Appendix D

Installation guide

D.1 XANADU

The QDP/PLT software is a stand-alone segment of a larger package called *XANADU* (*X-ray ANalysis And Data Utilization*). Subroutines commonly used in XANADU software are placed in a library/archive called XANLIB. QDP/PLT is supplied with source code for the entire XANLIB library. If you are short of disk space, it is possible to delete many of the supplied files. This appendix describes the minimum set of source files you need from these directories. Of course, if you don't plan to modify the code you can delete all source files once the object library has been created.

All XANADU software uses a virtual disk called `XANADU` to hide system-dependent disk names. It is possible to run the QDP/PLT software without implementing the virtual disk, however, some commands such as interactive help will not be able to locate supporting files.

D.2 VMS instructions

All the QDP/PLT software can be used by adding a few lines to your `LOGIN.COM` file. First, you must define the `XANADU` logical name (to create a virtual disk). In brief, if QDP is in the directory called `DRB1:[XRAY.BIN]`, then the following line should be in your `LOGIN.COM`:

```
$ DEFINE/TRANS=(TERM,CONCEAL)   XANADU   DRB1:[XRAY.]
```

(The above line is *very* system dependent.) In this example, `DRB1:` is the physical device name. If you do not translate to the physical device name, but to a logical name (for example, `USER:[XRAY.]`) then you should omit the `TERM` option for the `/TRANS` switch. You will need to define the the following logical names:

```
$ DEFINE   GRAPHICS   XANADU:[PLOT.PGPLOT]
$ DEFINE   GRPSHR     GRAPHICS:GRPSHR.EXE
$ DEFINE   PLT$FONT   GRAPHICS:GRFONT.DAT
$ DEFINE   UFNYSHR    XANADU:[LIB]UFNYSHR.EXE
```

`GRAPHICS` should point to the top level PGPLOT directory. In the above example, this was assumed to be `XANADU:[PLOT.PGPLOT]` and should be changed appropriately if PGPLOT lies in another location. `GRPSHR` and `PLT$FONT` should not need to be changed. You will also need to set up symbols pointing to both QDP and COD with

```
$ QDP      :==$ XANADU:[BIN]QDP
$ COD      :==$ XANADU:[BIN]COD
```

D.3 SUN UNIX instructions

In order to create a *virtual* disk on a Unix system, it is necessary to create a link in the root directory. To do this, you must be root on your machine. If QDP is in the directory `/user/xray/bin`, then go to `/` and type

```
% ln -s /user/xray xanadu
```

In your `.cshrc` file, you will need to define the the following PGPLOT environment variable:

```
setenv PGPLOT_FONT /xanadu/plot/pgplot/grfont.dat
```

Finally you will need to include the XANADU `bin` directory in your current path. If you have not already added other directories to your upath, you should add the following line to your `.cshrc` file:

```
set upath = (/xanadu/bin)
```

D.4 NeXT NextStep instructions

The UNIX interface on the NeXT is highly compatible with the SUN. Therefore you should follow the instructions given in the previous section. Currently, QDP must be run from within a terminal application and not as a stand-alone program with its own icon.

D.5 MS DOS instructions

PGPLOT and QDP/PLT have been tested under MS DOS. Currently the only PGPLOT device handler that supports IBM graphics cards (EGA and so forth) requires the use of Microsoft Fortran 5.0 (or later). If you do not have this version of Microsoft Fortran, you will need to write a new device handler as described in the PGPLOT manual. DOS versions of the system-dependent routines exist, and all the of the QDP/PLT software has been compiled and tested with the Microsoft Fortran compiler. Command line editing is fully implemented under DOS, but you will need to install the DOS `ANSI.SYS` device driver. This is done by adding a line like the following to your `CONFIG.SYS` file.

```
Device   = C:\ANSI.SYS
```

To default with command line editing switched on you should add the following line to your `AUTOEXEC.BAT` file.

```
SET GTBUF_EDIT=ON
```

D.6 Portability

The QDP/PLT software was originally developed on a DEC-VMS system and later ported to a SUN-UNIX and NeXT NextStep systems. QDP/PLT is currently supported on these three systems. Since standard Fortran is extremely portable, it should be a simple matter to port QDP/PLT to other systems. Support for other systems will be minimal (due to lack of time), but occasionally QDP/PLT will be tested on other systems, such as MS DOS, to ensure that major problems are not being introduced.

The author is interested in any attempt to port the QDP/PLT software to other systems and should be consulted before any such attempt is made. System-dependent routines have been isolated into two files called `SYSIO.xxx` and `SYS.xxx`, where `xxx` denotes the operating system. If the software is ported to new systems, it may be necessary to add additional routines to the `SYS.xxx` file. Although `SYS.xxx` should only be used on one system, you should still write the routines in standard Fortran so that others can use your file as a template.

D.7 Relation to PGPLOT

The QDP/PLT software can be considered to be a layer on top of PGPLOT. Thus when installing QDP/PLT on your system, you must first get PGPLOT working. PGPLOT can either be obtained directly from Tim Pearson or from the author when you obtain the QDP/PLT software. If you are already running PGPLOT on your system, then you will need to check that the installed version is sufficiently recent to work with QDP/PLT. If your version of PGPLOT does not contain the routines PGBBUF/PGEBUF, then you will need to update it. You may still wish to update PGPLOT if you discover that your version is older than the version currently being used with QDP/PLT. Of course, if you decide to update your version of PGPLOT, you should be careful to save any locally written or modified versions of PGPLOT device drivers.

If you are not running PGPLOT on your system, then you will need to install it. Since PGPLOT uses its own set of logical names to locate supporting files, it can be located anywhere. However, for consistency with other XANADU systems, you may wish to install PGPLOT in the `XANADU:[PLOT.PGPLOT...]` directory.

D.8 Directory structure

In this documentation, VMS file names have been used. It is a simple matter to map VMS file names to UNIX or DOS names. A file name of `XANADU:[LIB.PLT]PLT.FOR` implies that the file `PLT.FOR` is on the `XANADU` disk, in the `PLT` sub-directory of a main directory called `LIB`. For example, on a Unix system, the above file would be called `/xanadu/lib/plt/plt.for`. For ease of sharing XANADU files with systems that are not case sensitive, only lower case file names are allowed under Unix. On an MS-DOS system, `XANADU` should be the top level directory on your default hard disk; so our sample file would be called `\XANADU\LIB\PLT\PLT.FOR`. An attempt should be made to preserve this type of organization on other systems.

The following directories of the `XANADU` virtual disk contain files needed by the QDP/PLT software:

`XANADU:[BIN]` contains the executable versions of both COD and QDP.

`XANADU:[LIB]` contains the XANLIB object library.

`XANADU:[LIB.COD]` contains a set of sample COD files.

`XANADU:[LIB.PLT]` contains the source code for the PLT subroutine and for other supporting routines.

`XANADU:[LIB.TERMIO]` contains the source code for the low level terminal IO routines.

`XANADU:[LIB.UFNY]` contains the source code for the demo user function. On VMS systems `UFNY.FOR` should NOT be installed into the XANLIB library, but rather the sharable library used instead.

`XANADU:[LIB.XANLIB]` contains the system dependent routines. Currently, there is a `SYS.VMS` should be used on VAX VMS systems, `SYS.SUN` on SUN UNIX systems, and `SYS.NEX` on NeXT NextStep systems.

`XANADU:[LIB.XCOMS]` This directory contains the system-wide indirect command files. For example, the file `HARD.PCO` should create a hardcopy file and then queue the file to the printer.

`XANADU:[LIB.XHELP]` contains the source code for the interactive help.

`XANADU:[LIB.XPARSE]` contains the XSPEC parser routines. Currently the following files are required from this directory: `IXPLWR.FOR`, `XCHKBL.FOR`, `XCHKDL.FOR`, `XCHOSE.FOR`, `XCREAD.FOR`, `XGTARG.FOR`, `XMATCH.FOR`, `XQMTCH.FOR`, `XQUEST.FOR`, `XSQUEZ.FOR`, `XUNIDS.FOR`, and the include file `XPARINC.INC`.

`XANADU:[PLOT.QDP]` contains the source code for the COD and QDP programs, the interactive help files, and the demonstration QDP files.

`XANADU:[PLOT.QDP.MANUAL]` contains the source listings of this manual in LaTeX format. To print the manual, you will need to issue the command `LATEX QDP` twice, to ensure that the table of contents is correct.

`XANADU:[SRC.DOC]` contains the source code for the system level help programs. These are not needed to run the QDP/PLT software, but are useful when creating the help libraries.

D.9 Porting to other systems

When porting QDP/PLT to a new system, you should first get the PGPLOT software working. There are extensive instructions in the PGPLOT manual to help you in this area. Once PGPLOT is working you should next direct your attention to the TERMIO software.

D.9.1 Porting TERMIO software

The TERMIO routines provide the basic terminal I/O for XANADU. Currently the TERMIO software reads and writes to the terminal using one of two completely different methods. One method involves standard Fortran I/O and the second method involves single-character I/O. The single-character method supports command line editing, which should be regarded as a convenience and not as a necessity. Therefore when porting TERMIO to a new operating system, it is a good idea to first get the code working without command line editing. Once the code is working and you have some free time, then you can go back and add the single character I/O routines.

The system dependent routines are in `SYSIO.xxx`. Initially `SYSIO.xxx` should be fairly simple and only allow standard Fortran I/O. To disable single character I/O, you will need to create a new version of `SYSIO.xxx` (where xxx is your system descriptor). This file should contain six routines, and five of these routines can return without doing anything. The routine `FORTYP` should return with the variable `IFTYPE` set equal to zero. When this is done, it becomes impossible to accidently turn on command line editing, which would cause the program to go into an infinite loop if the `RDCHR` and `PUTSTR` routines have not been implemented.

The `[LIB.TERMIO]` directory contains several `makefile`'s, plus an example program `TSTREC.FOR`. You should try out the TSTREC program on your system to ensure that the basic terminal I/O routines are working.

Once you have the standard Fortran version working, and are fishing around for something to do, you should then implement the single character I/O routines. This means implementing the other routines found in `SYSIO.FOR`. Since many UNIX systems do not allow system functions to be called directly from Fortran, many systems use some C code to provide this interface. Under SUN UNIX some routines are implemented using the file `ciosun.c`. The `SYSIO.FOR` routine `TTINIT` should put the terminal into single character I/O mode (called cbreak in UNIX). `TTRSET` should return the terminal to normal mode (called cooked in UNIX). `RDCHR` should read a single character, and `PUTSTR` should write a string to the terminal. Both `RDCHR` and `PUTSTR` should work in passthru mode where control characters (such as escape) are not interpreted by the terminal driver but rather passed to the terminal. `PUTSTR` can buffer the output, until `FLUSH` is called, whereon all data should actually be written to the terminal. Finally, you will need to modify `FORTYP` to return either +1 or -1 depending on whether Fortran on your system outputs a return before or after each Fortran `WRITE` operation. This will also allow command line editing to be switched on.

To test command-line editing, try the `TSTREC.FOR` program and use the interactive `%ED%ON` command to switch it on. Once you are convinced that command-line editing is working you can then use the GTBUF_EDIT logical name/environment variable to automatically switch on editing for all XANADU software.

D.9.2 Creating a new SYS.xxx routine

Before you can run QDP you will need to create one other system dependent file called `XANADU:[LIB.XANLIB]SYS.xxx`. As described below, not all routines in `SYS.xxx` need to be fully implemented in order to use QDP/PLT. In addition, some of the routines are used by other parts of the XANADU software and are not needed by QDP/PLT This section only deals with the routines that are needed by QDP/PLT.

The following routines are implemented in standard Fortran in `SYS.SUN` and therefore you should be able use that version without modification:

```
FRELUN   Free up a logical unit number
GETLUN   Get a free logical unit number
LOCASE   Convert to lower case
UPC      Convert to upper case
```

The following routines can be considered optional in that QDP/PLT will still work if these routines do nothing. Of course, implementing these routines will increase the usefulness of QDP/PLT.

```
CONC     Converts filename to system preferred case
PLTTER   Toggle graphics/alpha mode on terminal
RDFORN   Read a foreign command, {\it i.e.}, the command line
```

`SPAWN`	`Spawn to the operating system`

The last set of routines need to be implemented, although in many cases some loss of functionality is allowed. For example, subroutine `PROMT` must display the prompt. It would be nice if the cursor was left at the end of the prompt line, but if you don't know how to do that on your system, use a standard Fortran write operation.

`DIRPOS`	`Return the number of characters in directory spec`
`OPENWR`	`Wrapup for the Fortran OPEN statement`
`PROMPT`	`Write a prompt on the user's terminal`
`PTEND`	`Add prefix (disk and directory name) to file names`
`TRLOG`	`Translate logical name/environment variable`

D.9.3 Compile and link the QDP program

The final step should be fairly simple. You should compile all the programs in the `[LIB.PLT]` and `[LIB.XHELP]` directories. You should also compile the needed files from `[LIB.XPARSE]` and `[LIB.UFNY]`. The non-system dependent routines should all compile without errors. If you do get an error (for example, due to some non-standard Fortran sneaking in) then please let the author know so that the original can be made made more portable. All files in the `[LIB...]` directory tree should be placed in a library/archive. Finally you should move to the `[PLOT.QDP]` directory and try to link the QDP program. There are several `makefile`'s that can be used as examples.

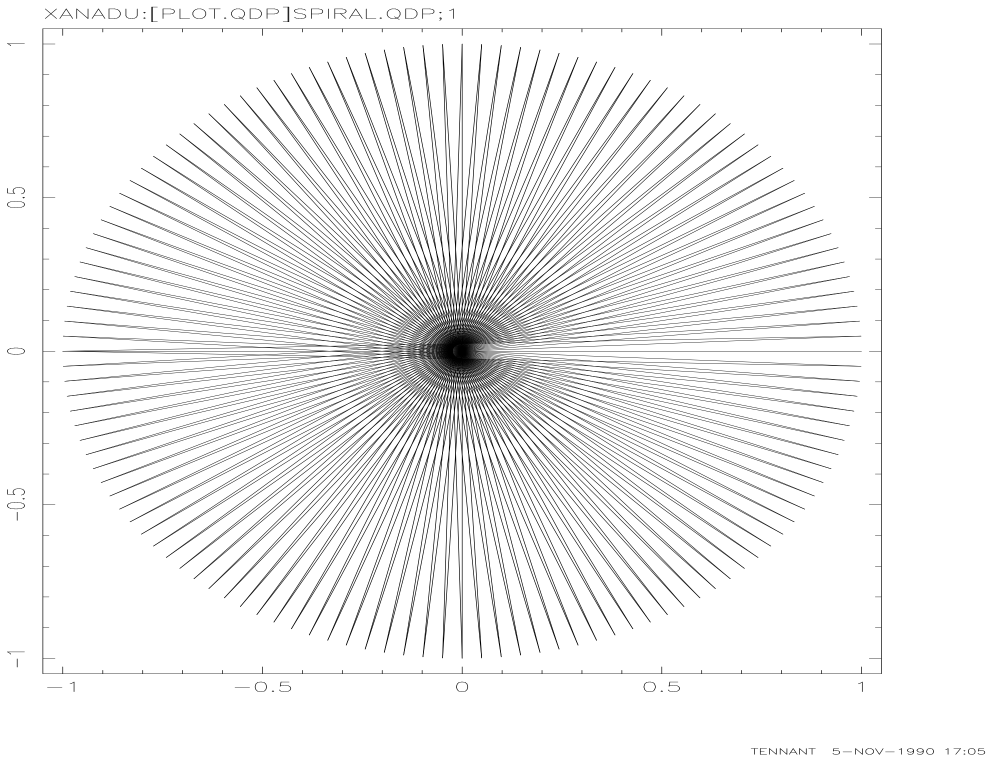
Appendix E

Sample plots and QDP files

This appendix contains several sample figures that were created with the QDP program. In all the figures, the name of the QDP file used to generate the figure is displayed in the top left corner. For the first figure this name is `XANADU:[PLOT.QDP]SPIRAL.QDP`. It is possible to reproduce this figure by using:

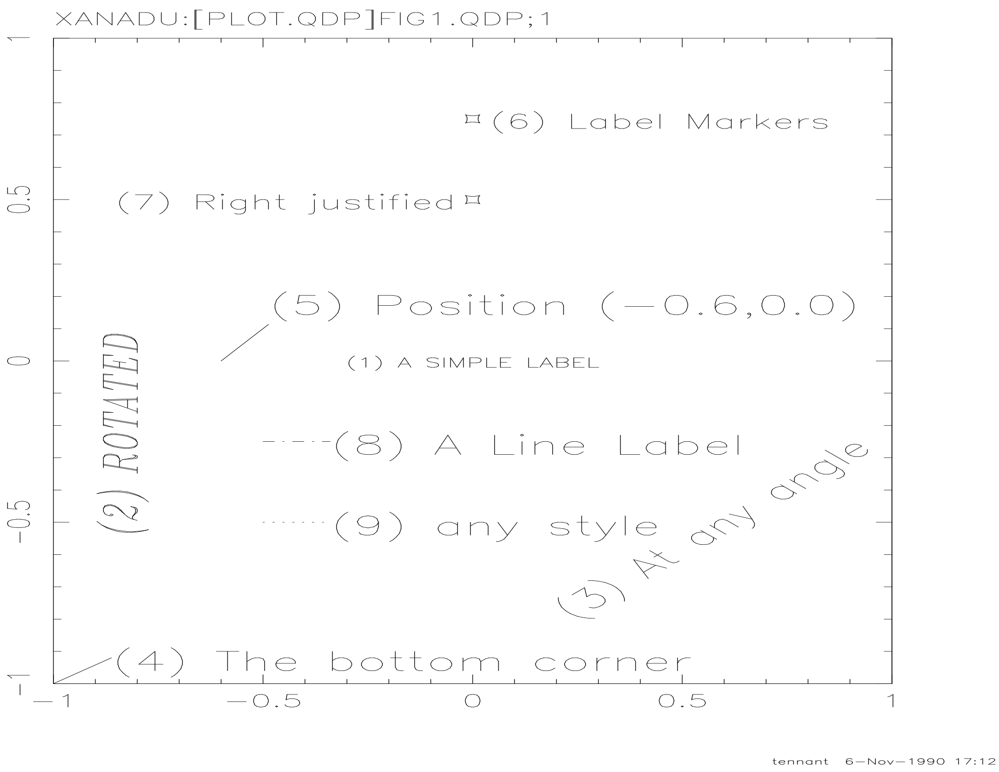
```
$ QDP XANADU:[PLOT.QDP]SPIRAL
```

For reader convenience, the top section of each QDP file is listed. In cases where the QDP also uses a PLT command file, the top section of the `.PCO` file is also listed. A careful comparison of each figure and the PLT commands used to generate it, will illustrate the actions of many PLT commands.



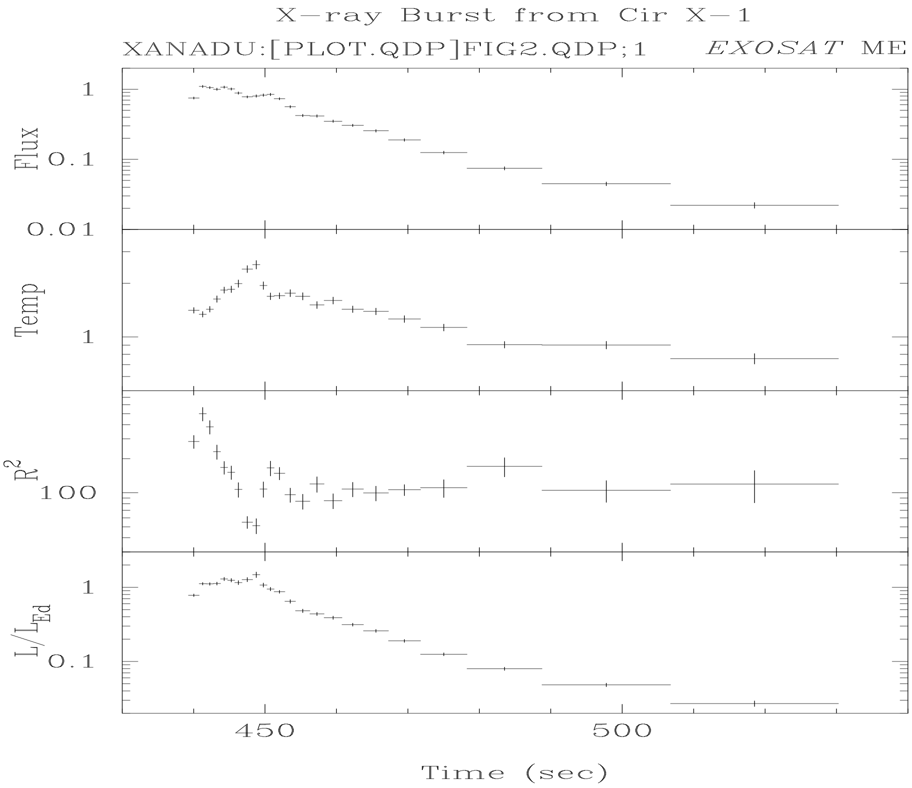
The `SPIRAL.QDP` file contains:

```
 1.0000  0.0000
-1.0000  0.0000
 0.9988  0.0491
-0.9988 -0.0491
 0.9952  0.0980
-0.9952 -0.0980
 0.9892  0.1467
-0.9892 -0.1467
 0.9808  0.1951
-0.9808 -0.1951
 0.9700  0.2430
-0.9700 -0.2430
 0.9569  0.2903
-0.9569 -0.2903
 0.9415  0.3369
-0.9415 -0.3369
 0.9239  0.3827
-0.9239 -0.3827
 0.9040  0.4276
-0.9040 -0.4276
    "       "
```



The **FIG1.QDP** file contains:

```
CS  1.3
GAP 0.0
LAB 1 "(1) A SIMPLE LABEL"
LAB 2 P -0.80 -0.50 J L RO 90 CS 2 "\FI(2) ROTATED"
LAB 3 P .6 -.5  CS 2 CO 3 RO 45 "(3) At any angle"
LAB 4 P -1  -1 LI 30 CS 2 CO 4 "(4) The bottom corner"
LAB 5 P -.6 0 LI 45 CS 2 CO 5 "(5) Position (-0.6,0.0)"
LAB 6 P 0. .75 MA 10 CS 1.5 CO 6 "(6) Label Markers"
LAB 7 P 0. .50 MA 10 CS 1.5 CO 6 J R "(7) Right justified"
LAB 8 P -.50 -.25 LI 0 LS 3 CO 8 CS 2 "(8) A Line Label"
LAB 9 P -.50 -.5  LI 0 LS 4 CO 8 CS 2 "(9) any style"
!
-1      -1
 1      -1
 1       1
```

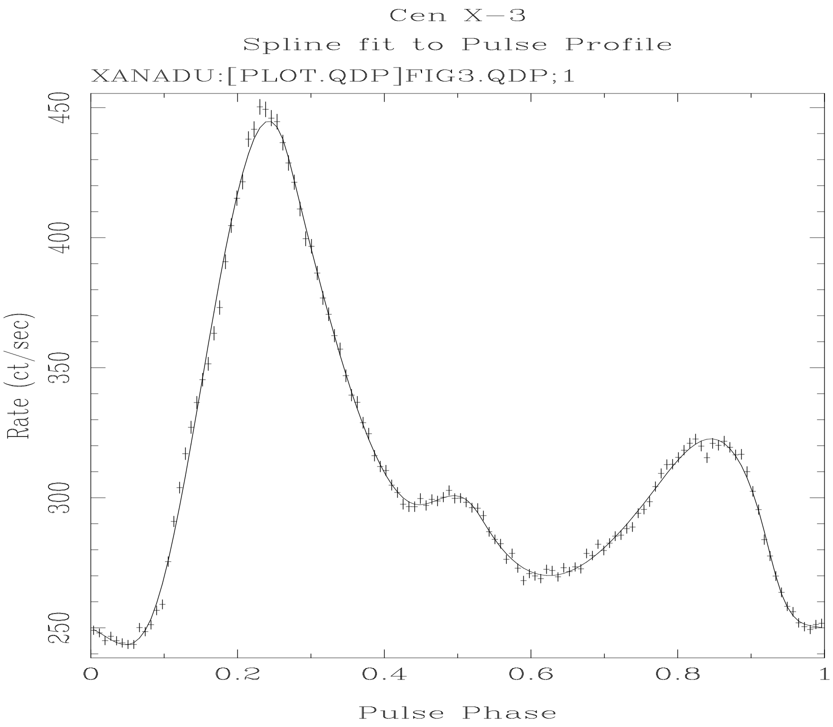


The FIG2.QDP file contains:

```
READ SERR 1 2 3 4 5
@"XANADU,PLOT/QDP,FIG2"      ! See PLT Command File "FIG2.PCO"
!
440.02 0.75  0.749 2.8E-2  1.41 5.2E-02  284.3 38.0  0.1445/.185 6.0E-03/.185  62.7
      "           "            "              "                 "                "
```

The above QDP file calls the PLT command file FIG2.PCO:

```
COL  OFF 1,6
CSIZ 1.3
FONT RO
LAB  1 CS 1.3 CEN BOT JUS RIG VP .9 .91 "\FIEXOSAT \FRME"
LAB  RO
LAB  T  X-ray Burst from Cir X-1
LAB  G1 Time (sec)
R    Y1 430 540
LAB  G2 Flux
R    Y2 .01 2
LAB  G3 Temp
R    Y3 .5 4
LAB  G4 R\U2
R    Y4 30 800
LAB  G5 L/L\DEd
R    Y5 .02 3
LAB  G6 Chi
LAB  POS Y 2.5
LOG  Y
P    V
TIM  OFF
VIEW .15,,.90
```

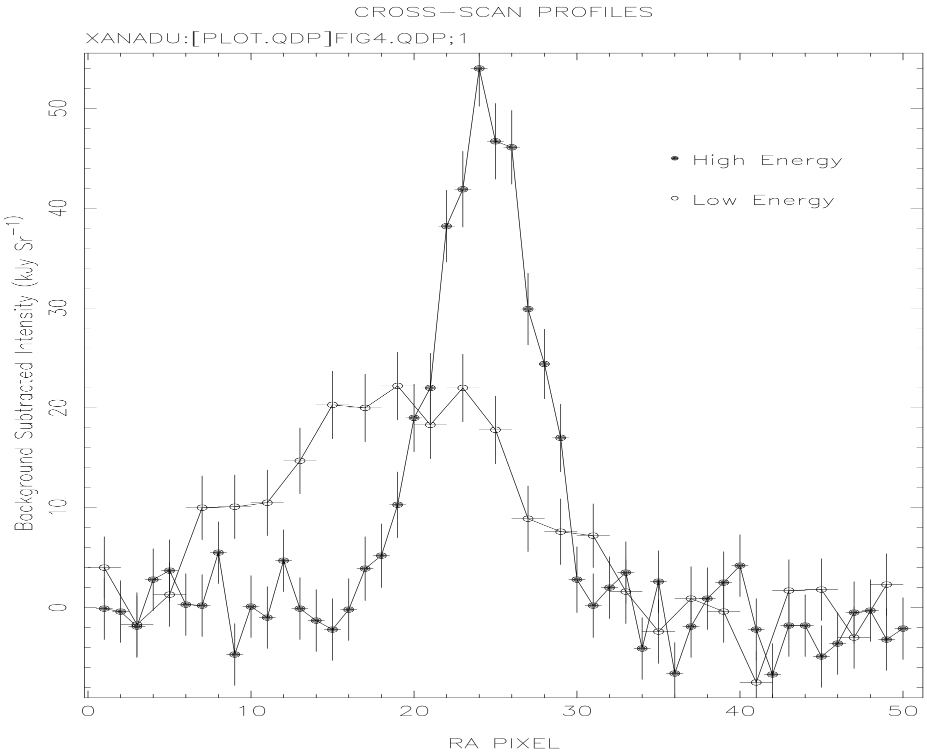


The FIG3.QDP file contains:

```
! FOLD.  Scatter errors.
! Start time(   82135.4), End time(   94966.3)
! Period( 4.827641200D+00), Phase 0( 4.000D-01), DPDOT( 0.000D+00)
READ SERR 1
XAXIS LIN  3.9062500E-03  7.8125000E-03
@"XANADU,PLOT/QDP,FIG3"      ! See PLT Command File "FIG3.PCO"
!
   249.0985       1.712956
   248.0650       1.642315
   245.0948       1.686881
   246.7742       1.685483
   245.0568       1.691913
      "               "
```

The above QDP file calls the PLT command file FIG3.PCO:

```
CSIZ 1.2
FONT RO
LAB  OT  Cen X-3
LAB  T   Spline fit to Pulse Profile
LAB  X   Pulse Phase
LAB  Y   Rate (ct/sec)
LAB  PAR OFF
MOD  @"XANADU,PLOT/QDP,FIG3"
R    X   0 1
TIM  OFF
VIEW .15 .15
```

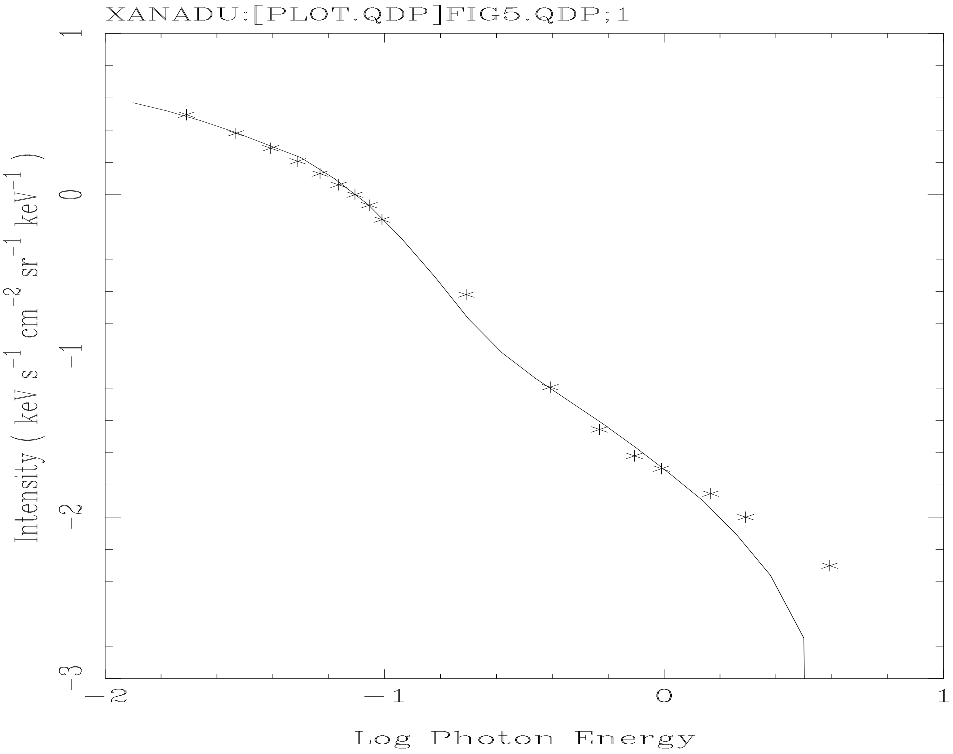


The FIG4.QDP file contains:

```
READ SERR 1 2
@"XANADU,PLOT/QDP,FIG4"      ! See PLT Command File "FIG4.PCO"
!
 1.00  0.50 -1.E-01  3.1
 2.00  0.50 -0.4  3.1
 3.00  0.50 -1.9  3.1
 4.00  0.50  2.8  3.1
 5.00  0.50  3.7  3.1
 6.00  0.50  0.3  3.1
 7.00  0.50  0.2  3.1
 8.00  0.50  5.5  3.1
 9.00  0.50 -4.7  3.1
     "          "
```

The above QDP file calls the PLT command file FIG4.PCO:

```
SKIP SING
LINE ON   1
MARK   17 ON   1
LINE ON   2
MARK   21 ON   2
MARK SIZE  1.5
LAB  1 MAR 17 JUS Lef
LAB  1 POS 36 45 "High Energy"
LAB  2 MAR 21 JUS Lef
LAB  2 POS 36 41 "Low Energy"
LAB  T  CROSS-SCAN PROFILES
LAB  X  RA PIXEL
LAB  Y  Background Subtracted Intensity (kJy Sr\u-1)
TIME OFF
```

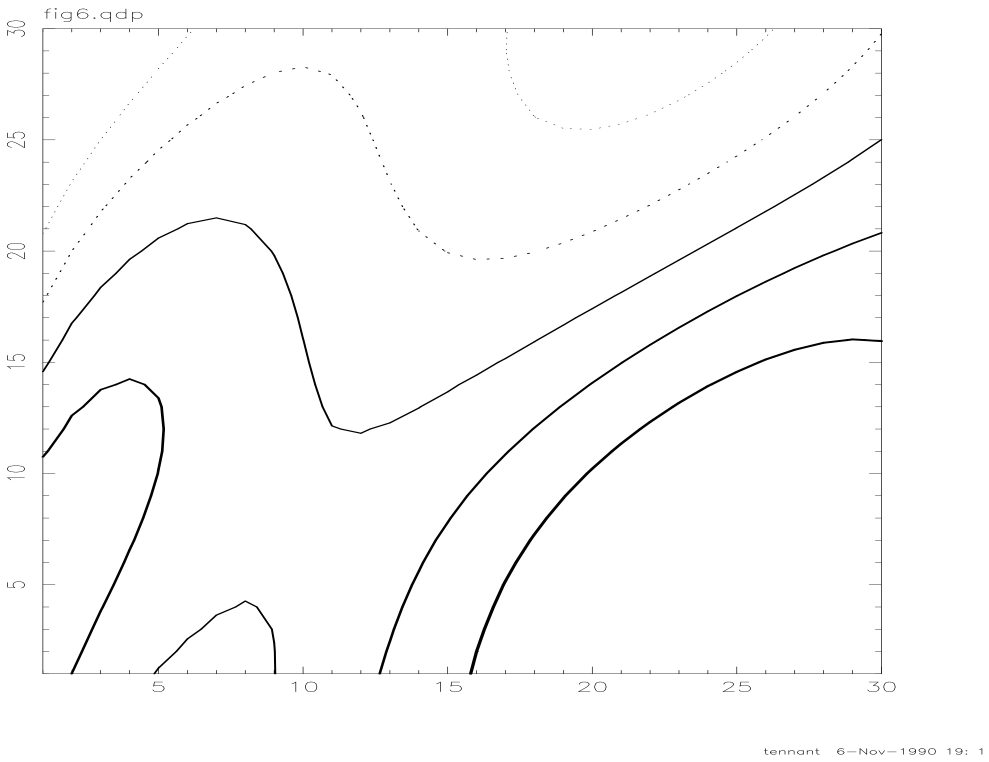


The `FIG5.QDP` file contains:

```
skip on
@"xanadu,plot/qdp,fig5"         ! See PLT Command File "FIG5.PCO"
  -7.300000     -0.6800003
  -7.180000     -0.4200001
  -7.060000     -0.1800003
  -6.940000      6.0000420E-02
  -6.820000      0.3000002
  -6.700000      0.5400000
  -6.580000      0.7700005
  -6.460000       1.010000
  -6.340000       1.240000
  -6.220000       1.450000
  -6.100000       1.750000
       "               "
```

The above QDP file calls the PLT command file `FIG5.PCO`:

```
CSIZ 1.3
FONT ROMAN
LAB  X Log Photon Energy
LAB  Y Intensity ( keV s\u-1\d cm\u-2\d sr\u-1\d keV\u-1\d )
LINE ON 1
MARK 3 ON 2
MARK SIZE 2.0
R   -2 1 -3 1
TIME OFF
```



The FIG6.QDP file contains:

```
COL OFF 1..999
CON LEV -1. -.5 0. .5 1.  LS 4 4 1 1 1   LW 1 2 3 4 5
XAX LIN   1 1
YAX LIN   1 1
R   1 30 1 30
!
 .748  .795  .826  .842  .841  .823  .788  .734  .664  .577  .474  .358  .228-
   8.700E-02 -6.300E-02 -.220 -.383 -.548 -.713 -.876 -1.035 -1.187 -1.331-
  -1.464 -1.585 -1.692 -1.783 -1.859 -1.917 -1.958
 .499  .574  .639  .691  .731  .755  .764  .756  .731  .689  .631  .555  .464-
   .358  .239  .107 -3.400E-02 -.184 -.340 -.500 -.662 -.823 -.981 -1.135-
  -1.281 -1.418 -1.545 -1.658 -1.758 -1.842
 .281  .364  .442  .513  .573  .622  .658  .681  .688  .679  .654  .613  .556-
   .483  .395  .292  .177  5.000E-02 -8.600E-02 -.230 -.381 -.535 -.691 -.846-
  -.999 -1.147 -1.289 -1.422 -1.544 -1.654
   "
```

The QDP/PLT User's Guide

Allyn F. Tennant

X-Ray Astronomy Branch
NASA Marshall Space Flight Center

January 25, 1991

Please address questions, suggestions for improvements, and/or reports of software bugs, to the author:

Allyn Tennant, ES-65
NASA MSFC
Huntsville, AL 35812
USA

Telephone: 205 544-3424
FAX: 205 544-7754

SPAN: SSL::TENNANT or 7207::TENNANT
Internet: tennant%ssl.msfc.nasa.gov

Contents