     YES  YES  YES
 107  6B  S_SERIAL     Return Serial Number     NO   NO   YES
 108  6C  ?
 109  6D  C_MODE       Get/Set Console Mode     NO   NO   YES
 110  6E  C_DELIMIT    Get/Set Delimiter        NO   NO   YES
 111  6F  C_WRITEBLK   Write Console Block      NO   NO   YES
*112  70  L_WRITEBLK   Write List Block         NO*  NO*  YES

 128  80  M_ALLOC      Allocate Memory Segment  NO   NO   YES
 129  81  M_ALLOC      Allocate Memory Segment  NO   NO   YES
 130  82  M_FREE       Free Memory Segment      NO   NO   YES
 131  83  DEV_POLL     Poll Device              NO   NO   YES
 132  84  DEV_WAITFLAG  Flag Wait               NO   NO   YES
 133  85  DEV_SETFLAG   Flag Set                NO   NO   YES
*134  86  Q_MAKE       Make Queue               YES  YES  YES
 135  87  Q_OPEN       Open Queue               YES  YES  YES
 136  88  Q_DELETE     Delete Queue             YES  YES  YES
 137  89  Q_READ       Read Queue               YES  YES  YES
 138  8A  Q_CREAD      Conditionally Read Queue  YES  YES  YES
 139  8B  Q_WRITE      Write Queue              YES  YES  YES
 140  8C  Q_CWRITE     Conditionally Write Queue  YES  YES  YES
 141  8D  P_DELAY      Delay Process            NO   NO   YES
 142  8E  P_DISPATCH   Dispatch Process         NO   NO   YES
*143  8F  P_TERM       Terminate Calling Process  NO   NO   YES
*144  90  P_CREATE     Create Process           NO   NO   YES
 145  91  P_PRIORITY   Set Priority of Process  NO   NO   YES
```

146	92	C_ATTACH	Attach Console	NO	NO	YES
147	93	C_DETACH	Detach Console	NO	NO	YES
148	94	C_SET	Set Console	NO	NO	YES
149	95	C_ASSIGN	Assign Console	NO	NO	YES
150	96	P_CLI	Call Command Line Interp.	NO	NO	YES
151	97	P_RPL	Call Resident Procedure Lib.	NO	NO	YES
152	98	F_PARSE	Parse Filename	NO	NO	YES
153	99	C_GET	Get Console	NO	NO	YES
154	9A	S_SYSDAT	Get SYSDAT Address	NO	NO	YES
*155	9B	T_SECONDS	Get Date and Time	YES	NO	YES
156	9C	P_PDADR	Return Proc. Desc. Address	NO	NO	YES
157	9D	P_ABORT	Abort Specified Process	NO	NO	YES
158	9E	L_ATTACH	Attach List Device	YES	YES	YES
159	9F	L_DETACH	Detach List Device	YES	YES	YES
160	A0	L_SET	Set List Device	NO	NO	YES
161	A1	L_CATTACH	Conditionally Attach List	YES	YES	YES
162	A2	C_CATTACH	Conditionally Attach Con.	NO	NO	YES
163	A3	S_OSVER	Return Version Number	NO	NO	YES
164	A4	L_GET	Get List Device Number	NO	NO	YES

2.2 Implementation notes

Several system calls are flagged with an asterisk ("*") in Tables 2-2 through 2-6, to indicate that the network context affects their operation, or introduces some new considerations. In all cases, the exception is only a concern when the call is networked. The calls earmarked in the tables are described below in numeric order.

Table 2-7. Implementation notes

Format: Number, Mnemonic
Description

0 P_TERMCPM
Before a process is terminated by the local operating system, the NDOS logs off all servers in the process's Log-on Table. P_TERMCPM does not affect the networkstatus of any parent or child processes.

3 C_RAWIN
The Concurrent CP/M server is programmed to interpret a function 3 as a raw console input. This maintains compatibility with CP/NET Version 1.2 requesters, which allowed local consoles to be mapped to remote servers. Under DR Net, consoles cannot be mapped. For more information on the use of this function, see the "CP/NET System Manual".

4 C_RAWOUT
Like C_RAWIN, this function is interpreted by the Concurrent CP/M server to maintain compatibility with CP/NET Version 1.2. Consoles cannot be mapped to servers under DR Net. For more information on the use of this function, see the "CP/NET System Manual".

5 L_WRITE
The NDOS does not output every L_WRITE function mapped to a remote list device (printer). Instead, it accumulates the characters in a buffer until 64 characters (in requesters based on Concurrent CP/M or CP/M-86) or 128 characters (in requesters based on CP/NET Version 1.2) have been gathered. When the buffer is full, the call is sent as a function 5 with the buffer contents. When the shadow process receives a function 5, it knows to print the entire contents of the message, rather than a single character.

The bundling and transfer of data blocks is handled transparently with one important difference: the last buffer might need to be flushed explicitly. Any of the following calls can be used for this purpose:

00: P_TERMCPM
65: N_LOGOFF
74: N_DETACH
143: P_TERM
159: L_DETACH
160: L_SETNUM

Note: MP/M II servers interpret an 0FFh in a print file as an end of the listing indicator. DR Net servers do not interpret an 0FFh as a special character, and treat it as any other character in the print file.

9 C_WRITESTR
This function is only supported for the sake of CP/NET Version 1.2 requesters. When the requester receives a function 9, it decomposes the call into a series of function 6s. DR Net does not allow the mapping of consoles over the network.

10 C_READSTR
This function is only supported for the sake of CP/NET Version 1.2 requesters. When the requester receives a function 10, it decomposes the call into a series of function 6s. DR Net does not allow the mapping of consoles over the network.

11 C_STAT
This function is only supported for the sake of CP/NET Version 1.2 requesters. DR Net does not allow the mapping of consoles over the network.

12 S_BDOSVER
In nodes based on CP/M-86 and Concurrent CP/M, the NDOS always traps this call and return BDOS Version number 1431h (= 8086 Concurrent CP/M Release 3.1). In requesters based on CP/NET Version 1.2, the version number returned is 0122h (= 8080 MP/M II Release 2.2).

13 DRV_ALLRESET
This function is decomposed into the following two calls: DRV_SET (select disk A) and DRV_RESET (drive vector = 0FFFFh). Note, however, that function 13 only resets a drive when there are no open files on it. If one drive has an open file, no drives are reset.

32 F_USERNUM
A DR Net requester does not send this function to the server. However, file

user numbers are supported. (The file user number is sent with all read or write file messages.)

37 DRV_RESET
Just like function 13, DRV_RESET is conditional under DR Net. That is, an attempt to reset a remote drive returns failure when there is an open file on it.

38 DRV_ACCESS
Recall that this call inserts a special open file item into the system Lock List for every drive specified. The item is not entered, however, if the Lock List is full or the new entry puts the Lock List over its limit. In the network environment where the drive vector in register DX can reference drives on many different servers, it is possible to get a rejection from one server and success from others. Function 38 enters the item in the server's Lock List until it encounters a full server. Subsequently, the error message is returned and the function aborted. The successful entries remain intact, however.

44 F_MULTISEC
Multisector transfers can be made from requesters based on CP/M-86. A special buffer is reserved within DR Net, and the NDOS is written to provide all the support for this call.

45 F_ERRMODE
Use of the non-default BDOS error modes ("Return" or "Return and Display") is available to CP/M-86 requesters. Function 45 can be invoked without ill effects, and causes the NDOS to intercede when a non-network error response is received from a server. (The NDOS's response to network errors is controlled by function 72, N_ERRMODE.) However, local error conditions always result in the default BDOS error mode ("Display and Abort"), regardless of the mode selected by function 45.

47 P_CHAIN
The Concurrent CP/M SUP module decomposes this function into a series of other system calls. Because DR net supports all functions in this series, P_CHAIN can be used when the files referenced are remote.

50 S_BIOS
The NDOS does not intercept BIOS, XIOS, or S_BIOS calls. However, Concurrent CP/M converts S_BIOS calls to a list device into function 5 calls. These are handled as explained in the function 5 description above.

59 P_LOAD
P_LOAD is similar to P_CHAIN in that both are decomposed into a series of other functions. Because DR Net supports all functions in this series, P_LOAD can be used when the files referenced are remote.

105 T_GET
The CP/M-86 requester NDOS traps this call, and sends it to the default server for execution. In a requester based on Concurrent CP/M, however, the NDOS does not output this function. Instead, it is routed to the local system for execution.

106 F_PASSWD

The default password stored by this function is suitable only to provide file access. Whether the file is local or remotely-based is irrelevant. A default-server password can be stored in a requester, but this parameter is entered at system generation time, rather than by function call.

112 L_WRITEBLK
The Concurrent CP/M requester translates this call into a series of function 5, L_WRITE, calls.

134 Q_MAKE
All queue-related functions are supported with the following limitation: the maximum queue size is defined as the maximum message size minus one.

143 P_TERM
Like P_TERMCPM, function 143 is passed on to the local operating system for execution. However, before doing so, the NDOS sends a series of N_LOGOFF messages to all servers in that process's Log-on Table, and clears the print buffer. P_TERM does not affect the network status of any parent or child process.

144 P_CREATE
After a process has been logged on a server, all child processes it creates are automatically logged on as well. When P_CREATE is called, the NDOS intercepts it, and checks the calling process's parent to see if it is logged on. If the parent is logged on, the child is given a network environment consisting of a Network Data Area, Requester Configuration Table pointer, and Log-on Table.

Note: Although a child process inherits its parent's log-on Table, N_LOGON message are not sent until the child accesses a mapped resource. When this occurs, DR Net sends the N_LOGON to the designated server only. Other servers in the child's Log-on Table are also logged on only as needed.

155 T_SECONDS
Like function 105, T_GET, the Concurrent CP/M server process is equipped to perform function 155, but only CP/M-86 requesters are capable of issuing it. A requester based on Concurrent CP/M always routes this function to its local system.

EOF

DNPG3.WS4 (= "DR Net Programmer's Guide", section 3)

DR Net
Programmer's Guide

First Edition: March 1984

(Retyped by Emmanuel ROCHE.)

Section 3: DR Net error handling

Programs can encounter two kinds of errors when using the network:

- the standard array of BDOS physical errors, extended errors, and error codes caused by failures of the local or remote system

- network errors caused by failures in the network interface

This section describes how DR Net responds to BDOS failures within the local and server systems, and to network failures.

Note: This description of error handling is applicable only to requester nodes running CP/M-86 or Concurrent CP/M, and servers nodes running Concurrent CP/M. For more information regarding CP/NET Version 1.2 error handling, see the "CP/NET Reference Manual".

3.1 Local and remote BDOS errors

BDOS-related errors are reported to a program according to the conventions of the operating system under which the failure occurred. In a requester node, an error condition resulting from a call to a local resource is reported without DR Net's intervention. DR Net returns error codes resulting from remote disk drive, list device, or queue calls in registers AX, BX, and CX according to Concurrent CP/M Release 3.1 convention, in the standard display format.

BDOS errors alone do not bring down the server. Nor do they terminate the connection between a requester and its server. When a server returns the error code, the requester remains logged on, and subsequent network transactions can be conducted between the two. However, if the requester aborts as a result of the error, the server is logged off.

How the requester system responds to a BDOS error depends on the resident operating system, the current BDOS error mode, and whether the error resulted from a local or remote operation. In CP/M-86 requesters, all local and remote errors are handled according to the ruls of the "Abort and Display" error mode. However, function 45, F_ERRMODE, can be used to select one of the alternative modes, "Return" or "Return and Display", to handle remote errors. In this case, the NDOS mediates, and presents the error code to the calling

program according to the mode selected. The use of function 45 does not change CP/M-86's response to local errors from the default mode.

Under Concurrent CP/M, the default error mode is the "Display and Abort" mode. The "Return" or the "Return and Display" mode can be selected with function 45. In this case, the mode selected changes Concurrent CP/M's response for both local and remote errors.

The BDOS error mode selected with function 45 does not determine the NDOS's response to network errors. The network error mode is set by function 72, N_ERRMODE.

3.2 DR Net errors

Network-related error codes are returned according to the standard Concurrent CP/M rules for extended errors. This convention reports an error by returning an 0FFh in register AL, and the error code in register AH. How the error is reported to the calling process depends upon the current network error mode. In the default mode, a network error causes the program to abort, and the screen to display one of the following error messages. The number in parenthesis indicates the BDOS extended error code returned to the calling process.

Network Error: Server Not Logged (0CFFh)
Network Error: Requester Error (0DFFh)
Network Error: Physical Transmission Error (0EFFh)

3.2.1 DR Net error codes

Table 3-1 lists the error conditions by the value returned in register AH. The descriptions indicate the possible causes for the error, and the impact of that error with respect to the requesters relationship to the server.

Table 3-1. DR Net error codes

Format: AH value, Error name
Description

0Ch Server Error
Code 0Ch indicates that the call was received by the server, but that it was unable to execute it for one of the following reasons:

- The requester is not logged on that server.

- The server does not support the requested function.

0Dh Requester Error
Code 0Dh indicates that the call could not be implemented for one of the following reasons:

- The requester is logged on too many servers.

- A process called N_ATTACH when the maximum number of Requester Configuration Tables have already been copied.

- An illegal configuration table operation has occurred.

An illegal configuration table operation occurs when function 69, N_RCT, attempts an illegal change to the Requester Configuration Table. For example, a request to change the mapping of drive U is illegal.

0Eh Physical Network Error
Code 0Eh is returned when the NIOS returns an error condition. This results from one of the following reasons:

- The NIOS could not guarantee that the message was received intact.

- The NIOS did not receive a response from the server within a time limit specified by the transaction time-out.

0FFh Network Not Attached
Code 0FFh is not actually a network error. It is the error code returned to indicate that an unimplemented function has been called. This error code occurs in the network context when a network function is called before the network has been attached.

Network errors 0Ch and 0Eh cause selective log-offs from the designated server. For example, when a requester receives either error code 0Ch or 0Eh, the server is logged off for that process. However, no other servers are logged off, and the offending server remains logged on in all other processes. On the other hand, error 0Dh never logs off a server. None of these error conditions detach the node from the network.

3.2.2 DR Net time-out errors

DR Net contains the following time-out mechanisms:

- Requester-based transaction time-out

- Server-based keep-alive time-out

In the following descriptions of these timers, reference is made to the DR Net Parameter and Server Configuration Tables. For information on the Parameter Table, see the description of function 77, N_PARATAB, in Section 3. For information on the Server Configuration Table, see the description of function 71, N_SCT, also in Section 4.

Transaction time-out

The requester-based transaction time-out is set when the requester successfully sends a message to a server. It limits the time the requester waits for a server's response. Should the transaction time-out expires before the response message has been received, an extended physical error, code 0Eh, is returned to the calling process.

The duration of the transaction time-out is defined during DR Net system generation from a prompt in the GENNET program. The value is then stored in the Parameter Table.

Keep-alive time-out

The server-based keep-alive time-out is provided to prevent a requester that has crashed, or from any reason become disconnected from the network, from locking up server resources. Periodically, each requester sends a special keep-alive message to every server it is logged on to. When a server does not receive this signal within a given time period, it terminates all shadow processes created for that requester, and purges the requester node from its Server Configuration Table. No error message is returned to the requester from a keep-alive time-out.

The duration of the keep-alive time-out is proprietary to DR Net, and cannot be changed. Note that slack has been built into the server, so that one or two lost keep-alive messages (for example, from spurious transmission failures) are tolerated.

3.2.3 DR Net error modes

DR Net allows you to select between one of three error response modes with function 72, N_ERRMODE. (N_ERRMODE is described in Section 4.) The mode selected determines only the NDOS response to network errors 0Ch, 0Dh, and 0Eh. The response mode to local and remote BDOS errors is selected only with function 45. The three options, which closely parallel the BDOS error modes, offer the alternatives described in Table 3-2.

Table 3-2. DR Net error modes

Format: Mode
Meaning

"Return"
Write an 0FFh to AL; write the network error code (0Ch, 0Dh or 0Eh) to AH; duplicate these values in BL and BH; and return to the calling program.

"Return and Display"
Return the error code in AX and BX as above, display the error message on the process's console, and return to the calling process.

"Abort and Display"
Abort the program, display the error message on the process's console, and

return to the system prompt level.

The "Abort and Display" mode is the default mode selected when DR Net is loaded.

EOF

DNPG4.WS4 (= "DR Net Programmer's Guide", section 4)

DR Net
Programmer's Guide

First Edition: March 1984

(Retyped by Emmanuel ROCHE.)

Section 4: DR Net system calls

This section describes the DR Net network system calls. Of the 10 functions supported in the NDOS, functions 64 to 71 are compatible with with their CP/NET Version 1.2 namesakes. The functions in the range 72 through 77 are new. Table 4-1 lists the network functions alphabetically, and summarizes the input parameters and return values for each function.

Table 4-1. DR Net system calls

Mnemonic	Dec	Hex	Input parameter	Return values
N_ATTACH	73	49	DX: Process ID	AX: Return Code
N_ATTRIB	70	46	DL: Compatibility Mode Byte	AX: Return Code
N_DETACH	74	4A	DX: Process ID	AX: Return Code
N_ERRMODE	72	48	DL: Error Mode	AX: 0000h
N_LOGOFF	65	41	DS,DX: Long pointer to Log-on Parameter Block	AX: Return Code
N_LOGON	64	40	DS,DX: Long pointer to Log-on Parameter Block	AX: Return Code
N_PARATAB	77	4D	DS,DX: Long pointer to Parameter Table Buffer	AX: Return Code
N_RCT	69	45	DS,DX: Long pointer to RCT Control Block	AL: Return Code
N_SCT	71	47	DS,DX: Long pointer to Server Configuration Table (SCT) Buffer	AX: Return Code
N_STAT	68	44	None	AX: Network Status Word

The DR Net calling convention regarding the use of the CPU registers is shown in Table 4-2.

Table 4-2. System call register usage

Entry parameters	Returned values
CL: Hex function number	AX: Return code
DL/DX: Byte/Word value	BX: Same as AX
DX: Address -- Offset	CX: Concurrent CP/M error code (if any)

DS: Address -- Segment

In operation, DR Net preserves the contents of registers SI, DI, BP, SP, SS, DS, and CS. ES is also preserved, unless the value is replaced by a return segement value. This rule applies to calls trapped and executed remotely, as well as those network calls executed locally by DR Net.

The remainder of this section describes each network function: its input parameters, options, and return values.

Note: The possible error codes returned from the network functions include not only the extended network errors 0CFFh, 0DGGh, and 0EFFh, but also your operating system's error code for an illegal function. This code occurs whenever the network is called but the process is not attached. In this case, the function is passed to the local operating system, where it is not recognized as a legitiamte system call.

In the descriptions that follow, error code 0FFFFh indicates an illegitimate system call. This code is consistent with Concurrent CP/M's response to an illegal system call, and is accompanied by a 02h in register CX. Other operating systems might treat illegal functions differently. Refer to the manual provided with your operating system for the illegal system call error code.

N_ATTACH

Attach process to network

Entry Parameters:
Register CL: 49h (73)
DX: 0ffset to Process Descriptor
(0000h attaches calling process)

Returned Values:
Register AX: 0000h if success
0FFFFh if attach unsuccessful
BX: Same as AX

Function 73 attaches the process specified in DX to the network. The value in DX is offset to the Process Descriptor. If this value is 0000h, the current process is assumed.

N_ATTACH returns 0000h in AX if the attach is successful, or 0FFFFh if the attach is unsuccessful. Failure to attach is attributable to an invalid Process Descriptor address, to an overflow of available Static Buffer (see Note below) space, or to an inability to create another Requester Configuration Table.

Note: The DR Net "Static Buffer" is a reserved data area created implicitly during DR Net system generation from the number of server and requester processes specified. The Network Data Area is the work space required by DR Net to sustain an attached process. The Network Data Area is only deallocated

when N_DETACH is called and there is no previous Requester Configuration Table to replace the current one.

If both the calling process and its parent are not attached, a N_ATTACH causes the following events to occur:

- The network flag in the process's Process Descriptor is set.

- The process is allocated a Network Data Area (NDA) within the DR Net Static Buffer (see Note above).

- The process is allocated a blank Log-on Table.

- The process is allocated a first generation copy of the master Requester Configuration Table.

- The process is given an arbitrarily assigned network process ID, to distinguish it from other attached processes running on the same requester.

Child processes created by an attached process are automatically attached when they are created. In this case, the child is allocated its own NDA, its own Log-on Table (copied from its parent's Log-on Table), and its own network process ID. However, the child shares its parent's Requester Configuration Table, until the child calls N_ATTACH to get its own copy.

Note: Existing child processes are not automatically attached when their parent is attached.

If an attached process calls N_ATTACH, there is a slightly different result. No new NDA, Log-on Table, or network process ID are allocated, but a new copy of the requester Configuration Table is made. This copy is made from the process's current requester Configuration Table, rather than the master.

N_ATTACH does not log on a requester to a server. This requires a separate N_LOGON call. In all cases, a process must be attached before it can log on.

N_ATTRIB

Set compatibility attributes

Entry Parameters:
Register CL: 46h (70)
DL: Compatibility Mode Byte

Returned Values:
Register AX: 0000h if success
0EFFFh if physical network error
0FFFFh if network not attached
BX: Same as AX

Function 70 is called only in CP/M-86 requesters. It signals a server that a

program with compatibility attributes is to be loaded. This allows older applications to run without modification on Concurrent CP/M servers. There is never any need to call this function from an application. The Compatibility Mode Byte is constructed by the NDOS from compatibility attributes F1 through F4.

N_DETACH

Detach process from current environment

Entry Parameters:
Register CL: 4Ah (74)
DL: Offset to Process Descriptor
(0000h detaches calling process)

Returned Values:
Register AX: 0000h if success
0FFFFh if detach failed
BX: Same as AX

Function 74 detaches the process specified in DX from its current environment. It DOES NOT necessarily detach the process completely from the network. The value in DX is the offset to the Process Descriptor. If 0000h is specified, the current process is assumed.

N_DETACH returns 0000h in AX if successful, or 0FFFFh if the detach failed. Failure of this call results when an invalid Process Descriptor is put in DX, when the calling process is not already attached to the network, or when the process specified in DX is not attached.

What happens to a process that calls N_DETACH depends upon the generation number of its Requester Configuration Table (RCT). (See below for a review of RCT generations.) If the process has a first generation Requester Configuration Table, the following actions result:

- A N_LOGOFF message is sent to all servers in the process's Log-on Table. This terminates the process's shadow processes.

- The process's NDA and Log-on Table are deallocated.

- The process's network print buffer is flushed. (See the description of function 5, L_WRITE, in Section 2.2, "Implementation notes", for a description of the network print buffer.)

If the process has a second or higher generation Requester Configuration Table, an N_DETACH results in the following actions:

- The process's network print buffer is flushed.

- The process's Requester Configuration Table is replaced with the previous generation copy.

- The former Requester Configuration Table is deallocated if there is no possibility that an attached process will ever refer to it. Otherwise, the Requester Configuration Table remains allocated.

No other processes are affected by an N_DETACH.

A first generation Requester Configuration Table (RCT) is a copy of the master Requester Configuration Table. Second and third generation copies result when a process calls the N_ATTACH function a second and third time. See Section 1.3, "Network environments", for the description of how you use the N_ATTACH and N_DETACH functions to change generations of Requester Configuration Tables.

N_ERRMODE

Set network error mode

Entry Parameters:
Register CL: 48h (72)
DL: Error Mode

Returned Values:
Register AX: 0000h if success
0FFFFh if network not attached
BX: Same as AX

Function 72 determines how DR Net responds to network errors. The contents of register DL specify one of three modes. This function always returns a success code, unless the network is not attached.

N_ERRMODE is similar to function 45, F_ERRMODE, with respect to the three response modes. N_ERRMODE differs in that it controls the response to extended network errors 0CFFh, 0DFFh, and 0EFFh only.

The three error mode options are listed in Table 4-3 by their register DL code. All values are dexadecimal, and "xx" means any value other than "Ff" or "FE".

Table 4-3. Network error modes

Format: Code and Error response mode
Definition

FF "Return" mode
Return to the calling process with the error code in registers AX, BX, and CX. (Recall that CX contains the value 02h for all extended network errors.) Error recovery is expected to be performed by the application.

FE "Display and Return" mode
Display the error message on the console, and return the error code to the process as described in "Return" mode. Error recovery is expected to be performed by the application.

xx "Display and Abort" mode
Display the error message on the console, and abort the process.

The "Display and Abort" mode is the default error mode. When a network error occurs in this mode, the process's print buffers are flushed, and the server specified in the message is logged off before the message is displayed and the process aborted. The operator is returned to the host operating system, and the node remains attached to the network.

For more information on the extended network errors and how DR Net handles all error conditions, see Section 3, "DR Net error handling".

N_LOGOFF

Terminate the link with a server node

Entry Parameters:
Register CL: 41h (65)
DX: LPB Address -- Offset
DS: LPB Address -- Segment

Returned Values:
Register AX: 0000h if success
0CFFh if server error
0DFFh if process specified is not attached
0EFFh if physical network error
0FFFFh if network not attached
BX: Same as AX

Function 65 disconnects a process from a server. It requires a long pointer to the Log-on Parameter Block (see Figure 4-1) in registers DS and DX. This data structure contains the offset to the process descriptor to be logged off and the server node ID number. If 0000h is specified in the process ID field, the current process is assumed. Note that the ASCII Password field in the Log-on Parameter Block is not used with N_LOGOFF.

N_LOGOFF indicates success by returning 0000h in register AX. Failure is indicated by a number of error codes. Table 4-4 lists the codes and their meanings.

Table 4-4. N_LOGOFF error code definitions

Format: Error code
Definition

0CFFh
The process specified in the Log-on Parameter Block is not logged on the specified server.

0DFFh

The process specified in the Log-on Parameter Block is not attached.

0EFFh
A network error occurred during the log off. This code can be interpreted to indicate that either the request or response got garbled in transmission, the server is off-line, or the server is already logged off.

0FFFFh
The calling process is not attached.

Calling N_LOGOFF results in the following sequence of actions:

- The process's print buffer is flushed. (See the description of function 5, L_WRITE, in Section 2.2, "Implementation notes", for an explanation of the print buffer.)

- A log-off message is created and sent to the specified server.

- The server checks its list of shadow processes. When the requester process specified in the Log-on Parameter Table is matched, the shadow is terminated and the server returns a success message. If no match is found, the server returns error 0CFFh.

- The requester purges the server's ID number from the specified process's Log-on Table.

N_LOGON

Log on a process to a server

Entry Parameters:
Register CL: 40h (64)
DX: LPB Address -- Offset
DS: LPB Address -- Segment

Returned Values:
Register AX: 0000h if success
00FFh if no shadow process available
07FFh if wrong password
0CFFh if server network error
0DFFh if maximum servers are logged on, or
specified process not attached.
0EFFh if physical network error
0FFFFh if calling process not attached
BX: Same as AX

Function 64 connects a requester process to a server. It requires a long pointer to the Log-on Parameter Block in registers DS and DX. Figure 4-1 illustrates the contents of the Log-on Parameter Block, and Table 4-5 describes each field.

```
        0        1 2 3 4 5 6 7 8       9 A
      +-----------------+---------------+---------------+
      | Server ID Number | ASCII Password | Process ID (*) |
      +-----------------+---------------+---------------+
```

(*) Offset to process descriptor: a 0000h value indicates current process.

Figure 4-1. Log-on Parameter Block

Table 4-5. Log-on Parameter Block field descriptions

Format: Field
Meaning

Server ID Number
The logical node number of the server. The number is valid for values 00h through FEh.

ASCII Password
The password that must be matched by the server before it creates a shadow process for the requester.

Process ID
The offset to the process descriptor of the process to be logged on. A 0000h value indicates the current process.

N_LOGON indicates success by returning 0000h in register AX. Failure is indicated by the error codes shown. Table 4-6 describes the meaning of each error code.

Table 4-6. N_LOGON error code definitions

Format: Error code
Meaning

00FFh
The server could not create another shadow process.

07FFh
The password specified was not correct.

0DFFh
This code indicates one of the following:

- Although the current process is attached to the network, the process specified in the Log-on Parameter Block is not.

- The process specified is already logged on 16 servers.

0EFFh
A physical transmission occurred and prevented the original message from getting to the server, or the server's response message from getting to the requester.

0FFFFh
The calling process is not attached.

Calling N_LOGON results in the following sequence of events.

1. If the node is attached, the process's Log-on Table is checked to determine whether or not another server can be accommodated (limit 16). If the node is not attached, error 0FFFFh is returned to the calling process. If the Log-on Table is full, error code 0DFFh is returned.

2. When another server can ba accommodated, a N_LOGON message is sent to the server.

3. The server determines if it can accommodate another requester with a shadow process. Success is returned if it has an unallocated server process available in its pool. Otherwise, an 00FFh is returned.

4. The requester, only after receiving a success code, adds the server ID number to its Log-on Table, and returns the success code to the calling process. Otherwise, the error code is returned.

Note: A child process created by a logged_on process automatically receives a copy of the parent's Log-on Table. (This results because the parent is attached; see the N_ATTACH description above.) This grants the child access to all the server's that the parent has logged on. However, this privilege is not extended to a child process created before the parent logged on. Separate N_LOGON functions must be called by these processes.

N_PARATAB

Get Parameter Table

Entry Parameters:
Register CL: 4Eh (77)
DX: Parameter Table Buffer Address -- Offset
DS: Parameter Table Buffer Address -- Segment

Returned Values:
Register AX: 0000h if success
0FFFFh if network not attached
BX: Same as AX

Function 77 copies the node's DR Net Parameter Table into memory at the address specified in registers DS and DX. This function always returns success, unless the network is not attached, in which case 0FFFFh is returned in register AX.

The Parameter Table is 1Fh (31) bytes long, and contains the global system definitions for each network node. Figure 4-2 illustrates the format. Table 4-

7 contains the field definitions.

Numbers indicate hex offset from first byte in table.

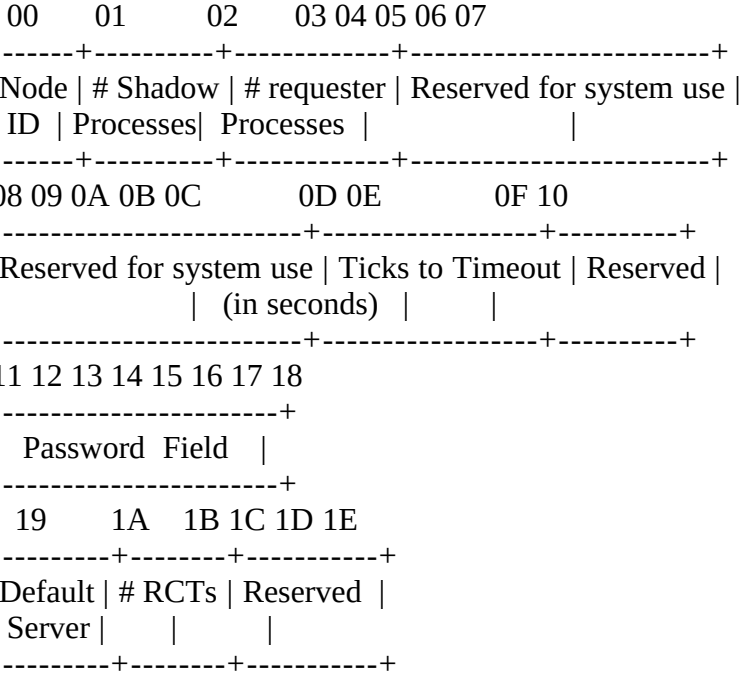


Figure 4-2. Parameter Table

Table 4-7. Parameter table field defintions

Format: Field name
Definition

Node ID
The hexadecimal network ID number that uniquely identifies the node.

Server Processes
The total number of simultaneous, shadow processes that can be supported on this server node.

Requester Processes
The total number of simultaneous, attached requester processes that can be supported on this requester node.

Transaction Time-out
The number of seconds allowed for a network transaction to complete before the requester aborts the call. See Section 3, "DR Net time-out errors", for a complete description of the consequences.

Password
In requester nodes, the default password transferred in every N_LOGON message. If a password is entered by the operator, the contents of this field are ignored. In server nodes, the password that must be matched to log on.

Default Server

The hexadecimal ID number of the server node referenced when no server is specified.

RCT
The number of Requester Configuration Tables that can exist simultaneously in this requester. (Recall that every N_ATTACH makes another copy of the Requester Configuration Table.)

N_RCT

Set Requester Configuration Table, or get environment

Entry Parameters:
Register CL: 45h (69)
DX: RCT Control Block -- Offset
DS: RCT Control Block -- Segment

Returned Values:
Register AX: 0000h if success
0DFFh if illegal configuration requested
0FFFFh if network not attached

Function 69 changes a disk drive, list device, or queue entry in the process's Requester Configuration Table, or retrieves the process's Requester Configuration Table and Log-on Table. The call requires along pointer in registers DX and DS to the RCT Control Block.

N_RCT returns 0000h in register AX to indicate success. If the value in the first byte, the command byte, of the RCT Control Block specified a read operation, success indicates that the entire Requester Configuration and Log-on Tables have been transferred into memory, starting at pointer address. If the command byte specified one of the write operations, the remainder of the RCT Control Block is used as replacement values for the values in the Requester Configuration Table. Values in the Log-on Table cannot be changed with this or any other function.

Figure 4-3 illustrates the formats of the RCT Control Block for each command byte value. The contents of bytes 01h through 03h are only valid for command types 00, 02, 03, and 04. When 0FFh is specified in the command byte, it is assumed that a total 1A0h (316) bytes are reserved for the Requester Configuration Table and the Log-on Table, starting from the pointer location. Table 4-8 describes the contents of each field in the control block.

Note: The description of the RCT Control Block anticipates that you are familiar with the Requester Configuration and Log-on Tables, and the format of each map entry. See Section 1.3, "Network environments", for the description of the fields in both of these tables.

Numbers indicate hex offset from first byte in table.

Command +---+--> High-order 3 Bits Not Used
Type: 00 01 | | 02 03

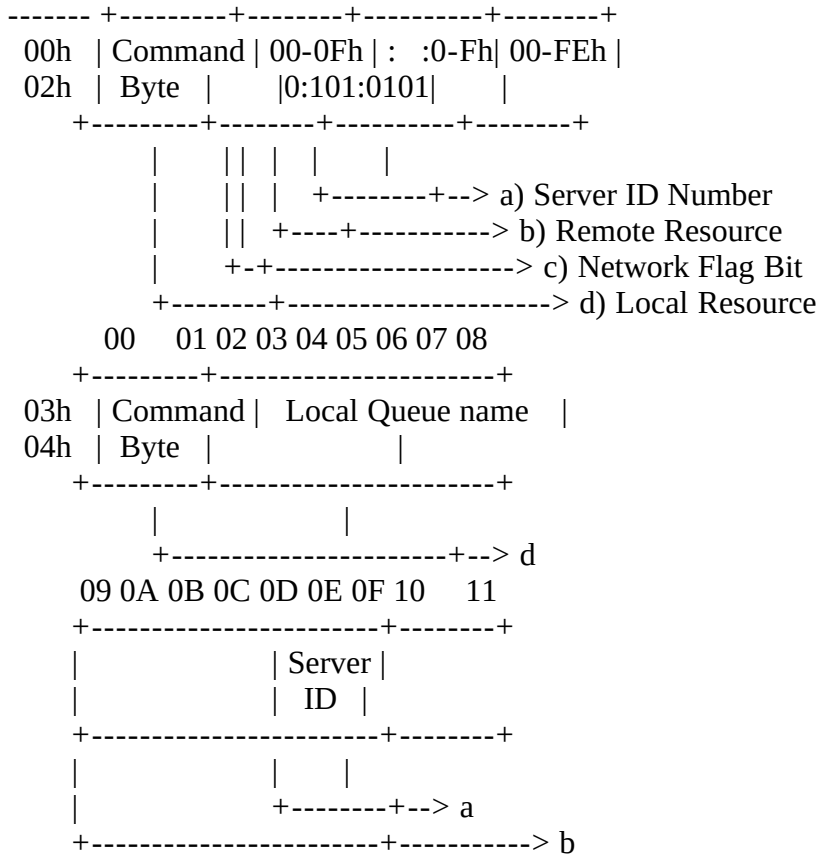


Figure 4-3. RCT Control Block command formats

Table 4-8. RCT Control Block field definitions

Format: Field
Definition

Command Byte
Only the following values are legal for the command byte:

00h: Change a disk drive assignment.
02h: Change a list device assignment.
03h: Map a local queue to a server.
04h: Remove mapping of a local queue.
0FFh: Read Requester Configuration and Log-on Tables.

Local Resource
For command byte values 00h and 02h, the local resource is a single-byte field that indicates the physical number of the local disk drive or list device, respectively, to be changed. Only numbers in the range 00h to 0Fh are accepted. For command byte values 03h and 04h, the local resource is an eight-byte field that specifies the local queue name. Remember that case is significant in queue name; lowercase letters are not translated into uppercase.

Network Flag Bit
This bit is relevant to command byte values 00h and 02h, and determines whether system calls are implemented on the local resource, or are sent to the

remote resource specified in the low-order nibble of this byte. The device is selected according to the following bit values:

0: local resource
1: remote resource

If the value of the flag bit is 0, the remainder of the RCT Control Block is ignored.

Remote Resource
For command byte values 00h and 02h, this is a four-byte value that indicates the physical number of the replacement disk drive or list device. Valid entries are in the range 00h to 0Fh. For command byte value 03h, the remote resource is an eight-byte field that contains the replacement queue name. Remember that case is significant in queue name. For command byte value 04h, this field is irrelevant.

Server ID Number
For command byte values 00h, 02h, and 03h, this is a single-byte field that contains the hexadecimal ID number of the destination server node. Valid entries are in the range 00h to 0FEh. This field is ignored when the Network Flag Bit is 0, or the command byte value is 04h.

Errors are indicated in register AX, where one of two values can be expected. Table 4-9 lists the values and their meanings.

Table 4-9. N_RCT error code definitions

Format: Error code
Definition

0DFFh
An illegal configuration operation was specified. For example, an attempt to map drive 1Fh would return error 0DFFh.

0FFFFh
The process is not attached to the network.

N_SCT

Get Server Configuration Table

Entry Parameters:
Register CL: 47h (71)
DX: SCT Buffer -- Offset
DS: SCT Buffer -- Segment

Returned Values:
Register AX: 0000h if success
0CFFh if server not logged on
0EFFh if physical network error

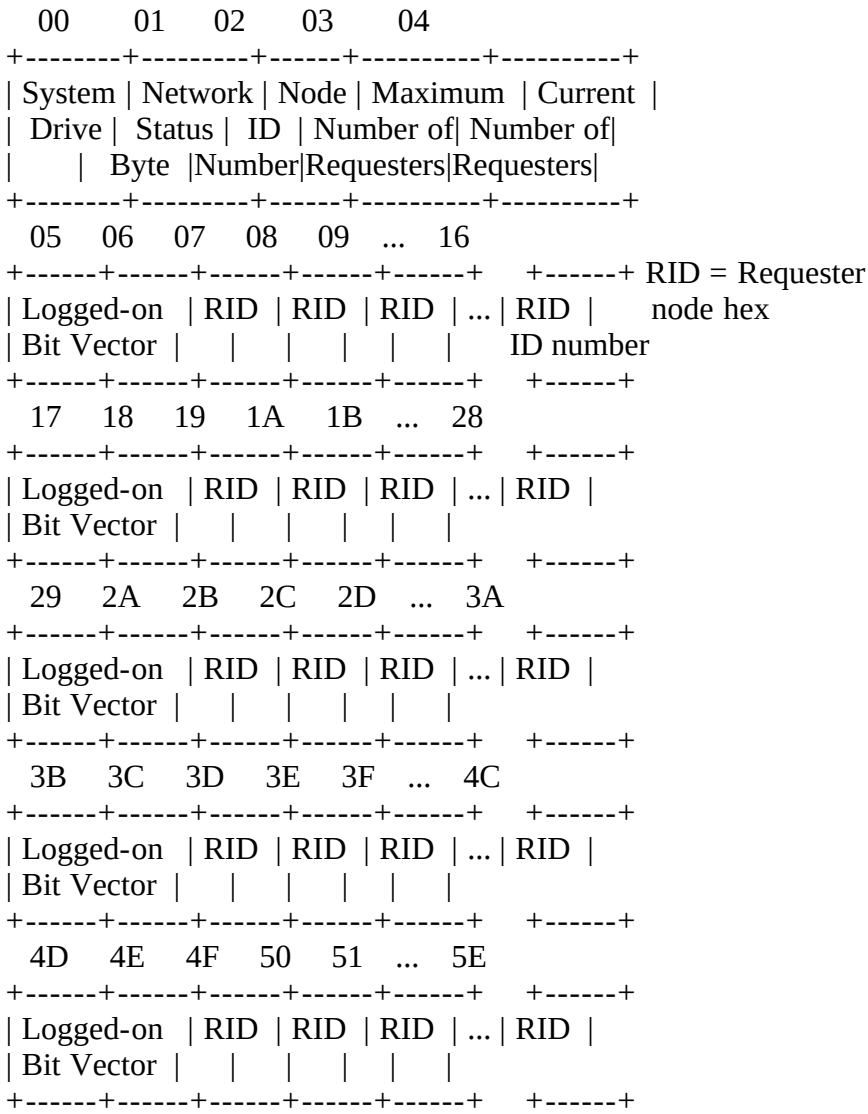
0FFFFh if network not attached
BX: Same as AX

Function 71 retrieves the Server Configuration Table (SCT) from a designated server. The buffer specified in registers DS and DX must be 70h (112) bytes long. The success or failure of this function is indicated by the return code in register AX.

The SCT Buffer serves two purposes. First, the initial byte is read to determine which server to interrogate. Second, the Server Configuration Table is written into it. The server number in the first byte is overwritten in the process. Figure 4-4 illustrates the Server Configuration Table. Table 4-10 describes the different fields.

Note: DR Net does not actually maintain a Server Configuration Table. Instead, it manufactures one when it receives a function 71. In addition, this table differs from the Server Configuration Table maintained in a CP/NET Version 1.2 server. The two are the same for bytes 00h through 16h in Figure 4-4. At this point, the CP/NET Version 1.2 SCT ends.

Numbers indicate hex offset from first byte in table.



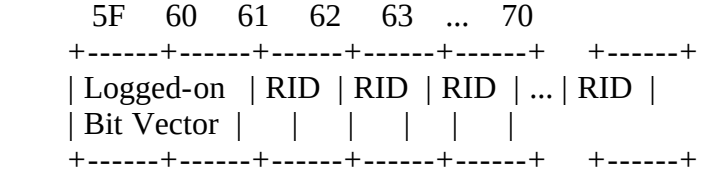


Figure 4-4. Server Configuration Table

Table 4-10. Server Configuration Table field definitions

Format: Field
Definition

Default File Drive
This field contains the physical number for the server's default disk drive. This is also the drive used by the server for the storage of temporary files.

Network Status Byte
This field is available for use by the communications software. It has no meaning in DR Net.

Node ID Number
The hexadecimal, server ID node number is stored in this field. The DID (Destination ID) field of all incoming messages is compared against this value before a message is accepted.

Maximum Number of Requesters
The maximum number of requesters supported by this server is recorded in this field. The value is copied from byte 01h of the Parameter Table, and indicates the number of shadow processes that can be supported on this node.

Current Number of Requesters
This value indicates the current number of shadow processes.

Log-on Bit Vector
This two-byte field indicates which of the next 16 requester ID fields in the table contains an entry.

RID
The Server Configuration Table contains 96 single-byte fields in which is stored the hexadecimal ID number for each requester node logged on. The requesters are grouped in sequences of 16, each preceded by a Log-on Bit Vector.

The return value in register AX indicates success with a value of 0000h. Failures are also indicated in register AX, and can result for a number of reasons. Table 4-11 lists and defines the possible error codes.

Table 4-11. N_SCT error code definitions

Format: Error code
Definition

0CFFh
The server specified in the SCT Buffer is not logged on.

0EFFh
There is a network physical error; no response was received.

0FFFFh
The calling process is not attached to the network.

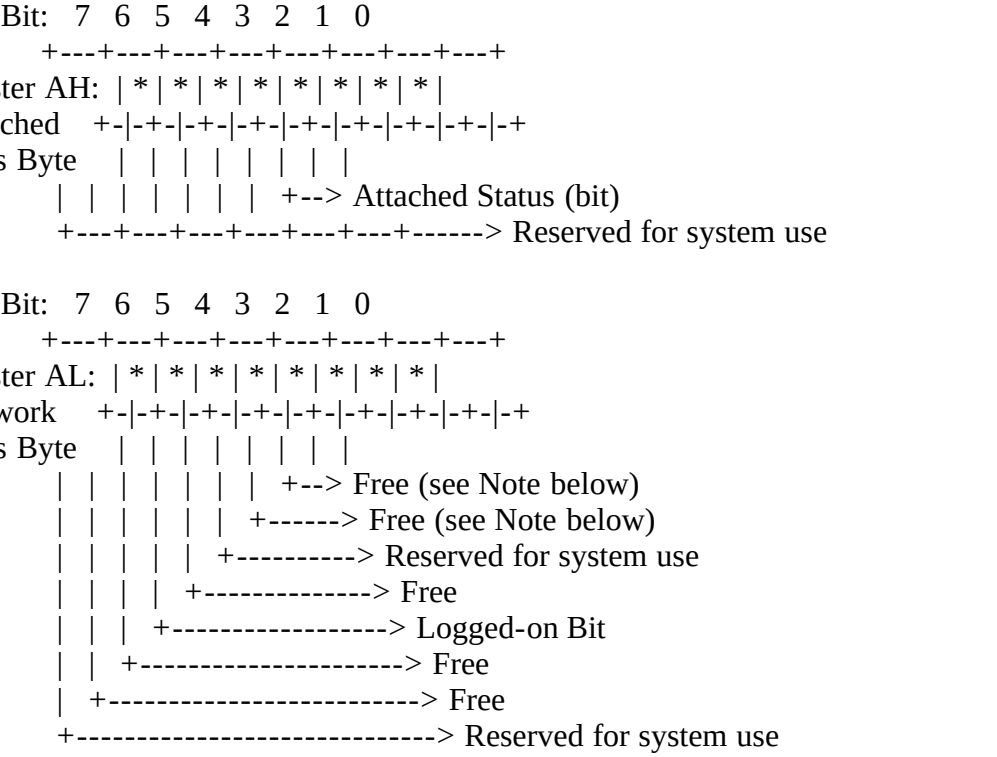
N_STAT

Get network status word

Entry Parameters:
Register CL: 44h (68)

Returned Values:
Register AX: Network Status Word
0FFFFh if network not attached
BX: Same as AX

Function 68 returns the network status word. This value consists of the Network Status Byte in register Al, and the Attached Status Byte in register AH. Figure 4-5 illustrates the contents of the entire word.



(Note: Under CP/NET Version 1.2, bit 1 = RCVERR, and bit 0 = SNDERR.)

Figure 4-5. Network status word

The Attached Status Byte shown in Figure 4-5 indicates whether or not the calling process is attached to the network. Only the least significant bit (bit 0) of AH is used, where:

0 = not attached
1 = attached

Bits 7 through 1 are reserved for future use. The program example below shows the use of this byte in a routine found in many of the DR Net utilities.

The Netswork Status Byte returned by DR Net is compatible with the status byte in CP/NET Version 1.2. However, CP/NET Version 1.2 does not recognize the Attached Status Byte in register AH. Three of the bits in this byte are reserved for system use. Of these three, only bit 4 is significant; it indicates whether or not process is logged on a server. Bits 7 and 2 are reserved for system use. The remainder of the bits are not used by DR Net. However, they can be programmed by a special NIOS function.

Note: Under CP/NET Version 1.2, bits 1 and 0 of the network status byte were defined as follows:

- Bit 1 -- RCVERR
  An error was encountered by the NIOS receive message routine.

- Bit 0 -- SNDERR
  An error is encountered by the NIOS send message routine.

These bits are not implemented under DR Net.

Listing 4-1 is an example routine that determines if the calling process is attached to the network. Only when the process is attached can it call the network functions. Otherwise, the host operating system returns an 0FFFFh.

Listing 4-1. Check attach status

```
/*----------------------------------------
      CHECK THE ATTACH STATUS
----------------------------------------*/
#include <stdio.h>

EXTERN BDOS();          /* BDOS Interface */

Boolean net_check()
{
      return ((BDOS(68)&0xFF00)==0x0100)
}


EOF
```

DNPG5.WS4 (= "DR Net Programmer's Guide", appendixes)

DR Net
Programmer's Guide

First Edition: March 1984

(Retyped by Emmanuel ROCHE.)

Appendixes

Appendix A: System call summary

All Concurrent CP/M Release 3.1 and DR Net network functions are listed below by function number. The summary indicates the input parameters and the returned values for each function. See Tables 2-2 through 2-6 in Section 2 to determine which calls are supported over the network.

The command summaries are presented in the following format:

```
Dec  Hex  Mnenonic     Input Parameters (IP)
        Description  Returned  Values (RV)
```

These fields are defined as follows:

- "Dec" and "Hex" are the system call's decimal and hexadecimal numbers, respectively.

- "Mnemonic" is the function name.

- "Description" is a short description of the function's purpose.

- "Input Parameters" describes the register and value expected by the system call.

- "Returned Values", unless stated otherwise, describes the value returned in register AX by the system call.

The "Returned Values" field frequently contains one of two values which indicate that the function succeeded or failed. When this is the case, the two values are shown in the form nnn/mmm, where "nnn" indicates the success, and "mmm" indicates the failure code. In both the nnn and mmm values, xx and yy are used to indicate that directory codes and/or error codes are returned. See the "Concurrent CP/M Programmer's Reference Guide" for the return code values.

Concurrent CP/M's system call calling and return conventions for register use are reviewed in Table A-1.

Table A-1. Concurrent CP/M system call calling and return conventions

Entry parameters by register	Returned values by register
CL: System call number	CX: Error code
DL: Byte parameter	AL: Byte return
DX: Word parameter	AX: Word return, or
DS/DX: Address -- Segment & Offset	AX: Address -- Offset
	ES: Address -- Segment
	BX: Same as AX

Under Concurrent CP/M, the contents of registers SI, DI, BP, SP, SS, DS, and CS are preserved. The ES register is also preserved, but only when it is not used to hold a return segment value. Only error codes are returned in CX; BDOS file errors and extended errors are always returned in AX.

In Table A-2, the input value of the CL register and the output value of the BX register are not shown. In all cases, CL indicates to the operating system which call is to be executed, and should be filled with the function value listed in the first two columns. Register BX always contains a duplicate of the register AX contents.

Table A-2. System call summary

Dec	Hex	Mnenonic	Input Parameters (IP)
		Description	Returned Values (RV)
0	00	P_TERMCPM	IP: none
		System Reset	RV: none
1	01	C_READ	IP: none
		Console Input	RV: AL: Character
2	02	C_WRITE	IP: DL = Character
		Console Output	RV: none
3	03	AUX_READ	IP: ?
		Auxiliary Input	RV: ?
4	04	AUX_WRITE	IP: ?
		Auxiliary Output	RV: ?
5	05	L_WRITE	IP: DL = Character
		List Output	RV: none
6	06	C_RAWIO	IP: DL = 0FFh, 0FEh, 0FDh, or Character
		Direct Console I/O	RV: AL: Status, or Character
7	07	AXI_STAT	IP: ?
		Auxiliary Input Status	RV: ?
8	08	AXO_STAT	IP: ?
		Auxiliary Output Status	RV: ?
9	09	C_WRITESTR	IP: DS/DX: Source buffer address
		Print String	RV: none
10	0A	C_READSTR	IP: DS/DX: Destination buffer address
		Read Console Buffer	RV: none
11	0B	C_STAT	IP: none
		Get Console Status	RV: AL: 01/00 (Char. Ready/Not Ready)
12	0C	S_BDOSVER	IP: none
		Return BDOS Version Number	RV: AL: BDOS version number
13	0D	DRV_ALLRESET	IP: none
		Reset Disk System	RV: AL: 00h/0FFh

14 0E DRV_SET IP: DL = Disk number (00h-0Fh)
Select Disk RV: 00xxh/xxFFh
15 0F F_OPEN IP: DS/DX = FCB address
Open File RV: 00xxh/xxFFh
16 10 F_CLOSE IP: DS/DX = FCB address
Close File RV: 00xxh/xxFFh
17 11 F_SFIRST IP: DS/DX = FCB address
Search for First RV: 00xxh/xxFFh
18 12 F_SNEXT IP: DS/DX = FCB address
Search for Next RV: 00xxh/xxFFh
19 13 F_DELETE IP: DS/DX = FCB address
Delete File RV: 00xxh/xxFFh
20 14 F_READ IP: DS/DX = FCB address
Read Sequential RV: 0000h/xxyyh
21 15 F_WRITE IP: DS/DX = FCB address
Write Sequential RV: 0000h/xxyyh
22 16 F_MAKE IP: DS/DX = FCB address
Make File RV: 0000h/xxFFh
23 17 F_RENAME IP: DS/DX = FCB address
Rename File RV: 0000h/xxFFh
24 18 DRV_LOGINVEC IP: none
Return Log-in Vector RV: Log-in vector
25 19 DRV_GET IP: none
Return Current Disk RV: Drive number (00h-0Fh)
26 1A F_DMASET IP: DX = New DMA segment address
Set DMA Offset RV: none
27 1B DRV_ALLOCVEC IP: none
Get Allocation Vector RV: Alloc. Vector Addr. -- Offset
Address for Current 0FFFFh if failure
Drive ES = Alloc. Vector Addr. -- Segment
28 1C DRV_SETRO IP: none
Write Protect Disk RV: 00h/0FFh
29 1D DRV_ROVEC IP: none
Get Read/Only Vector RV: Read/Only Vector
30 1E F_ATTRIB IP: DS/DX = FCB address
Set File Attributes RV: 0000h/xxFFh
31 1F DRV_DPB IP: none
Get Disk Parameter RV: DPB Address -- Offset
Block Address 0FFFFh if failure
(DPB) ES = DPB Address -- Segment
32 20 F_USERNUM IP: DL = 0FFh (Get user code)
Get/Set User Code xxh (Set user code, xx = code)
RV: Current user code on Get
33 21 F_READRAND IP: DS/DX = FCB address
Read Random RV: 0000h/xxyyh
34 22 F_WRITERAND IP: DS/DX = FCB address
Write Random RV: 0000h/xxyyh
35 23 F_SIZE IP: DS/DX = FCB address
Compute File Size RV: 0000h/xxFFh
36 24 F_RANDREC IP: DS/DX = FCB address
Set Random Record RV: none
37 25 DRV_RESET IP: DX = Drive vector
Reset Drive RV: 00h/0FFh
38 26 DRV_ACCESS IP: DX = Drive vector

Access Drive RV: 0000h/xxFFh
39 27 DRV_FREE IP: DX = Drive vector
Free Drive RV: none
40 28 F_WRITEZF IP: DS/DX = FCB address
Write Random with Zero Fill RV: 0000h/xxyyh
41 29 F_TWREC IP: DS/DX = FCB address
Test and Write Record RV: 0000h/xxyyh
42 2A F_LOCK IP: DS/DX = FCB address
Lock record RV: 0000h/xxyyh
43 2B F_UNLOCK IP: DS/DX = FCB address
Unlock Record RV: 0000h/xxyyh
44 2C F_MULTISEC IP: DL = Number of sectors
Set Multisector Count RV: AL: 00h/0FFh
45 2D F_ERRMODE IP: BDOS error mode
Set BDOS Error Mode RV: none
46 2E DRV_SPACE IP: DL = Drive number
Get Free Disk Space RV: 0000h/xxFFh
47 2F P_CHAIN IP: none
Chain to Program RV: 0000h/0FFFFh
48 30 DRV_FLUSH IP: none
Flush Buffers RV: 0000h/xxFFh
49 31 SCB_GETSET IP: ?
Get/set CP/M Plus SCB RV: ?
50 32 S_BIOS IP: DS/DX = BIOS descriptor address
Direct BIOS Call RV: BIOS return code
51 33 F_DMAOFF IP: DX = New DMA address -- Offset
Set DMA Base RV: none
52 34 F_DMAGET IP: none
Get DMA Base RV: DMA address -- Offset
ES = DMA address -- Segment
53 35 MC_MAX IP: DS/DX = Memory Control Block address
Get Maximum Memory RV: AL: 00h/0FFh
54 36 MC_ABSMAX IP: DS/DX = Memory Control Block address
Get Absolute Memory RV: AL: 00h/0FFh
55 37 MC_ALLOC IP: DS/DX = Memory Control Block address
Allocate Memory RV: AL: 00h/0FFh
56 38 MC_ALLOCABS IP: DS/DX = Memory Control Block address
Allocate Absolute Memory RV: AL: 00h/0FFh
57 39 MC_FREE IP: DS/DX = Memory Control Block address
Free memory RV: AL: 00h/0FFh
58 3A MC_ALLFREE IP: none
Free All Memory RV: none
59 3B P_LOAD IP: DS/DX = FCB address
Program Load RV: Program's Base Page Address
0FFFFh if failure
60 3C RSX_CALL IP: ?
Call Resident System Ext. RV: ?
61 3D ?
62 3E ?
63 3F ?
64 40 N_LOGON IP: DS/DX = Log-on Parameter Block Address
Log On a Server RV: 0000h/xxFFh
65 41 N_LOGOFF IP: DS/DX = Log-on Parameter Block Address
Log Off a Server RV: 0000h/xxFFh

```
 66  42  N_SENDMSG           IP: ?
     Send a Message       RV: ?
 67  43  N_GETMSG            IP: ?
     Get a Message        RV: ?
 68  44  N_STAT              IP: none
     Get Network Status    RV: Network Status Word
 69  45  N_RCT               IP: DS/DX = RCT Control Block Address
     Get Req. Config. Table  RV: 0000h/xxFFh
 70  46  N_ATTRIB            IP: DL = Compatibility mode byte
     Set Comp. Attributes  RV: 0000h/xxFFh
 71  47  N_SCT               IP: DS/DX = SCT destination buffer address
     Get Ser. Config. Table  RV: 0000h/xxFFh
 72  48  N_ERRMODE           IP: DL = Network error mode
     Set Network Error Mode  RV: 0000h
 73  49  N_ATTACH            IP: DX = Process Descriptor Address
     Attach Network       RV: 0000h/0FFFFh
 74  4A  N_DETACH            IP: DX = Process Descriptor Address
     Detach Network       RV: 0000h/0FFFFh
 75  4B  N_BUFSIZ            IP: ?
     Set Message Buffer Size RV: ?
 76  4C  ?
 77  4D  N_PARATAB           IP: DS/DX = Buffer address
     Get Parameter Table   RV: 0000h

 99  63  F_TRUNCATE          IP: DX = FCB address
     Truncate File        RV: 0000h/xxFFh
100  64  DRV_SETLABEL        IP: DS/DX = XFCB address
     Set Directory Label   RV: 0000h/xxFFh
101  65  DRV_GETLABEL         IP: DS/DX = Drive number
     Return Directory Label  RV: Directory label data byte
                          00h if no directory label found
                          xxFFh if failure
102  66  F_TIMEDATE          IP: DS/DX = XFCB address
     Read File XFCB       RV: 0000h/xxFFh
103  67  F_WRITEXFCB         IP: DS/DX = XFCB address
     Write File XFCB      RV: 0000h/xxFFh
104  68  T_SET               IP: DS/DX = TOD address
     Set Date and Time     RV: none
105  69  T_GET               IP: DS/DX = TOD address
     Get Date and Time     RV: AL: Seconds
106  6A  F_PASSWD            IP: DS/DX = Password address
     Set Default Password  RV: none
107  6B  S_SERIAL            IP: DS/DX = Destination buffer address
     Return Serial Number  RV: none
108  6C  ?
109  6D  C_MODE              IP: DX = Console mode, or 0FFFFh
     Get/Set Console Mode  RV: Console mode
110  6E  C_DELIMIT           IP: DX = Output delimiter, or 0FFFFh
     Get/Set Delimiter     RV: AL: Output delimiter
111  6F  C_WRITEBLK          IP: DX = Char. Control Block Address
     Write Console Block   RV: none
112  70  L_WRITEBLK          IP: DX = Char. Control Block Address
     Write List Block      RV: none
128  80  M_ALLOC             IP: DS/DX = Memory Parameter Block Address
```

Allocate Memory Segment RV: 0000h/0FFFFh
129 81 M_ALLOC IP: Same as above
Allocate Memory Segment RV: Same as above
130 82 M_FREE IP: DS/DX = Parameter Block Address
Free Memory Segment RV: 0000h/0FFFFh
131 83 DEV_POLL IP: DL = Device number
Poll Device RV: 0000h/0FFFFh
132 84 DEV_WAITFLAG IP: DL = Wait flag
Flag Wait RV: 0000h/0FFFFh
133 85 DEV_SETFLAG IP: DL = Flag number
Flag Set RV: 0000h/0FFFFh
134 86 Q_MAKE IP: DS/DX = Queue Descriptor Address
Make Queue RV: 0000h/0FFFFh
135 87 Q_OPEN IP: DS/DX = Queue Descriptor Address
Open Queue RV: 0000h/0FFFFh
136 88 Q_DELETE IP: DS/DX = Queue Descriptor Address
Delete Queue RV: 0000h/0FFFFh
137 89 Q_READ IP: DS/DX = Queue Descriptor Address
Read Queue RV: 0000h/0FFFFh
138 8A Q_CREAD IP: DS/DX = Queue Descriptor Address
Conditionally Read Queue RV: 0000h/0FFFFh
139 8B Q_WRITE IP: DS/DX = Queue Parameter Block Address
Write Queue RV: 0000h/0FFFFh
140 8C Q_CWRITE IP: DS/DX = Queue Parameter Block Address
Conditionally Write Queue RV: 0000h/0FFFFh
141 8D P_DELAY IP: DX = Number of system ticks
Delay Process RV: none
142 8E P_DISPATCH IP: none
Dispatch Process RV: none
143 8F P_TERM IP: DL = Termination code
Terminate Calling Process RV: none
144 90 P_CREATE IP: DS/DX = Process Descriptor Address
Create Process RV: 0000h/0FFFFh
145 91 P_PRIORITY IP: DL = Priority
Set Priority of Process RV: none
146 92 C_ATTACH IP: none
Attach Console RV: none
147 93 C_DETACH IP: none
Detach Console RV: 0000h/0FFFFh
148 94 C_SET IP: DL = Console number
Set Console RV: 0000h/0FFFFh
149 95 C_ASSIGN IP: DS/DX = Assign Control Block Address
Assign Console RV: 0000h/0FFFFh
150 96 P_CLI IP: DS/DX = CLI Buffer Address
Call Command Line Interp. RV: 0000h/0FFFFh
151 97 P_RPL IP: DS/DX = RPL Parameter Block Address
Call Resident Procedure Lib. RV: RPL return value
01h if RPL not found
ES = RPL return segment (if addr.)
152 98 F_PARSE IP: DS/DX = Parse Filename Control Block Addr.
Parse Filename RV: 0 if end of string, or
address of next item
0FFFFh if failure
153 99 C_GET IP: none

Get Console RV: AL: Console number
154 9A S_SYSDAT IP: none
Get SYStem DATa RV: SYSDAT Address -- Offset
segment address ES = SYSDAT Address -- Segment
155 9B T_SECONDS IP: DS/DX = TOD Address
Get Date and Time RV: none (TOD filled in)
156 9C P_PDADR IP: none
Return Process RV: Process Descriptor Address -- Offset
Descriptor Address ES = Process Descriptor Address -- Segment
157 9D P_ABORT IP: DS/DX = Abort Parameter Block Address
Abort Specified Process RV: 0000h/0FFFFh
158 9E L_ATTACH IP: none
Attach List Device RV: none
159 9F L_DETACH IP: none
Detach List Device RV: 0000h/0FFFFh
160 A0 L_SET IP: DL = List Device Number
Set List Device RV: none
161 A1 L_CATTACH IP: none
Conditionally Attach List RV: 0000h/0FFFFh
162 A2 C_CATTACH IP: none
Conditionally Attach Con. RV: 0000h/0FFFFh
163 A3 S_OSVER IP: none
Return Version Number RV: AH: Operating system type
AL: Current version number
164 A4 L_GET IP: none
Get List Device Number RV: AL: List Device Number

Appendix B: DR Net utilities summary

This appendix contains summaries of the utilities provided with DR Net. Table B-1 lists the programs, the manual in which the primary explanation is provided, and their function. The acronyms in the "Manual" column represent the following manuals:

UG -- "DR Net User's Guide"
SM -- "DR Net System Manager's Guide"

Table B-1. DR Net utilities

Format: Name, Manual
Description

ADDNET.CMD SM
Merge DR Net module with Concurrent CP/M system module.

LOCAL.CMD UG/SM
Remove the remote mapping of a local device in the Requester Configuration Table.

LOGOFF.CMD UG
Log off a server.

LOGON.CMD UG
Log on a server.

GENNET.CMD SM
Generate a DR Net server, requester, or server/requester for a system based on Concurrent CP/M.

GENRQR.CMD SM
Generate a DR Net requester for a system based on CP/M-86.

NAMES.CMD UG/SM
Display current entries in default server's NAMSVR.DAT file.

NAMESMOD.CMD SM
Change node ID number to name assignments on specified drive.

NET.CMD UG/SM
Map a local resource to a server in the requester Configuration Table.

NETLDR.CMD UG
Load and attach DR Net in CP/M-86 requester.

NETOFF.CMD UG
Detach requester based on Concurrent CP/M from the network.

NETON.CMD UG
Attach requester based on Concurrent CP/M to the network.

NETSTAT.CMD UG/SM
Display current local to remote device assignments and logged on.

Table B-2 lists all these utilities and their command syntax, and describes their options. All options are shown in lowercase in the syntax models.

Table B-2. DR Net utility command summary

Format: Utility
    Command syntax
    Summary

ADDNET.CMD
ADDNET
ADDNET filename.typ
ADDNET merges the DRNET.CMD system image file with the CCPM.SYS Concurrent CP/M system image. The option allows you to specify a file name under which the old CCPM.SYS is saved.

LOCAL.CMD
LOCAL ld:
LOCAL "lqueue"
LOCAL removes the remote mapping of the local resource specified in the command line. Local devices "ld:" are specified by name, where a local drive would be A: through P:, and a local list device would be LST0: through LST15:.

Quotation marks (") must be placed around queue names, so that the local knows not to translate the lowercase characters in the name to uppercase. All subsequent calls to the resource specified are referred to the local operating system for execution. All changes made with LOCAL are cancelled by NETOFF, or when the system is turned off.

LOGOFF.CMD
LOGOFF
LOGOFF nn::
LOGOFF nodename
LOGOFF logs off the current process and all ancestor processes from the specified server. The server to be logged off is designated in the command line, where a null entry indicates the default server, nn:: indicates a server by its hex ID number nn, and "nodename" indicates the server by name. Once the default server is logged off, only the ID number can be used to specify a node. Note that this command affects no other related or unrelated processes.

LOGON.CMD
LOGON
LOGON nn::
LOGON nodename
LOGON logs on the current process and all ancestor processes to the specified server. The server to be logged on is designated in the command line, where a null entry indicates the default server, nn:: indicates a server by its hex ID number nn, and "nodename" indicates a server by name. The third form is only available if name service has been implemented, and the operator is logged on the node's default server.

GENNET.CMD
The GENNET DR Net system generation program combines the NIOS.CMD file with either the RNET.CMD, SNET.CMD, or RSNET.CMD file to create the DRNET.CMD file which contains the DR Net system image. The program has an interactive portion that allows the system implementer or system manager to cast the initial mapping of local to remote drives, list devices, and queues, to set the node's ID number, and to define system and network dependent variables.

GENRQR.CMD
The GENRQR DR Net system generation program is used to create a requester-only DRNET.CMD system image file for CP/M-86 systems from NIOS.CMD and RNET.CMD files. The program is similar to GENNET in that it allows the system implementer to define interactively the hardware and network dependent variables, and to set the initial resource map. In CP/M-86 requesters, DR Net is loaded and the network attached with the NETLDR utility.

NAMES.CMD
NAMES
NAMES namespec
NAMES references the NAMSVR.DAT file on the node's default server to provide node number to name service. If the first form is invoked, the entire list of servers and requesters is displayed. The second form can be used with or without the "?" or "*" wilcard characters, to select a class of servers and requesters from the entire list.

NAMESMOD.CMD

This utility is used to create and modify the entries in the NAMSVR.DAT file. After NAMESMOD is loaded, a menu is displayed that allows you to create a new NAMSVR.DAT file, add a new node to an existing NAMSVR.DAT file, delete a node from this file, or display the contents of NAMSVR.DAT. All changes made are permanently recorded in the NAMSVR.DAT file.

NET.CMD
NET ld:=rd:
NET ld:=rd:nn::
NET ld:=rd:nodename
NET "lqueue"="rqueue"
NET "lqueue"="rqueue"nn::
NET "lqueue"="rqueue"nodename
NET is used to map a local resource to a server. The command line components shown here are defined as follows:

- ld:
  A local disk or list device to be mapped.

- rd:
  The remote replacement disk drive or list device.

- nn::
  The remote server expressed as a hex ID number.

- nodename
  The remote server expressed as a nodename.

- "lqueue"
  A local queue name to be mapped.

- "rqueue"
  The remote replacement queue name.

Note that the NET command for queues must include the quotation marks ("), and that the case of each character in the queue name is significant. The destination server is designated in the command line, where a null entry indicates the default server, nn:: indicates a server by its hex ID number nn, and "nodename" indicates a server by name. The third form is only available if name service has been implemented, and the operator is logged on the node's default server. All changes made by NET are cancelled when NETOFF is run, or when power is turned off.

NETLDR.CMD
Only requesters with the CP/M-86 operating system have use for NETLDR. This program reads the DRNET.CMD file created by GENRQR into memory, to load the DR Net system image and attach the node to the network. Only in the latter case can NETLDR be considered similar to NETON. There are no options available from this program. To detach from the network, CP/M-86 users must reset their computers.

NETOFF.CMD
Requester nodes running Concurrent CP/M use NETOFF to detach the computer from the network. This causes an automatic log-off to all servers before the detach

is effected. Before logging back on a server, the operator must attach his system first with NETON. There are no options available from this command.

NETON.CMD
requester and requester/server nodes running Concurrent CP/M use NETON to attach the system to the network. Although the DR Net system image is merged with the Concurrent CP/M system image, a process family has no network environment until NETON is run. This allows the server function to be available in a requester/server node while the requester function is implented. There are no options available from this program.

Note: NETON attaches an entire process family, from the current process on up its ancestral tree, if it finds a process that either has no parent or is already attached. In addition, once NETON is run, all child processes are automatically attached to the network.

NETSTAT.CMD
NETSTAT displays the current disk drive, list device, and queue map of local to remote resources, and lists all logged-on servers. The display has the following format:

Local = Remote on Node Logged On?

All local disk drives, list devices, and queues for which there is a map entry in the Requester Configuration Table are listed in the "Local" column. For each entry, the "Remote" column indicates the replacement resource, while "Node" contains the server ID number or name. The "Logged On?" column indicates whether the server is logged on or not. Underneath the display of the resource map, the logged servers are listed by ID number and name. The name only appears in the map, and in the list of logged-on servers when name service has been implemented and the node is logged on its default server.

Appendix C: Network message contents

Message output by DR Net requesters and servers have a fixed format. In turn, DR Net requesters and servers expect all messages received from the network to be in this same format. This appendix describes the DR Net message format and the message contents for all system calls that can be executed remotely.

C.1 DR Net logical message format

The DR Net logical message is the standard medium of exchange between requesters and servers. The requester module automatically gathers all pertinent data, and assembles the message into this format when a system call references a remote resource. All the information required by the server to perform the function is transferred in the DR Net.

The same format is used for servers' response messages. In this case, the return value and all data requested in the initial call are returned, so that the requester can present the response according to the calling convention.

Figure C-1 illustrates the format of the DR Net message. Each field is described in Table C-1.

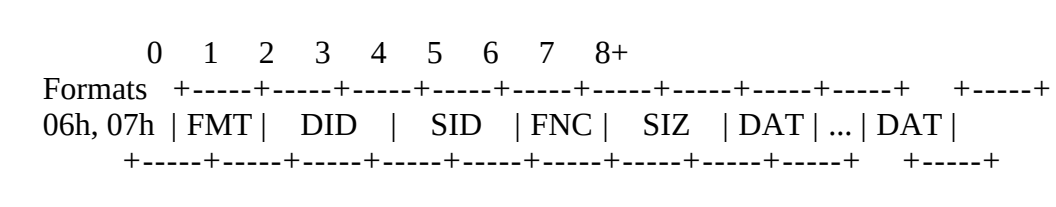


Figure C-1. DR Net logical message formats

Table C-1. DR Net logical message field descriptions

Format: Field
Description

FMT -- The format code
A message's format code indicates the number of bytes for each field, and whether the message is a request or a response. Table C-2 below lists the different FMT codes. Even-number FMT codes indicate the message is a request; off-number FMT codes indicate the message is a response.

DID -- Destination node ID number
The DID field is a 1- or 2-byte value that indicates the message's destination node, or its destination node and process. In a single-byte DID, only the destination's hexadecimal ID number is transferred. In a 2-byte DID, the node ID is recorded in the low-order byte, and the contents of the high-order byte depend upon whether the message is a request or a response. In a request, the high-order byte is meaningless. In a response, the high-order byte contains the requester process's arbitrarily assigned process ID.

SID -- Source node ID number
The SID field is a 1- or 2-byte value that indicates the message's source node, or its source node and process. In a single-byte SID, only the source node's hexadecimal ID number is transferred. In a 2-byte SID, the node ID is recorded in the low-order byte, and the contents of the high-order byte depend upon whether the message is a request or a response. In a request, the high-order byte contains the requester process's arbitrarily assigned process ID. In a response, this field is meaningless.

FNC -- System call function number
The FNC field is always a single byte, and contains the system call's hexadecimal number.

SIZ -- Data field size
The SIZ field is a 1- or 2-byte value that indicates the length of the data field that follows. This number is always the number of bytes in the field minus 1.

DAT -- Data field
The DAT field is variable length, and contains all the input values and data required by the server to perform the function, or all the return values and data returned by the function.

Table C-2. DR Net message formats

FMT Code	Number of bytes in DID	SID	FNC	SIZ	DAT	Description
00	1	1	1	1	1-256	CP/NET V1.2 Request
01	1	1	1	1	1-256	CP/NET V1.2 Response
02	1	1	1	2	1-65536	Not used (Request)
03	1	1	1	2	1-65536	Not used (Response)
04	2	2	1	1	1-256	Not used (Request)
05	2	2	1	1	1-256	Not used (Response)
06	2	2	1	2	1-65536	Concurrent CP/M Request
07	2	2	1	2	1-65536	Concurrent CP/M Response
08-127						Undefined
128-255						User definable

The format of the DAT field is also important to both the server and requester. Table C-3 lists the message contents for all system calls that can be networked.

C.2 Special terms in Table C-3, "Message contents"

The following characters, symbols, and terms are used in the following DAT field descriptions:

xx
Appears occasionally in the DAT field descriptions, to indicate that the value is irrelevant.

- EE -
Indicates that the function can return BDOS extended errors ("EE"), as well as extended network errors 0CFFh, 0DFFh, or 0EFFh. Any message can return an extended network error.

+
Appears as a suffix to the SIZ field value, and indicates that the value is variable. When it is used, consider the SIZ value shown to indicate the minimum.

Simulation Count

The largest number of 128-byte records that can fit in a single transaction. This value is calculated by the NDOS when function 44, F_MULTISEC, is called to reconcile the size of the DR Net message buffer (through which all messages flow to and from the network) with the length of the multisector data block. Divide the Multisector Data Block by the Simulation Count to obtain the number of network transactions required to complete a multisector data transfer.

Table C-3. DR Net message contents

System call	FMT	FNC	SIZ	DAT
* 3: C_RAWIN	0	03	0000	0000-0000 Server Console Number
	1	03	0000	0000-0000 Character Input
* 4: C_RAWOUT	0	04	0001	0000-0000 Server Console Number
				0000-0001 Raw Character to Output
	1	04	0000	0000-0000 00
5: L_WRITE	0	05	00nn	0000-0000 Server List Number
- EE -				0001-00nn Characters To Be Listed
	1	05	0000	0000-0000 00
				Note: nn = 01h to 80h.
* 11: C_STAT	0	0B	0000	0000-0000 Server Console Number
	1	0B	0000	0000-0000 Console Status Byte

* These functions are supported by Concurrent CP/M servers for CP/NET Version 1.2 compatibility only. They are never generated by a DR Net requester.

14: DRV_SET	0	0E	0000	0000-0000 Selected Disk
- EE -	1	0E	0000	0000-0000 Return Code
15: F_OPEN	0	0F	002C	0000-0000 User Number
				0001-0024 FCB
- EE -				0025-002C Password
	1	0F	0024	0000-0000 Directory/Return Code
				0001-0024 Opened FCB
16: F_CLOSE	0	10	002C	0000-0000 User Number
				0001-0024 FCB
- EE -				0025-002C Not Used
	1	10	0024	0000-0000 Directory/Return Code
				0001-0024 FCB
17: F_SFIRST	0	11	0025	0000-0000 Current Disk
				0001-0001 User Number
- EE -				0002-0025 Search FCB
	1	11	0020	0000-0000 Directory/Return Code
				0001-0020 Directory Entry

Note: "Current Disk" is valid only when there is
a "?" in the search FCB drive byte field.

18: F_SNEXT 0 12 0001 0000-0000 xx
- EE - 1 12 0020 0000-0000 Directory/Return Code
0001-0020 Directory Entry

19: F_DELETE 0 13 002C 0000-0000 User Number
0001-0024 FCB
0025-002C Password
- EE - 1 13 0000 0000-0000 Directory/Return Code

20: F_READ 0 14 0024 0000-0000 User Number
0001-0024 FCB
- EE - 1 14 00A5+ 0000-0000 Return Code
0001-0024 FCB
0025-xxxx Data That Was Read
yyyy-yyyy Multisector Transfer
Return Code

Note: xxxx = 00A4 + (80h * (Simulation Count - 1))
yyyy = xxxx + 1

21: F_WRITE 0 15 00A4+ 0000-0000 User Number
0001-0024 FCB
- EE - 0025-xxxx Data To Be Written
1 15 0025 0000-0000 Return Code
0001-0024 FCB
0025-0025 Multisector Transfer
Return Code

Note: xxxx = 00A4 + (80h * (Simulation Count - 1))

22: F_MAKE 0 16 002D 0000-0000 User Number
0001-0024 FCB
- EE - 0025-002C Password
002D-002D Password Mode Field
1 16 0001 0000-0000 Directory/Return Code
0001-0024 FCB

23: F_RENAME 0 17 002C 0000-0000 User Number
0001-0024 FCB in RENAME Format
- EE - 0025-002C Password
1 17 0000 0000-0000 Directory/Return Code

24: DRV_LOGINVEC 0 18 0000 0000-0000 xx
1 18 0001 0000-0001 Log-in Vector

27: DRV_ALLOCVEC 0 1B 0000 0000-0000 Current Disk
- EE - 1 1B nnnn 0000-nnnn Allocation Vector

Note: The size of the allocation vector is
dependent on the current disk's DPB.

```
28: DRV_SETRO        0   1C  0000    0000-0000 Current Disk
                 1   1C  0000    0000-0000 00

29: DRV_ROVEC        0   1D  0000    0000-0000 xx
                 1   1D  0001    0000-0001 Read/Only Vector

30: F_ATTRIB         0   1E  002C    0000-0000 User Number
                           0001-0024 FCB with File Attributes
  - EE -                   0025-002C Password
                 1   1E  0000    0000-0000 Directory/Return Code

31: DRV_DPB          0   1F  0000    0000-0000 Current Disk
  - EE -         1   1F  000F    0000-000F Disk Parameter Block

33: F_READRAND       0   21  0024    0000-0000 User Number
                           0001-0024 FCB
  - EE -         1   21  00A5+   0000-0000 Return Code
                           0001-0024 FCB
                           0025-xxxx Data That Was Read
                           yyyy-yyyy Multisector Transfer
                                 Return Code

                 Note: xxxx = 00A4 + (80h * (Simulation Count - 1) )
                      yyyy = xxxx + 1

34: F_WRITERAND      0   22  0024    0000-0000 User Number
                           0001-0024 FCB
  - EE -                   0025-xxxx Data To Be Written
                 1   22  0024    0000-0000 Return Code
                           0001-0024 FCB

                 Note: xxx = 00A4 + (80h * (Simulation Count - 1) )

35: F_SIZE           0   23  0024    0000-0000 User Number
                           0001-0024 FCB
  - EE -         1   23  0024    0000-0000 Return Code
                           0001-0024 FCB

36: DRV_RANDREC      0   24  0024    0000-0000 User Number
                           0001-0024 FCB
  - EE -         1   24  0024    0000-0000 Return Code
                           0001-0024 FCB

37: DRV_RESET        0   25  0001    0000-0001 Drive Vector
                 1   25  0000    0000-0000 Return Code

38: DRC_ACCESS       0   26  0001    0000-0001 Drive Vector
  - EE -         1   26  0000    0000-0000 Return Code

39: DRV_FREE         0   27  0001    0000-0001 Drive Vector
                 1   27  0000    0000-0000 Return Code

40: F_WRITEZF        0   28  00A4+   0000-0000 User Number
                           0001-0024 FCB
```

```
 - EE -                  0025-xxxx Data To Be Written
            1  28  0024   0000-0000 Return Code
                          0001-0024 FCB

            Note: xxxx = 00A4 + (80h * (Simulation Count - 1) )

42: F_LOCK        0  2A  0026   0000-0000 User Number
                          0001-0024 FCB
 - EE -                   0025-0026 File ID
            1  2A  0024   0000-0000 Return Code
                          0001-0024 FCB

43: F_UNLOCK      0  2B  0026   0000-0000 User Number
                          0001-0024 FCB
 - EE -                   0025-0026 File ID
            1  2B  0024   0000-0000 Return Code
                          0001-0024 FCB

44: F_MULTISEC    0  2C  0001   0000-0000 Multisector Count Requested
                          0001-0001 Count Used for Simulating
                                    Multisector I/O Across
                                    Network
            1  2C  0000   0000-0000 Return Code

46: DRV_SPACE     0  2E  0000   0000-0000 Drive ID
            1  2E  0003   0000-0000 Return Code
 - EE -                   0001-0003 Number of Free Records

64: N_LOGON       0  40  0012   0000-0007 Server Password
                          0008-0009 Process ID
                          000A-0011 BDOS Default Password
                          0012-0012 Compatibility Attributes
                          0013-0013 Version
                          0014-0014 First Log Flag
            1  40  0001   0000-0000 Return Code
                          0001-0001 Reserved for System Use

            Note: The Process ID is a number arbitrarily
                  assigned by the requester. It is not
                  the process descriptor address.

65: N_LOGOFF      0  41  0009   0000-0007 Unused
                          0008-0009 Process ID
            1  41  0000   0000-0000 Return Code

            Note: The Process ID in this message is a
                  number arbitrarily assigned when the
                  process logged on.

70: N_ATTRIB      0  46  0000   0000-0000 Compatibility Attributes
            1  46  0000   0000-0000 xx

            Note: For CP/M-86 requesters only.
```

```
71: N_SCT          0   47  0000    0000-0000 xx
               1   47  004C    0000-0000 Server Temporary File Drive
                           0001-0001 Server Network Status Byte
                           0002-0002 Server Node ID
                           0003-0003 Maximum Possible Req.
                           0004-0004 Num. of Req. Logged On
                           0005-0006 Requester Log-on Vector
                           0007-0017 IDs of 16 Requesters
                           0017-0018 Requester Log-on Vector
                           0019-0028 IDs of 16 Requesters
                           0029-002A Requesters Log-on Vector
                           002B-003A IDs of 16 Requesters
                           003B-003C Requester Log-on Vector
                           003D-004C IDs of 16 Requesters

75: N_BUFSIZ       0   4B  0001    0001-0001 Requested Message Buf. Size
  (For system      1   4B  0002    0000-0000 Return Code
   use only.)                     0001-0002 Maximum Allowable Message
                                    Buffer Size

77: N_KEEPALIVE    0   4D  0000    0000-0000 xx

               Note: This message is unique in that there
                   is no return message.

99: F_TRUNC        0   63  0024    0000-0000 User Number
                           0001-0024 FCB
  - EE -           1   63  0024    0000-0000 Directory/Return Code
                           0001-0024 FCB

100: DRV_SETLABEL  0   64  0034    0000-0000 Unused
                           0001-0024 Directory Label FCB
  - EE -                     0025-002C Old Default Password
                           002D-0034 New Default Password
               1   64  0000    0000-0000 Return Code

101: DRV_GETLABEL  0   65  0000    0000-0000 Drive ID
  - EE -           1   65  0000    0000-0000 Directory Label Byte

102: F_TIMEDATE    0   66  0024    0000-0000 User Number
                           0001-0024 FCB
  - EE -           1   66  0024    0000-0000 Directory/Return Code
                           0001-0024 XFCB

103: F_WRITEXFCB   0   67  0034    0000-0000 User Number
                           0001-0024 XFCB
  - EE -                     0025-002C Old Password
                           002D-0034 New Password
               1   67  0000    0000-0000 Directory/Return Code

105: T_GET         0   69  0000    0000-0000 xx
               1   69  0004    0000-0001 Days since 31 DEC 1977
                           0002-0002 BCD Hours
                           0003-0003 BCD Minutes
```

0004-0004 BCD Seconds = 00

106: F_PASSWD 0 6A 0007 0000-0007 Default Password To Be Set
1 6A 0000 0000-0000 Return Code

134: Q_MAKE 0 86 001B 0000-001B Queue Descriptor
1 86 0003 0000-0003 Return Code (AX and CX)

135: Q_OPEN 0 87 000F 0000-000F Queue Parameter Block
1 87 002E 0000-0003 Return Code (AX and CX)
0004-0013 Queue Parameter Block
0014-002E Queue Descriptor

136: Q_DELETE 0 88 001B 0000-001B Queue Descriptor
1 88 0003 0000-0003 Return Code (AX and CX)

137: Q_READ 0 89 000F 0000-000F Queue Parameter Block
1 89 0003+ 0000-0003 Return Code (AX and CX)
0004-xxxx Queue Message

Note: The queue message length cannot exceed
the maximum message buffer size.

138: Q_CREAD 0 8A 000F 0000-000F Queue Parameter Block
1 8A 0003+ 0000-0003 Return Code (AX and CX)
0004-xxxx Queue Message

Note: The queue message length cannot exceed
the maximum message buffer size.

139: Q_WRITE 0 8B 000F+ 0000-000F Queue Parameter Block
0010-xxxx Queue Message
1 8B 0003 0000-0003 Return Code (AX and CX)

Note: The queue message length cannot exceed
the maximum message buffer size.

140: Q_CWRITE 0 8C 000F+ 0000-000F Queue Parameter Block
0010-xxxx Queue Message
1 8C 0003 0000-0003 Return Code (AX and CX)

Note: The queue message length cannot exceed
the maximum message buffer size.

158: L_ATTACH 0 9E 0000 0000-0000 Server List Device
1 9E 0003 0000-0003 Return Code (AX and CX)

159: L_DETACH 0 9F 0000 0000-0000 Server List Device
1 9F 0003 0000-0003 Return Code (AX and CX)

161: L_CATTACH 0 A1 0000 0000-0000 Server List Device
1 A1 0003 0000-0003 Return Code (AX and CX)

EOF