XXDP

Notes for XXDP+ and XXDP V2 Operating Systems

Pete Turnbull, February 1993

XXDP - Diagnostic Program Monitor

XXDP is a small, compact operating system designed to provide a suitable environment across the range of DIGITAL Q-bus and Unibus machines to run diagnostic (test and maintenance) software. The name comes from the four-character names given to diagnostic programs. XXDP is not intended to provide a software-development environment, this is usually done under, for example, RSX11.

Early versions were quite simple and consisted of a monitor program such as RKDP.SAV (the monitor for RK05-based systems) and a few utilities. Later versions, called XXDP+, added a few more operations and used monitors with names of the form HMddvp.SYS (HM signifies a monitor, *dd* is the device mnemonic, and *v* and *p* are the version and patch levels).   XXDP+ could tell what kind of bus system it was running on, unless it was an 11/23 or 11/24. XXDP+ Version 1.1, sometimes called XXDP++ or XXDP Super-plus, was basically similar but with some enhancements, notably the introduction of Diagnostic Runtime Services which provided a common supervisor and setup functions for compatible diagnostics. XXDP Version 2 added memory management, English-type commands in addition to the single-letter monitor commands, and device drivers separate from the monitor program itself.

Naming Conventions:

XXDP.???	where XX is a device mnemonic, and DP stands for Diagnostic Program, is used on oldest versions for monitor and bootstrap programs (eg RKDP.SAV), and for directory listings (eg RLDP.DIR). It is also used for distribution media (eg RKDP is XXDP on an RK05 disk pack).

H*xmnrp*.SYS	used for XXDP+ system files. H means a system file; *x* may be M,D,U,S,Q representing Monitor, Device Driver, Utility, Supervisor, or miscellaneous files; *mn* is the device mnemonic for the medium supported; *rp* specifies the revision and patch levels. Thus HMRLA1.SYS is the system monitor for an RL01/RL02 system, first issue, patch level 1. Special mnemonics include DI (directory utility), SU (setup utility), AA (XXDP supervisor file), AB (PT/AMS supervisor file), SA (user manual).

tmnirp.BI?	used for XXDP+ (and later) diagnostics, also some .OBJ files.

t	processor type code, eg V
mn	device mnemonic, eg RQ
i	a unique program identifier, eg C
rp	revision and patch level, eg B2

Thus ZRQCH0.BIC is for any processor, using an RQDX controller, the third diagnostic of a set (this one is a disk formatter), eighth version, never patched or updated.

D*tirp*.BIN	used for Unibus tests; *t*=type of test, otherwise as above

0	instruction test
1	addressing test
6	AA11/VT06/LAB11 test
8	Unibus test

Standard Files:

COPY.BIN	XXDP+ copy program
DATE.SYS	XXDP Ver.2 date utility
DRSSM.SYS	XXDP Ver.2 Diagnostic Runtime Services supervisor for use under the small monitor
DRSXM.SYS	XXDP Ver.2 DRS supervisor for use under the extended monitor
DIR.SYS	XXDP Ver.2 directory utility
DUSZ.SYS	XXDP Ver.2 ????
DXCL.BIN	DEC/X11 (automated testing) configurator and linker
HELP.TXT	Help Text file
MN.SYS	XXDP Ver.2 device driver for device *MN*, eg DL.SYS for RL01/2
PATCH.BI?	program to patch diagnostics
SETUP.BI?	Setup utility for DRS-compatible diagnostics
UPD*n*	XXDP+ Update utility, *n* is version 1 or 2 (1 is obsolete, smaller and less useful)
UPDAT.BIC	XXDP Ver.2 Update utility
XTECO.BI?	simple text editor
XXBLD.BI?	utility to build new media containing XXDP system
XXDPSM.SYS	XXDP Ver.2 small monitor (runs in 28KW or less)
XXDPXM.SYS	XXDP Ver.2 extended monitor, has 22-bit memory management and can use up to 124KW memory
*n*DIREC.TRY	directory listing file, *n*=1..2..3...

Standard File Types:

.BAK	backup file created by XTECO
.BIC	chainable binary image file
.BIN	binary image file
.CCC	chain file (a batch file containing textual commands)
.DIR	directory listing file, as created by DIR utility
.LIB	library file
.MPG	???
.OBJ	DEC/X11 object code
.SAV	memory image file (usually executable code)
.SYS	system file
.TRY	directory text file
.TXT	ASCII text file

Characters and Wildcards:

As XXDP stores filenames in RADIX-50 format, the only legal characters in filenames and extensions are upper-case letters A...Z, and numerals 0...9. The dollar symbol "$" and full-stop or period "." have special significance and should not be used, except that a full-stop is used to separate a filname from its extension. Spaces are not allowed.

In most situations, XXDP supports the use of wildcard characters in filenames.

?	represents a single character
*	represents any number of characters

Processor Types:

A	11/05, 11/15, 11/20
B	11/40
C	11/45
D	GT40, general Unibus
E	11/70
F	11/34
G	11/04
H	used for system files
J	11/23, 11/24
K	11/44
M	MNC-11
N	System Industrie 9400 controller diagnostics
P	PDT-11, Plessey diagnostics
Q	11/60
R	LPA-11
T	.MPG files
V	11/03
X	DEC/X11 OBJ files
Z	any processor

Device Mnemonics used in Diagnostics:

AA	AA11, AAV11
AD	AD11, ADV11, AD01
AF	AFC11
AR	AR11
AX	AXV11
BB	KIT11
BM	BM873, BDV11
BT	Bus Tester
CB	CB11
CD	CD11, CDS11
CL	CL11
CM	CM11
CP	DEC/X11 CPU
CT	CT11
CR	CR11
DC	DC11
DF	DFC11
DH	DH11, DHV11
DJ	DJ11
DL	DL11, DLV11
DM	DM11, DM*x*11
DN	DN11
DP	DP11, DUP11, DPV11
DQ	DQ11, DQV11
DR	DR11, DRV11
DT	DT07, DTE20
DU	DU11, DUV11
DV	DV11, DLV11
DX	DX11
DZ	DZ11, DZV11
FP	FP11, FPF11
GT	GT40, GT44
IB	IBV11
IR	ICR11
IT	ITEP
KA	LSI-11 CPU
KB	CPU
KC	CPU
KD	PDT11, 11/23, 11/24 CPU
KE	CPU
KG	CPU
KH	KIT11
KL	KL11
KK	Cache Memory
KM	KMC11, Memory
KP	KPV11
KT	Memory Management
KU	KUV11
KX	KXT11
KW	KW11, KWV11

Device Mnemonics used in Diagnostics (continued):

LA	LA11, LC11, DEC Printers
LC	LC11
LK	LK11
LP	LP11, LP*x*11
LQ	LQP
LS	LS11
LV	LV11
M8	BDV11
M9	M9301, M9312, REV11, TEV11
MF	Memory
MJ	Memory
MK	Memory
ML	Memory
MM	Memory
MN	MNC11
MR	MR11, MRV11-B
MS	Memory
MX	MXV11
NC	NC11, NCV11
PA	PA611, TYP11
PC	PC11, PCS11, IPV11/12
PL	PCL11
PM	PDM70
QE	CPU
QM	Memory
QK	CPU
QU	DEC/X11
R6	RK611, RK06/07
RC	RC11, KLESI-QA, RC25
RF	RF11
RH	RH70
RJ	RP04/05/06
RK	RK11, RKV11, RK05
RL	RL11, RLV11/12, RL01/02
RM	RH11, RH70, RM02/03
RP	RP11, RP02/03
RQ	MSCP disk controllers and drives
RS	RH11, RS03/04
RT	RT01/02
RX	RX11, RXV11, RXV21, RX01/02
TA	TA11
TC	TC11
TE	TE16, TU77, TM03
TK	TKQ50, TK50/75
TM	TM11, TMA11, TE10
TR	TR79F
TS	TS11, TS03, TSV05
TU	TM02, TU10, TU58
UD	UDC11, UDA50, KDA50-Q, RA60/80/81
VS	VS11, VSV11, VS60
VT	VT
XY	XY11

Note that these are not the same as standard DIGITAL device-driver mnemonics.

HELP for XXDP V2 **(HELP.TXT Rev: 6.0)**

LIST>

The commands outlined here are for XXDP V2 only!

LP: = line printer
TT: = Console terminal
SM: = Small monitor
XM: = Extended monitor

HELP is available on the following topics:

BOOT BOOTSM CHAIN CLEAR
COPY DATE DEFSM DEFXM
DELETE DIRECTORY ENABLE HELP
INFONBOOT INITIALIZE LOAD NOTES
PRINT RENAME RUN SET
START SMALLMON SWITCHES TYPE
UPDAT V2.4 VERSION

INFONBOOT>

XXDP V2 can boot two monitors (XXDPSM or XXDPXM). The “SM” or Small monitor is for systems that are 28k or smaller in size, the “XM” or Extended monitor is for systems larger than 28k. The “SM” monitor works like the monitor of XXDP+ V1 and will run all programs written for V1. The “XM” monitor offers the new features advertised in XXDP V2. By default you will boot up the “XM” monitor provided the system has enough memory and memory management capability.

Start-up file BOOT.CCC

At boot time, a start-up file BOOT.CCC will be executed as a chain file. If this file does not exist, there will be no warning message, boot will continue and by default the “XM” monitor will be booted up. If the file exists, it must contain one of the two commands listed below. Only one of the two commands may be used at a time and it should be on the first line of the file with no leading spaces. If the file does not contain one of the two commands it will boot up the “XM” monitor. Note, if the system does not have more than 28K worth of memory or memory management capability the “SM” monitor will be booted up.

SM	Boots the “SM” monitor.
QUIET	Boots “XM” monitor in UFD mode. (SYSTEM.CCC must exist)

SMALLMON>

SMALL MONITOR (XXDPSM)

The following are legal commands :   (format of each command is shown)

C   run a batch job (chain)
C filename.CCC[/switches]

D   list directory of load medium
D[/L][/F]   “/L” switch = on a LP:
“/F” switch = in a short form.

DAte   inspect or set the current system date.
(For more information see “HELP DATE”)

E   enable alternative drive for system device
E *n*   where *n* is the new drive number

H   type help information about XXDP V2
H[/L]   “/L” = print on a LP:

L   load a program
L filename[.ext]
Example:
.L DIAG   (load DIAG.BI?)
.L ZDJCA2.NEW   (load ZDJCA2.NEW)

R   run a program
R filename[.ext] [addr]

S   start a program
S [addr]

VT   toggles the console terminal type between video and hard copy
(The default terminal type is hard copy)

EXTENDED_MON>

EXTENDED MONITOR (XXDPXM)

The following are legal commands :   (format for each command is shown)
(Letters shown in upper case are required to make a command unique)

Boot   directs monitor to boot another XXDP device
BOOT dev:

BOOTSm   directs monitor to boot up the small monitor.
BOOTSM

COpy   transfer files and maintenance operations
COpy   [/BOOT] input-filespec output-filespec
[/FILES]
[/DEVICE]
[/DELETE]
Examples:
.COPY A.BIN DY0:   (Copy A.BIN from system device to DY0:)
.COPY/BOOT DM0: DM1:   (Copy monitor from DM0: to DM1:)
.COPY/FILES DL0: DL1:   (Copy all files from DL0: to DL1:)
.COPY/DEVICE DY0: DY1:   (Copy device image from DY0: to DY1:)

Chain   execute a batch, or chain file.   The file must have extension .CCC
CHAIN filespec[/switches]

CLear   Clears the SM and XM flags.   When these flags are cleared they allow the monitor to check bit 12 of location 52 before running any program.   If bit 12 is set the program will be run under the extended monitor. If bit 12 is not set then the program will be run under the small monitor.

Note :- On boot up (by default) these flags are cleared and can be set by issuing the DEFSM or DEFXM command only.

DATE   inspect or to set the current system date
DATE dd-mmm-yy   (Default dates are 01-JAN-84 for SM)
(01-JAN-87 for XM)
where
dd   = day (a decimal number from 1 to 31)
mmm = month, first three characters
yy   = year (a decimal number from 83 to 99)
Example (setting date):
.DATE 18-MAY-83
Example (obtaining date):
.DATE
18-MAY-83

DElete   deletes the file(s) that are specified.

```
DElete[/NOREWIND] filespec
      [/NONAMES ]
Example:
.DELETE DU0:ABC.BIN
.DELETE/NONAMES DU0:ABC?.BIN          (No file names will print)
.DELETE/NOREWIND MU0:*.TXT       (No tape rewind between files)
```

Directory list all files on a XXDP device.

```
DIRECTORY[/PRINTER]
         [/FAST]
```

where
/PRINTER = print on LP:
/FAST = show in short form

Enable enable alternative drive for system device

E *n* where *n* is the new drive number

Help types HELP.TXT which contains help on XXDP V2

```
H[/PRINTER]
```

Help	Displays topic list
Help ?	Displays topic list
Help *	Displays help on all Topics
Help Topic	Displays help on that Topic
Help To*	Displays help on all topics starting with the letters “To”

Note : To exit help, simply type in EXit or a carriage return.

INItialize initialize device directory into XXDP format.

INITIALIZE device: CAUTION: All data on device is lost!

Load load a file into memory

```
LOAD filespec
Example:
.LOAD ZKXCA0.BIC                                (load DIAG.BI?)
```

NOTE: This command is used to load programs whose extensions are .BIC, .BIN or .SYS only.

Print Print contents of files on the specified device on LP:

```
PRINT[/NOREWIND] filespec
Example:
.PRINT DY0:SYSTEM.CCC
.PRINT/NOREWIND MS1:*.TXT
```

REname change the file specification of an existing file

```
RENAME input-filespec output-filespec
```

Example:

```
.RENAME DX1:DIAG.OLD=DX1:DIAG.BIN
```

Run load and start a program that is stored on system device.

```
RUN filespec    [addr]
```

Examples:

```
.RUN UPD2                                        (load/start UPD2.BIN)
.RUN SAMPLE.XXX                              (load/start SAMPLE.XXX)
.RUN FXDIAG 204                  (load/start FXDIAG.BI? at location 204)
```

Set changes device characteristics and system parameters.

```
SET [device]:condition
      [ item ]
```

where “device” = device whose characteristics are modified.
“item” = system parameter that needs to be modified.

Examples:

```
.SET TT:SCOPE              (RUBOUT will delete characters - XM Default)
.SET TT:NOSCOPE  (RUBOUT will echo deleted characters - SM Default)
.SET TT:QUIET       (Prevent system from echoing lines from a chain file
                     or from diagnostics that are running from a chain
                    file)                                (Default = NOQUIET)
```

DEFSm Sets the SM flag. After issuing this command all programs will be run under the small monitor until the CLEAR command is issued. Bit 12 of location 52 will not be checked at all and by default any program(s) run after this command is issued will be run under SM.

Note of caution :- If you plan to run only one program under SM and issue this command make sure you issue the “CLEAR” command after your program has completed running. Else by default every program will run under SM regardless of whether bit 12 was set or not.

DEFXm Sets the XM flag. After issuing this command all programs will be run under the extended monitor until the CLEAR command is issued. Bit 12 of location 52 will not be checked at all and by default any program(s) run after this command is issued will be run under XM.

Note of caution :- If you plan to run only one program under XM and issue this command make sure you issue the “CLEAR” command after your program has completed running. Else by default every program will run under XM regardless of whether bit 12 was set or not.

Start start a file that has been loaded into memory via LOAD

```
START [addr]                  A starting address may be entered
```

Type prints the contents of a file on the terminal
TYPE[/NOREWIND] filespec
Example:
.TYPE/NOREWIND *.TXT

Version Prints information about the Extended monitor.
Version

V2.4>

There are some major differences in V2.4 and all of the previously released versions of XXDP V2. The main difference is the way programs will be run from now on.

In previous versions, programs were run under the same monitor that was displayed on boot up, i.e. if you booted the small monitor you ran all your programs under the small monitor. If you booted the extended monitor you ran all programs under the extended monitor. The problem with this was that you would have to reboot the other monitor if you had a program that ran on one and did not run on the other.

In V2.4 (and following versions), this problem is solved, programs will be run after a check is made on bit 12 location 52 (after the program has been loaded) and if this bit is set the program will be run under the extended monitor. If this bit is not set, the program will run under the small monitor. Unlike the previous versions there will be no need to reboot.

Bit 12 of location 52 should be set by using the “UPDAT” or “PATCH” utility. If UPDAT is used you might have to change the LOCORE of your program. Please sure the “CLR” command is issued before loading the program and changing the locore.

Note : Look at the “DEFSM”, “DEFXM” and the “CLEAR” commands in this help file if you are unable to set bit 12 by using the utilities.

SWITCHES>

Switches - are not applicable to all commands
- these switches are not system defaults
/PRINTER = print on LP:
/NONAMES = No file names will be shown
/NOREWIND = prevents tape rewinding between files

UPDAT>

UPDAT program

This program uses the same commands as UPD2. Commands SAVM and SAVE are no longer supported. The following command has been added to build bootable media. It will work with tapes or disks.
CREATE DY0: (will create a bootable DY from your system media)

NOTES>

CAUTION: Do NOT use XXDP+ V1 to write to XXDP V2 media.
XXDP V2 may read or write to XXDP+ V1.

NOTE: The monitor only sizes to 124k words.

For a detailed description of these commands see XXDP/DRS User Manual

DRS - Diagnostic Runtime Services

This is a sort of supervisor which oversees the running of diagnostics (except some old ones). It provides a standard format for dialogue etc, and a standard set of control functions and commands. The prompt is **DR>**, issued when a diagnostic is RUN from the monitor. Commands may be truncated to three characters and some may be modified by three-character switches. DRS deals with UUTs (Units Under Test) each of which is assigned a logical unit number (0-63). Each diagnostic is controlled by a hardware parameter table and a software parameter table, is divided into one or more TESTs, and will be executed in one or more PASSes. These tables can be pre-set by the SETUP utility, otherwise a series of questions will be asked after the diagnostic is STArted. Many questions have defaults (diagnostic-dependant) which will be displayed immediately before the question mark; all will show what kind of response is required by a single letter in parentheses.

Question types:

(B)	requires a binary number
(O)	requires an octal number
(D)	requires a decimal number
(A)	requires an ASCII character or string
(L)	requires a logical (Y or N) response

Error Messages

Errors are displayed at three levels. The header level shows only:

Z*NAME* *TYP* ERR *eeeee* ON UNIT *n* TST *ttt* SUB *sss* PC: *ppppp*

where Z*NAME* is the name of the diagnostic, *TYP* is the type of error (HaRD, SoFT), *eeeee* is a five-digit error code number (not a total), *n* is the unit number, *ttt* is the three-digit test number, *sss* is subtest or section number, *ppppp* is the value of the program counter at time of the error. The basic level adds an additional line of descriptive text, eg:

REGISTER FAILED TO CLEAR AFTER BUS RESET

and finally the extended level provides both of the above and also a line of supporting information such as CSR contents.

Commands:

STA[RT] reloads the trap catcher, initialises the diagnostic, clears all flags and runs all tests on all units, normally first asking the user to change the hardware and software tables

RES[TART] like START, but does not reload the trap catcher, may not fully initialise the diagnostic, and asks no hardware table questions

CON[TINUE] restarts the test (not the whole diagnostic) which stopped on an error, or was stopped by CTRL-C from the operator, for any remaining passes/units. Does not initialise the diagnostic nor ask hardware table questions but may ask software table questions

PRO[CEED] resumes testing from the point at which it halted on error (not in response to CTRL-C), for any remaining passes/units, without any initialisation at all

DRO[P] drops unit(s) from the test list. Drops all units unless /UNITS switch is used

ADD adds unit(s) to the test list. Adds all de-activated units unless the /UNITS switch is used

DIS[PLAY] shows hardware table parameters for units in the test list

FLA[GS] shows current status of all flags. Takes no parameters

ZFL[AGS] clears all flags to zeros. Flags are set by using /FLAGS switch with START, RESTART, CONTINUE or PROCEED

Switches:

/TES[TS]: followed by a list of test numbers to be executed. Separate test numbers with colons

/PAS[S]: followed by a decimal number in the range 1-65536, being the number of times each test should be executed

/FLA[GS]: followed by a list of flags to be set. Separate flags with colons

/EOP: followed by a decimal number in the range 1-65536, being the number of passes after which an "End of Pass" message is to be printed, giving the number of passes completed and the number of errors found

/UNI[TS]: followed by a list of unit numbers to be included. Separate unit numbers by commas, or describe a range by giving first and last members separated by a dash

Valid combinations of switches and commands:

	/TESTS	/PASS	/FLAGS	/EOP	/UNITS
START	yes	yes	yes	yes	yes
RESTART	yes	yes	yes	yes	yes
CONTINUE	.	yes	yes	yes	.
PROCEED	.	.	yes	.	.
DROP	.	.	.	.	yes
ADD	.	.	.	.	yes
PRINT	.	.	.	.	.
DISPLAY	.	.	.	.	yes
FLAGS	.	.	.	.	.
ZFLAGS	.	.	.	.	.
EXIT	.	.	.	.	.

Flags:

HOE Halt On Error - the diagnostic will return to DR> level if an error is detected

LOE Loop On Error - the diagnostic will loop continually through the sub-test which found the error, even if the error condition clears (allowing testing for intermittent errors), until the operator types CTRL-C

IER Inhibit Error Reports - prevents error messages (except some essential system error messages such as Illegal Interrupt) being typed. Does not affect end-of-pass messages

IBE Inhibit Basic Error reports - print only header messages

IXE Inhibit eXtended Error reports - print only basic error messages

PRI PRInt directly to lineprinter - send all messages except command messages to lineprinter instead of console

PNT Print Number of Test - print test number as it executes

BOE Bell On Error - sound bell if an error is detected

UAM UnAttended Mode - suppresses requirement for operator intervention, but may also suppress some testing

ISR Inhibit Statistical Reports - not all diagnostics support statistical reports anyway

IDR Inhibit DRopping of units - don't drop a unit from the test list if it generates an error. If not used, a UUT may be dropped if it reaches an error threshold (more than a pre-set number of errors) or generates a serious error.

ADR execute AutoDRop code - causes diagnostic to test for "device available" or "device ready". Not all diagnostics support this

LOT Loop On Test - causes DRS to continually execute tests in the test list, without re-executing initialisation or end-of-pass code

EVL use EVaLuation code - diagnostic-specific interpretation (or none)

UPD2 / UPDAT

File manipulation utility, which loads into the bottom part of memory. Used for building XXDP media, copying, loading, modifying files, etc.

Commands:

DIR
directory of device
DIR [dev:[outfile][/Q]] [dev:][filespec][/Q][/F}[/B][/L]

PIP
copy file(s) or device - with no autodelete/overwrite
PIP [dev:][outfilespec][/Q]=[dev:][infilespec][/Q][/N]
eg PIP MM0:FILE??.= copies all files from system device to MM0:, renaming them so they all begin "FILE".*

FILE
like PIP, but no rename possible, and will autodelete/overwrite
FILE dev:[/Q]=[dev:][infilespec][/Q][/N]

DEL
delete file
DEL [dev:]filespec[/Q][/N]

REN
rename file
REN [dev:]newfilespec=[dev:]oldfilespec

CLR
clear buffer (sets all buffer locations to zero)
CLR

LOAD
loads binary file into buffer
LOAD [dev:]file[/Q][/N]
You can use wildcards, but this will load each matching file in turn, one on top of the other!
/N = inhibit printing LOCORE, HICORE, and filename

MOD
modify contents of address, works rather like ODT. Linefeeds to move to next location, and null entries, are allowed.
MOD mmmmmm<CR>

LOCORE
set lower memory limit used by program buffer

HICORE
set upper memory limit used by program buffer

XFR
sets transfer address (load address) of program in buffer

DUMP
saves memory image between LOCORE and HICORE
DUMP [dev:]filename[/Q]

ZERO
initialise medium (creates empty XXDP directory)
ZERO dev:

COPY
copies entire medium, block (image) mode or file mode
COPY dev:=dev:[/I | F] (default is /I)

CREATE	*UPDAT only:* save monitor file (bootstrap) to disk or tape CREATE dev:
SAVE / SAVM	*UPD2 only:* saves monitor file to disk (SAVE) or tape (SAVM) SAVE dev: SAVM dev:
ASG	assign logical unit to number to device ASG dev:=n
DO	execute indirect command file. Such files cannot contain the EXIT command. Files can contain comments, if the first character on the line is ";" or "$" the line is merely printed. If "$" is used, operation is suspended until the operator types CTRL-X DO [dev:]filename
READ	reads file to check validity (ie a verify command) READ [dev:]filespec[/Q][/N}
EOT	writes logical EOT marker on a tape EOT dev:
DRIVER	loads additional device driver (or two (maximum)) DRIVER dd:[/dd:]
BOOT	bootstraps a device BOOT dev:
EXIT	return to XXDP monitor
PRINT	send file to printer PRINT [dev:]filename[/Q]
TYPE	send file to console TYPE [dev:]filename[/Q]

Switches:

/Q	don't rewind tape before access
/N	don't print filename(s) as they are found (or whatever)
/F	in DIR, gives FAST (short) form
/F	in COPY, uses FILE-by-file mode (default for COPY and PIP)
/B	in DIR, gives free BLOCKS information
/L	in DIR, send to LINEprinter
/I	in COPY, use IMAGE mode (default for FILE)

PATCH

Patch is rather like a batch form of UPDAT for modifying files which are too big to fit in UPDAT's buffer. It is used by building a file, or input table, which contains the changes required. Syntax is as the corresponding UPD2 / UPDAT commands except where shown:

Commands:

BOOT	boots a device
CLR	clears the input table CLR
EXIT	returns to the XXDP monitor
GETM	loads a DEC/X11 MAP file GETM [dev:]filename
GETP	loads a saved input table GETP [dev:]filename
KILL	deletes an address entry in the input table KILL addr
MOD	enters an address in the input table
PATCH	creates a patched file from the original file plus the input table PATCH [dev:]outputfile=[dev:]inputfile
SAVP	saves the input table SAVP [dev:]filename
TYPE	displays the input table on console **-** takes no parameters TYPE

SETUP

This is used to pre-build hardware tables for DRS-compatible diagnostics. It works by loading the diagnostic specified, and asking the questions contained therein, just as if running the diagnostic. It then saves the modified version, prompting for deletion of the original if the output filename given is the same as the input filename (a negative answer aborts the save). This is particularly useful for diagnostics to be used in batch files, where operator intervention and DRS dialogue is normally suppressed.

Commands:

LIST — shows all DRS-compatible diagnostics on a device
LIST [dev:][filespec]

SETUP — loads a diagnostic and starts the process (file extension must be BIN or BIC)
SETUP [dev:]outputfile=[dev:]inputfile

EXIT — return to XXDP monitor

XTECO

XTECO is a simple text editor, used primarily to build DO, STARTUP, SYSTEM, or CHAIN files (ie small files of type .CCC). One limitation is that it can only buffer small amounts of text in memory, so it is not always possible to go very far back in a file while editing. Editing is done by issuing commands to modify the contents of the buffer, or to move the buffer pointer. When in edit mode, the prompt changes from an asterisk to two quote characters. Any character, including <tab>, <carriage-return>, <linefeed>, etc, can be placed in a file, but not <escape>, which is used as a command terminator and is echoed as "$". All operations (except J, ZJ) are relative to the current pointer; many can take an optional positive or negative integer, eg to specify how many lines forward or back to move.

Commands:

TEXT

creates a new file
TEXT [dev:]filename

TECO

edits an existing file. Creates a backup (with .BAK extension) but deletes this once the modified file is saved. Only works on random-access devices (ie disks).
TECO [dev:]filename

EDIT

edits an existing file, saving the new version under a different name
EDIT [dev:]newfile=[dev:]oldfile

TYPE

sends a text file to the console

PRINT

sends a text file to the lineprinter

EXIT

returns to the XXDP monitor

Editing:

terminate commands with two <ALTMODE> or <ESCAPE>

L

move by a line; *n*L moves by *n* lines; *n*<0 means move back. The pointer will be placed at the start of the appropriate line.

C

move by a character; *n*C moves by *n* chars; *n*<0 means back.

J

jump to beginning of text in buffer

ZJ

jump to end of text in buffer

S

search for string in buffer, leaving pointer just past *end* of found string, or at end of buffer if no match found. The string may include control characters, such as <carriage-return>. S<string>$$

N

search for string in file, reading in more text (and writing out existing buffer) if required.

T

type line of text; *n*T types *n* lines; *n*<0 means lines before pointer; HT types the entire text. If the pointer is partway into a line, only the part beyond the pointer is typed, unless *n*=0

D delete character; *n*D delete *n* characters; *n*<0 means before pointer

K delete line; *n*K delete *n* lines; *n*<0 means before pointer. If the pointer is partway into a line, only the part beyond the pointer is deleted, unless *n*=0. This command operates exactly like the T command; use the matching T command to see exactly what will be deleted.

I insert string at pointer. I<string>$$

A append text, ie read more into the buffer.

EX finish editing, close files, return to command level (the prompt changes from two quotes to an asterisk).

Commands can be concatenated in any logical way, with commands separated by a single <ALTMODE> or <ESCAPE>.

Batch Files

XXDP Version 2 and later versions of XXDP+ have extensive batch-control facilities. Most monitor commands, utilities commands, and DRS comands and dialogue can be included. In addition, batch files support conditionals, GOTOs, tags (for GOTOs etc), WAIT, etc. lines starting with a semi-colon are regarded as comments.

Special Commands:

IF...THEN an ASCII string goes between the IF and THEN, and subsequent statements up to the END are conditionally executed.

IFERR THEN for use with DRS-type diagnostics only. If the last diagnostic run produced an error code, then the statements following THEN, up to the END, will be executed.

IFLMD *n* THEN to check for a particular medium. This checks the media type byte (at location 41 in XXDP++) and if it matches *n* the following statements, up to the END, will be executed.

END must be used to terminate the list of statements to be conditionally executed.

GOTO used to branch within a batch file, used with a tag.

TAG: an ASCII string followed by a colon, used as a label for GOTO.

R or RUN as the monitor command to run a diagnostic or utility, but the optional / *n* switch permits diagnostics to be run *n* times.

CHAIN runs another batch file. Note: only one level of nesting allowed; ie after running another batch file from within the first, the first will resume, however if a third was run from the second this would not work.

QUIET used as a toggle, controlling typing of the batch file during execution. Being a toggle, even numbers of occurences will turn typing back on.

PRINT used to force typing of a line while QUIET is in effect.
PRINT *text*

SMI set manual intervention; overrides the (normal) suppression of DRS dialogue in batch files.

CMI clear manual intervention; the opposite of SMI.

QUIT terminate batch job

WAIT suspends execution until the operator types CTRL-X

Devices Supported by XXDP

TU60	CT	
RP04/5/6	DB	
TU58	DD	
RK05	DK	
RL01/2	DL	
RK06/7	DM	
RP02/3	DP	
RM02/3	DR	
RS03/4	DS	
DECTape	DT	
RX01	DX	
RX02	DY	
Low Speed Paper Tape	KB	(no specific monitor)
Printer	LP	(no specific monitor)
TM02	MM	
TS04	MS	
TE10	MT	
PDT11	PD	
High Speed PT Reader	PP	(no specific monitor)
Low Speed PT	PT	(no specific monitor)
High Speed PT Reader	PR	(no specific monitor)
Console	TT	(no specific monitor)
MSCP disk	DU	
MSCP tape	MU	

All device drivers in XXDP are small and simple, and may not contain comprehensive error messages. All assume standard CSRs, but can be patched.