Contents

If you need Help i

1 General 1

2 Application Software 2

2.1 A new version of PAW 2
PAW++ 2
Ntuple variables and selection functions 5
Graphics developments 7
New and modified PAW commands 8
FIAF, the Parallel Interactive Analysis Facility 12

2.2 KUIP developments 13

2.3 HBOOK changes and new routines 17
New HBOOK file format 17
New Ntuples in HBOOK 17
New or modified HBOOK routines 18

2.4 New Version 1.41 of CMZ 22
Editorial

Your subscription to the CNL is controlled by sending an email to cnl@cernvm.cern.ch as follows:

- To subscribe for an address outside CERN, specify:
  
  **Subject: PAPER CNL OUTSIDE CERN**

  and the body of the mail containing your address and nothing else.

- To desubscribe inside CERN, specify:

  **Subject: NO PAPER CNL your_full_name / division**

  The body of the mail will be ignored.

- To get informed when the PostScript version is available, specify:

  **Subject: ANNOUNCE POSTSCRIPT CNL**

Contributions to the CNL are accepted as plain text, although marked-up text in \LaTeX{} or SGML is preferred. They should be sent as email to cnl@cernvm.cern.ch.

We want to express our special thanks to Michael Metcalf and Harry Renshall for proofreading the text and to Alexander Samarin for helping us with some \LaTeX{} aspects of the production of the present CNL.

The dates of the deadlines for contributions to the CNL in 1992 and 1993:

<table>
<thead>
<tr>
<th>Issue number</th>
<th>Final date</th>
</tr>
</thead>
<tbody>
<tr>
<td>209</td>
<td>1 November 1992</td>
</tr>
</tbody>
</table>

This document was produced with \LaTeX{} and the cerncn1 style. PostScript files, containing the complete printable version of this CNL or parts of it can be obtained by anonymous ftp as follows (commands to be typed by the user are underlined):

```plaintext
tftp cernvm.cern.ch
Trying 128.141.201.136...
Connected to crnvma.cern.ch.
220-FTPD at cernvm.CERN.CH, 08:39:44 SET THURSDAY 04/23/92
220 Connection will close if idle for more than 5 minutes.
Name: anonymous
230 ANONYMOUS logged in with no special access privileges
ftp> cd cnl.200
250-Working directory is CNL 200 (ReadOnly)
ftp> get cnl1208.ps
ftp> quit
```

The following files related to the present CNL are available on that minidisk:

- `cnl1208.ps` Complete CNL
- `cnl1208e.ps` Editorial page of the CNL
- `cnl1208t.ps` Chapter “Text Processing and Documentation”
- `cnl1208f.ps` “Fortran90 Tutorial”
- `cnl1208a.ps` “viemacs Tutorial”
- `cnl1208h.ps` Helppage of the CNL
- `cnl1208g.ps` Chapter “General”
- `cnl1208c.ps` Chapter “Communications”
- `cnl1208l.ps` “\LaTeX{} Tutorial”
- `cnl1208p.ps` Chapter “Application Software”

At CERN one has access to the terminal version of the CNL via the command:

```
IFIND CNL 208
```

On CERNVM IFIND will give access to the printable version as well.

**Responsible editor: Bernd Pollermann**

**Technical realization: Michel Goossens**
### IF YOU NEED HELP

<table>
<thead>
<tr>
<th>Service</th>
<th>Name</th>
<th>Address</th>
<th>Telephone</th>
<th>E-mail Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>USER CONSULTANCY OFFICE (UCO)</td>
<td></td>
<td>513/R-052</td>
<td>4952</td>
<td>uco@cernvm</td>
</tr>
<tr>
<td>CENTRAL COMPUTER OPERATIONS (24 HOURS)</td>
<td></td>
<td>513/R-066</td>
<td>5011</td>
<td>operator@cernvm</td>
</tr>
<tr>
<td>All aspects</td>
<td>J.-C. Juvet</td>
<td>513/R-037</td>
<td>4935,13+5544</td>
<td>juvet@cernvm</td>
</tr>
<tr>
<td>IBM-SIEMENS</td>
<td>David Underhill</td>
<td>513/R-035</td>
<td>4920</td>
<td>djuct@cernvm</td>
</tr>
<tr>
<td>Central VAXes</td>
<td>Tim Whibley</td>
<td>513/R-033</td>
<td>4849</td>
<td>tim@xcern</td>
</tr>
<tr>
<td>Networks &amp; Remote Printers</td>
<td>Remy Bouvry</td>
<td>513/R-004</td>
<td>4929,13+5548</td>
<td>bov@cernvm</td>
</tr>
<tr>
<td></td>
<td>Robert Ceccato</td>
<td>513/R-004</td>
<td>4929</td>
<td>cre@cernvm</td>
</tr>
<tr>
<td>CRAY</td>
<td>V.Dore</td>
<td>513/R-030</td>
<td>4976,13+3830</td>
<td>dore@cernvm</td>
</tr>
<tr>
<td>CERN PROGRAM LIBRARY OFFICE</td>
<td></td>
<td>513/1-005</td>
<td>4912</td>
<td>cernlib@cernvm</td>
</tr>
<tr>
<td>All aspects</td>
<td>Miguel Marquina</td>
<td>513/1-003</td>
<td>5010</td>
<td>cernlib@cernvm</td>
</tr>
<tr>
<td>Apollo, HP/UX Support</td>
<td>Gunter Folger</td>
<td>513/1-003</td>
<td>5010</td>
<td>cernlib@cernvm</td>
</tr>
<tr>
<td>DECstation, Sun Support</td>
<td>Ian McLaren</td>
<td>513/1-003</td>
<td>5010</td>
<td>cernlib@cernvm</td>
</tr>
<tr>
<td>Distribution of CERNlib material</td>
<td>Lysiane Besson</td>
<td>513/1-014</td>
<td>4951</td>
<td>cernlib@cernvm</td>
</tr>
</tbody>
</table>

### OTHER SERVICES

<table>
<thead>
<tr>
<th>Service</th>
<th>Name</th>
<th>Address</th>
<th>Telephone</th>
<th>E-mail Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Support (for UCO see above)</td>
<td>Harry Renshall</td>
<td>513/1-009</td>
<td>2377</td>
<td>hrrcr@cernvm</td>
</tr>
<tr>
<td>Operating Systems</td>
<td>Sverre Jarp</td>
<td>31/1-025</td>
<td>4944</td>
<td>sverre@cernvm</td>
</tr>
<tr>
<td>Tape operations</td>
<td>R.P.Minchin</td>
<td>513/R-009</td>
<td>13+5559</td>
<td>rpmct@cernvm</td>
</tr>
<tr>
<td>Tape purchase</td>
<td>Mario Vergari</td>
<td>513/R-009</td>
<td>13+5602</td>
<td>vnv@cernvm</td>
</tr>
<tr>
<td>Documentation office</td>
<td>Jean Zueras</td>
<td>513/1-022</td>
<td>2371</td>
<td>jrzcn@cernvm</td>
</tr>
<tr>
<td>Computer Science Library, mornings only</td>
<td>Jutta Megies</td>
<td>513/1-024</td>
<td>2379</td>
<td>megm@cernvm</td>
</tr>
<tr>
<td>Oracle</td>
<td>William Moorhead</td>
<td>31/3-008</td>
<td>2995,13+5580</td>
<td>wgsm@cernvm</td>
</tr>
<tr>
<td>Experimental Tape Allocation</td>
<td>Hansjorg Klein</td>
<td>13/3-024</td>
<td>2124,2060</td>
<td>hkl@cernvm</td>
</tr>
<tr>
<td>Computer Security</td>
<td>David Lord</td>
<td>13/3-024</td>
<td>3967</td>
<td>pfkdn@cernvm</td>
</tr>
<tr>
<td>Workstations General Information</td>
<td>Alan Silverman</td>
<td>31/1-030</td>
<td>4955</td>
<td>alain@cernapo</td>
</tr>
<tr>
<td>Apollo Workstations Hardware Support</td>
<td>K.S.Olofsson</td>
<td>513/R-009</td>
<td>7114,13+5564</td>
<td>soren@cernapo</td>
</tr>
<tr>
<td>Apollo Software Support</td>
<td></td>
<td>31/1-024</td>
<td>4911,13+5563</td>
<td>apollosp@cernapo</td>
</tr>
<tr>
<td>HP/UX Software Support</td>
<td></td>
<td>31/1-024</td>
<td>4911,13+5563</td>
<td>hpssp@cernapo</td>
</tr>
<tr>
<td>DECstation,VAX/ULTRIX support</td>
<td>Alan Lovell</td>
<td>31/1-024</td>
<td>4911,13+5605</td>
<td>ultrixsp@dcxcern</td>
</tr>
<tr>
<td>Sun Software support</td>
<td>Ignacio Reguero</td>
<td>31/1-027</td>
<td>4409</td>
<td>susap@cernvm</td>
</tr>
<tr>
<td>Central Vax Service Manager</td>
<td>Julian Bunn</td>
<td>31/1-025</td>
<td>5029,13+5551</td>
<td>julian@cernvm</td>
</tr>
</tbody>
</table>

Central Computer registration and accounting: See Group or Divisional Administrator or UCO

### ONLINE COMPUTING:
See "ONLINE", the Newsletter of Data acquisition and Computing for Experiments, available from Anne Perrelle

### COMPUTING FOR ENGINEERING:
See the CERN Computing Support for Engineering Newsletter.

### COMPUTER TIME ALLOCATION GROUP (COCOTIME):
Secretary: A.E.Ball/ECP 3849 aeb@cernvm

### DIVISIONAL REPRESENTATIVES FOR DATA COMMUNICATIONS AND TERMINALS

#### DIV. Representative

<table>
<thead>
<tr>
<th>Div.</th>
<th>AG</th>
<th>DG</th>
<th>PE</th>
<th>ST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V.Frigo</td>
<td>V.Frigo</td>
<td>D.Duret</td>
<td>G.Martin</td>
</tr>
</tbody>
</table>

#### Div. Representative

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AS</td>
<td>AS</td>
<td>M.Draper</td>
<td>AT</td>
<td>J.Schinzel</td>
<td>CN</td>
<td>A.Koppanyi</td>
</tr>
<tr>
<td></td>
<td>DG</td>
<td>DG</td>
<td>ECP</td>
<td>F.I.</td>
<td>S.Lauper</td>
<td>MT</td>
<td>S.Foffano</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>PE</td>
<td>PPE</td>
<td>H.Klein</td>
<td>PS</td>
<td>J.Riche</td>
<td>SL</td>
</tr>
<tr>
<td></td>
<td>ST</td>
<td>ST</td>
<td>TH</td>
<td>T.Sjostrand</td>
<td>TIS</td>
<td>G.R.Stevenson</td>
<td></td>
</tr>
</tbody>
</table>

### DIV. Representative

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AS</td>
<td>M.Draper</td>
<td>AT</td>
</tr>
<tr>
<td></td>
<td>ECP</td>
<td>E.Pagliola</td>
<td>FI</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>D.Duret</td>
<td>G.Shering</td>
</tr>
<tr>
<td></td>
<td>TH</td>
<td>T.Sjostrand</td>
<td>TIS</td>
</tr>
</tbody>
</table>

### Div. Representative

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AG</td>
<td>V.Frigo</td>
</tr>
<tr>
<td></td>
<td>ECP</td>
<td>E.Pagliola</td>
</tr>
<tr>
<td></td>
<td>PE</td>
<td>D.Duret</td>
</tr>
<tr>
<td></td>
<td>TH</td>
<td>T.Sjostrand</td>
</tr>
</tbody>
</table>


COMMUNICATIONS AND NETWORKS

CN Division is responsible for the general purpose internal Ethernet networks and their associated fibre backbone.

Consultation on network-related software purchases and Ethernet interfaces:
   J.Gamble, gamble@vxcern.cern.ch or
   J.M.Gerard, jmg@dxcern.cern.ch

General operational problems:
   netops@vxcern.cern.ch (if urgent: 4927, or 5011 night/weekend)

General Ethernet problems:
   sohet@cernvm.cern.ch or 2299 (if urgent: 2386)

General user consultancy:
   uco@cernvm.cern.ch (if urgent: 4952)

TCP/IP registration requests:
   tcpip@vxcern.cern.ch or tcpip@cernvm.cern.ch

LAT terminal server registration requests:
   lat-support@dxmint.cern.ch

DECnet Registration Requests and Queries:
   decnet-support@dxmint.cern.ch (DXMINT::DECNET-SUPPORT)

Other registration requests:
   netops@vxcern.cern.ch

Electronic mail queries:
   mail-support@dxmint.cern.ch

General PC networking advice:
   hagen@dxcons.cern.ch

Novell Netware coordination:
   David Foster CN/CE, david@cernvax.cern.ch

Advice on Macintosh networking and X terminals:
   jmg@dxcern.cern.ch

General consultations on internal networking:
   Software: gamble@vxcern.cern.ch
   Network infrastructure: jmj@dxcons.cern.ch or sohet@cernvm.cern.ch
   General: B.Carpenter, brian@dxcern.cern.ch
1. General
2. Application Software

2.1 A new version of PAW

*CNAS Graphics Section*

A new version 2.00 of PAW has been introduced in the production areas P00 on all supported computer platforms on January 19th 1993. This version was available in the development area NEW since the beginning of December 1992. The many new features and enhancements offered by the new version are briefly summarised in the following sections.

The full description of the new options can be obtained as usual by the HELP command inside PAW. The PAW manual will also be updated to fully reflect these changes in due time.

Suggestions or comments about the remaining section in this chapter should be sent to the following people:

<table>
<thead>
<tr>
<th>Name</th>
<th>Email address</th>
<th>Subject area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfred Nathaniel</td>
<td><a href="mailto:NATHANIE@CERNVM.CERN.CH">NATHANIE@CERNVM.CERN.CH</a></td>
<td>KUP</td>
</tr>
<tr>
<td>Nicole Cremel</td>
<td><a href="mailto:CREMEL@CERNVM.CERN.CH">CREMEL@CERNVM.CERN.CH</a></td>
<td>KUIP/MOTIF</td>
</tr>
<tr>
<td>Olivier Couet</td>
<td><a href="mailto:COUET@CERNVM.CERN.CH">COUET@CERNVM.CERN.CH</a></td>
<td>graphics and PAW/HLOT/HIGZ</td>
</tr>
<tr>
<td>Fons Rademakers</td>
<td><a href="mailto:RDMO@PSALO.CERN.CH">RDMO@PSALO.CERN.CH</a></td>
<td>HBOOK’s new Ntuples</td>
</tr>
<tr>
<td>Adam Cash</td>
<td><a href="mailto:CASH@CERNVM.CERN.CH">CASH@CERNVM.CERN.CH</a></td>
<td>Ntuples selection mechanisms</td>
</tr>
<tr>
<td>Carlo Vandoni</td>
<td><a href="mailto:VANDONI@CERNVM.CERN.CH">VANDONI@CERNVM.CERN.CH</a></td>
<td>SIGMA</td>
</tr>
<tr>
<td>John Allison</td>
<td><a href="mailto:ALLISON@CERNVM.CERN.CH">ALLISON@CERNVM.CERN.CH</a></td>
<td>commands SMOOTH and NT/DRAW</td>
</tr>
<tr>
<td>Otto Schalle</td>
<td><a href="mailto:SCHALLE@CERNVM.CERN.CH">SCHALLE@CERNVM.CERN.CH</a></td>
<td>ZEBRA browser of PAW++</td>
</tr>
<tr>
<td>Miguel Marquina</td>
<td><a href="mailto:MARQUINA@CERNVM.CERN.CH">MARQUINA@CERNVM.CERN.CH</a></td>
<td>installation procedures</td>
</tr>
<tr>
<td>René Brun</td>
<td><a href="mailto:BRUN@CERNVM.CERN.CH">BRUN@CERNVM.CERN.CH</a></td>
<td>any item not mentioned above</td>
</tr>
</tbody>
</table>

**PAW++**

PAW++ is an extended version of PAW with a new built-in MOTIF user interface. In addition to the conventional command line and macros types of interface, the following dialogue modes are also available:

- **Pull Down menus** They are useful to understand the structure of an interactive program.
- **Object Browser** This is in many ways similar to the well know browsers in the PC/MAC utilities or the visual tools on some workstations.
- **Direct graphics** One can click in the graphics area and identify automatically which object has been selected. A pop-up menu appears with a list of possible actions on this object. For example, by clicking on a histogram, one can make directly a gaussian fit, etc. Pop-up menus are available in clicking on the graphics window to automatically produce PostScript, Encapsulated PostScript, and \LaTeX\ files or print the picture on your local printer.
- **Histogram style panel widget** Buttons are available to change histogram attributes, colours, line styles, fonts, axes representation. 2-D histograms can be rotated interactively.
- **Ntuple viewer** Just click on the Ntuple column name to histogram the column, etc.

The new system is largely self-explanatory. Only a subset of PAW has been converted to this new user interface, but work is currently in progress to offer many new facilities in the coming releases.
Figure 2.1: PAW++ display showing various panels (see also the next figure).

- The upper left corner is the PAW++ Executive window, with its input pad at the bottom and the transcript pad at the top.
- The PAW++ Browser, where the various entities (pictures, 1-dim and 2-dim histograms and Ntuples) are all defined with their own symbol, is shown bottom left. A "pop-up" menu has been activated for the chosen 1-dimension histogram and the Plot and Smooth buttons have been activated.
- The graphics window is seen top right. Two 2-dim views (a lego-plot and a grey scale plot) and a 1-dim view of the data points are shown. For the 1-dim case, the results of a "smoothing" type of fit to the data points is also drawn. Information about the data and the fit can be found in the inserted window.
- The "Histogram Style Panel" at the lower right allows graphihcs attributes of the histogram to be controlled.
Figure 2.2: PAW++ display showing various panes (case of an Ntuple)

- The upper left corner shows the Ntuple Viewer. The left window shows the name of the various variables, characterizing the selected Ntuple. Other windows and press-buttons specify which combinations of the various variables and which events have to be treated (plotted, scanned...).
- The lower left contains the PAW++ Browser, with this time an Ntuple selected A “pop-up” menu has been activated and the Open Ntuple Viewer button activated (see above).
- The graphics window is seen top right and shows a 3-dimensional view of the combination PX1, PY1, PZ1 of the variables, whose cuts are specified with the cut editor (see below). A three dimensional view as well as two one-dimensional projections of the selected parameters are displayed.
- The “Cut Editor” panel, shown at the lower right, allows various combinations as cuts to be specified and applied.
Ntuple variables and selection functions

With the introduction of the new kind of Ntuples (see the section on HBOOK starting on page 17), the operands and expressions used as selection criteria in Ntuple operations can now be of the following types:

**Numerical**
Real and integer numbers, e.g. 23.877, 23, -2.3, 0

**Character string**
This is a string of characters (plus the wild characters 'r' and 'e' see "Other features" below) written in the following form: 'AbCd' (Upper and lower case, within quotes)

**Bit strings (unsigned integers)**
These are numbers written in Hexadecimal, Octal or Binary form enclosed in single quotes.
- Hexadecimal numbers are written in the form Z'xxxx', where x is a hexadecimal digit in the range 0 - F.
- Octal numbers are written in the form 0'0000', where o is an octal digit in the range 0 - 7.
- Binary numbers are written in the form B'0000', where b is either 0 or 1.

**Logical**
Logicals are written in the form TRUE (true) and FALSE (false), i.e. Upper or lower case without quotes. This implies that the names TRUE and FALSE are not permitted as Ntuple column identifiers.

**Cuts**
A cut is identified by a '$$' followed by an integer (written in the form $nnn$, where 0<n<100). A legal Cut expression may include any kind of expression, other cuts, masks or COMIS functions, provided the result is boolean. As the result of a boolean cut is mapped to the real numbers 0. and 1. so that it is possible to operate arithmetically on results of boolean cuts. For reasons of backward compatibility, the $ sign can be omitted on the CUT command (but it must appear on the selection criteria used by the NTUPLE/SCAN (or PLOT) command. Example of creating a cut

\[
\text{NTUPLE/CUT $1$ (SQRT(A+B).LE.3)}
\]
\[
\text{NTUPLE/CUT $2$ (A=3) .AND. (B=5)}
\]
\[
\text{NTUPLE/CUT $3$ NOT($1$) .AND. $2$}
\]

Assuming there exists a mask vector MSK and a COMIS function COMIS1.FTN
\[
\text{NTUPLE/CUT $4$ (COMIS1>30.2) .AND. MSK(3)}
\]

**Example of using a cut**

\[
\text{NTUPLE/PLOT 30.A $1.OR.(B=12)}
\]
\[
\text{NTUPLE/SCAN 30 $3$}
\]
\[
\text{NTUPLE/PLOT 30.A $4*5.3$}
\]
\[
\text{! (5.3 is weight)}
\]

**Masks**
The Mask facility allows up to 32 selection criteria associated with a Ntuple. One bit of a mask (which has a logical type) is identified by the mask name followed by an index of the bit within brackets. The index ranges from 1 to 32. Assuming there exists a mask vector MSK, then one can can write e.g.
\[
\text{NTUPLE/PLOT 30.X MSK(4) (bit 4)}
\]

The Mask vector is of type "bit string" and is identified by the name of the mask with no bit index and no brackets, e.g.
\[
\text{NTUPLE/PLOT 30.X MSK.EQ.B'101'}
\]

This is identical to the following, where the mask bits are referenced explicitly:
\[
\text{NTUPLE/PLOT 30.X .NOT. (MSK(2)).AND. .NOT. (MSK(1)).AND. (MSK(3))}
\]

The bits are treated from right to left, i.e. position 1 is the least significant bit.

**Operators**
The table below shows the order in which operations are carried out in an expression. Operations of identical precedence are evaluated left to right. The order can of course be changed by using parentheses.

<table>
<thead>
<tr>
<th>Operators</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>×</strong></td>
<td>raising to a power</td>
</tr>
<tr>
<td>*</td>
<td>multiply and divide</td>
</tr>
<tr>
<td>+ -</td>
<td>add and subtract</td>
</tr>
<tr>
<td>.GE.</td>
<td>relational operations</td>
</tr>
<tr>
<td>.AND.</td>
<td>logical operations</td>
</tr>
</tbody>
</table>

Note that the dots ., on either side of .GE. .GT. .EQ. .LT. .LE. .AND. .OR. or .NOT. can be omitted only when the operator appears between brackets.

**Other features**
Wild characters are allowed in character strings. The wild character is represented by '*' to replace any single character and a '?' to replace any number of characters (including zero characters). Examples of the uses of wild characters are, e.g.:

- '?' = 'PAW' or 'BACK' or any string of two or more characters, with the second character being 'A'.
- '*' = 'B' or 'ROB' or any string ending in 'B', with any number of characters preceding the 'B'.
- '?????' = 'ABCD' or 'abcd' or any string with exactly four characters.

More than one wildcard character can be used in a string and they can occur in any position. A standard substring selection notation STRING(j:k) is allowed in character strings, with j and k being integers satisfying the condition 1≤j≤k. As an example, if one has a Ntuple
variable of type character NAME = 'BOB' then both tests NAME(1:2) = 'BO' and NAME(1:) = 'BOB' will be true.
Mixed mode expressions are not allowed. For example, if X is of type Integer, then (X='TRUE') is an invalid test. Complex expressions can be constructed using the various operands and operators. The result of
one operation can be the operand for the next operator, but the data types must be compatible.

More on operations

In this section we review in more detail how the various allowed operators can be combined and what the type of
the result will be. The notation used is the following: N for numerical, C for a character string, L for logical, B bit string and
Bo for a boolean result.

The numerical operators

For the numerical operators **, *, /, - and + the operands and the result are both of type Numeric (N). This means that
any numerical combination of constants and Ntuple variables must be of type Real or Integer.
Example: if X is an Ntuple variable of type Real or Integer, then the result of the following examples is also numeric.

! weight X by Y
NTUPLE/plot 30.X Y
NTUPLE/plot 30.X X**2+Y**2

The relational operators

EQ , .EQ. = and NE, .NE. <>

operand(s): N or B or C - result: Bo

Mixed types are not permitted, i.e. an Integer cannot be compared to a Character string. (But Integer and Real types can be
compared).
Example, if X is of type Real or Integer, Y is of type Bit string and Z is of type Character string, then the following a valid commands:

NTUPLE/PLOT 30.Y X.EQ.9
NTUPLE/SCAN 30 Y>=2'8FFB'
NTUPLE/SCAN 30 Z='PAW'

GT, .GT., >, LT, .LT, <, GE, .GE, >=, LE, .LE, <= and CT,

operand(s): N - result: Bo

Example, if W and X are of type Real or Integer, then the following are valid commands:

NTUPLE/PLOT 30.W (W*3).GT.50
NTUPLE/SCAN 30 X>=39.5

Note that double comparisons of the type

NTUPLE/SCAN 30 Y<10<X

are no longer permitted and must be replaced by

NTUPLE/SCAN 30 (Y<10)AND(X>10)

The logical operators

AND, .AND., OR, .OR. and NOT, .NOT.

operand(s): Bo or L - result: Bo

These operands AND and OR can be used to join expressions which have a Boolean result or compare logical values. The
NOT command can be used on a Boolean expression or on a
Ntuple variable that has a Logical type.
Example, if W and X are of type Real or Integer, Z is of type
Character string and V is of type Logical, then the following are valid commands:

NTUPLE/SCAN 30 (W.EQ.4)AND(Z='CERN').OR.X=5
NTUPLE/PLOT 30.X (Z.NE.'CERN').OR.(NOT(V))
NTUPLE/SCAN 30 W.AND.T
NTUPLE/PLOT 30.W .NOT.((Z.EQ.'CERN').OR.V)

Note that AND, OR and NOT can no longer be represented by
*, +, -.

Functions

The table below contains the functions which can be used
with Ntuple commands. The argument n must be of type N
(Integer or Real) and can be any Ntuple variable, a constant
or an expression with a numerical result. The operand and
result are both of type N (numerical)

SIN(n) COS(n) SQRT(n)
EXP(n) LOG(n) ATAN(n)
ABS(n) INT(n) LOG10(n)
TANH(n) ACOS(n) ASIN(n)
TAN(n) SINH(n) COSH(n)
MOD(n,n) ATAN2(n,n) SIGN(n,n)

Bit handling routines

These are the routines belonging to BITPAK (CERN Program
Library M441) The operand(s) and result are both of type b
(bit string). In the discussion below, m and n, of type B (bit
string or unsigned integer), can be an Ntuple variable, a bit
string constant entered in the format discussed before or an
expression with a bit string result.

1. Logical operations

(a) IOR(m,n) inclusive OR of bit strings m and n.
(b) IAND(m,n) logical AND of bit strings m and n.
(c) NOT(n) logical complement of bit string n.
(d) IEOB(m,n) exclusive OR of bit strings m and n.
2. **Shift operations** The shift count is represented by \( k \).
   The absolute value of \( k \) specifies the number of positions to shift, while its sign specifies the direction of the shift, i.e. a left shift when \( k > 0 \), no shift when \( k = 0 \) and a right shift when \( k < 0 \).
   
   (a) ISHFT(m,k) provides the value of the argument \( m \) with the bits shifted. Bits shifted out to the left or right are lost, and zeros are shifted in from the opposite end.
   
   (b) ISHFTC(m,k,i,c) provides the value of the argument \( m \) with the rightmost \( i \) bits shifted and the remaining bits untouched. The shift is circular; i.e. no bits are lost.

3. **Bit subfields**
   
   (a) IBITS(m,i,1en) provides, right justified, the value of the 1en bits of the argument \( m \), starting from position \( i \).
   
   (b) MVBITS(m,i,1en,n,j) moves 1en bits of argument \( m \), starting at position \( i \), to the argument \( n \), starting at position \( j \). All other bits of \( n \) are left untouched. Arguments \( m \) and \( n \) may refer to the same numeric storage unit.

4. **Bit testing**
   
   (a) IBSET(n,i) has the value of argument \( n \) with bit \( i \) set to 1.
   
   (b) IBCLR(n,i) has the value of argument \( n \) with bit \( i \) set to 0.

An example of a bit string function is:

\[ \text{NTUPLE/SCAN 30 IAND(X,B'1010'),EQ.B'0101'} \]

The function BTEST has bit string arguments, but returns a boolean result.

   (a) BTEST(n,i) has the value 'TRUE' if bit \( i \) of argument \( n \) is set, and 'FALSE' otherwise.

**Restrictions for the use of the new Ntuples**

The following PAW commands cannot yet process the new style Ntuples:

- Ntuple/Create
- Ntuple/Read
- Ntuple/Merge
- Ntuple/Line

When structures are defined within a Ntuple, the only way to access these structures is via a COMIS function. In this case, however, the skeleton function may be generated via the command UWFUNC.

---

**Graphics developments**

**Improvements in histogram plotting**

The routines for drawing axes have been improved in the following areas:

- The routine used to optimize the axis labels has been modified in order to have better automatic labelling.
- The second and third levels of tick marks are now optimized like the first level.
- In the case of logarithmic axis, intermediate divisions are drawn if less than one decade is requested.

The option '+' in the command HISTO/PLOT now also works for histogram subranges and for a logarithmic scale (option LOD), e.g. add the contents of channels 11 to 30 of histogram 20 to those of channels 1 to 20 of histogram 10:

```
PAW > H/PLOT 10(1:20)  
PAW > H/PLOT 20(11:30) +
```

A bug was fixed in the command OPTION, because LINY was not correctly reset with OPTION *.

The command HIST/PLOT now can draw Multiquadratic fits produced by the SMOOTB command as a contour plot on top of the normal scatter plot of a bidimensional histogram.

The SET command now manages all IGSET options, i.e. if the requested option is not available under SET, it is passed to the IGSET command for possible processing.

HBOOK now allows histograms with alphanumeric binnings. The HPLOT package has been changed in order to plot automatically these new labels.

With OPTION ZFL1, the same picture name ("PICT00") is used for every new picture.

One can now specify with one SET command both border and inside colors for the Histogram, Box Page, and Function (HCOL, BCOL, FCOL, PCOL).

An example in the case of HCOL is shown below. The same mechanism also works for BCOL, FCOL and PCOL.

```
+----- The Histogram is filled  
|----- The border color is 2  
| | - The inside color is 3  
| |  
VVVV
PAW > SET HCOL 1203
```

**Improvements in the area of graphics**

In the X11 version of PAW, the graphics window is raised only if some graphics has been produced, i.e. if the execution
of a command or a macro does not produce any graphics, the graphics window is not raised. In the previous version the graphics window was raised after each command.

Improvements in the area of PostScript

PostScript text is now fully compatible with the escape characters of the TEXT command (see the HICZ/HPLOT). Default text is in Roman and text is case-sensitive.

<table>
<thead>
<tr>
<th>List of escape characters and their meaning</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; go to lower case (optional)</td>
<td>&gt; go to upper case (optional)</td>
</tr>
<tr>
<td>[ go to greek (Roman = default)</td>
<td>] end of greek</td>
</tr>
<tr>
<td>&quot; go to special symbols</td>
<td># end of special symbols</td>
</tr>
<tr>
<td>~ go to ZapfDingbats</td>
<td># end of ZapfDingbats</td>
</tr>
<tr>
<td>↑ go to superscript</td>
<td>? go to subscript</td>
</tr>
<tr>
<td>! go to normal level of script</td>
<td>&amp; backspace one character</td>
</tr>
<tr>
<td>$ termination character (optional)</td>
<td></td>
</tr>
</tbody>
</table>

The control characters can be escaped with 'Q'.

| print (in PS file) "abc" in Greek         | PAW > ITX 1.1. '[abc]'           |
| print (in PS file) "[abc]" literally      | PAW > ITX 1.1. '@[abc@]'         |

The PostScript marker size is now correct (previously is was wrong) if several pictures are produced in the same file.

New and modified PAW commands

Commands improvements

* NTUPLE/PLOT IDN [ UWFUNC NEVENT IFIRST NUPD OPTION ]

** COMMANDS IMPROVEMENTS **

| IDN | C 'Ntuple Identifier'
| UWFUNC | C 'Selection function' D='0'
| NEVENT | I 'Number of events' D=9999999
| IFIRST | I 'First event' D=1
| NUPD | I 'Frequency to update histogram' D=1000000
| OPTION | C 'Options'

New Possible values for OPTIONS are:

C Draw a smooth curve.
S Superimpose plot on top of existing picture.
+. Add contents of IDN to last plotted Ntuple.
B Bar chart format.
L Connect channels contents by a line.
P Draw the current polynmerker at each channel or cell.
+ Draw a at each channel.
U Update channels modified since last call.
E Draw error bars with current marker.
A Axis labels and tick marks are not drawn.
,Draw the Ntuple as an histogram.
PROF Fill a Profile histogram (mean option).
PROFS Fill a Profile histogram (spread option).

Options PROF and PROFS can be used to produce a profile histogram, e.g.

\[ \text{Ntuple/plot id.y}\%x \text{ ! ! ! ! PROF} \]

Ntuple/plot.id.sqrt(y)/x**2 option=PROF

* HISTOGRAM/FIT ID FUNC [ CHOPT NP PAR STEP PMIN PMAX ERRPAR ]

Options 'B', 'L' or 'E' do not influence the graphics output. To select M:NOS error evaluation (option E) and the graphics option 'E', select CHOPT=EE.

* /NTUPLE/CUTS CUTID [ OPTION FWNAME ]

CUTID C 'Cut identifier'
OPTION C 'Options' D='P' R='G,P,,-,R,W,D'
FWNAME C 'File name' D=''

Possible values for OPTIONS are:

G Define a new cut CUTID using graphics input on the latest 1-Dim or 2-Dim projection of the Ntuple. For a 1-Dim projection, give 2 points cutmin and cutmax. For a 2-Dim projection, give up to 20 points to delimit the selected area. The polygon will automatically be closed by PAW.
P Print definition of cut CUTID.
- Reset cut CUTID.
R Read definition of cut CUTID from file FWNAME.
W Write definition of cut CUTID on file FWNAME (text file).
D Draw cut contour.

The cut identifier CUTID is defined with the format $nn, where nn is an integer between 1 and 99. This cut can then be used in subsequent commands NTUPLE/PLOT or NTUPLE/PROJECT.
A cut CUTID can also be define with an OPTION parameter of the form OPTION='expression', e.g. a cut $\text{S1}$ is defined with the command:

```
PAW > CUTS $\text{S1}$  $\text{X}<0.8$ .and.$\text{Y}<\text{SQRT}($$\text{X}$$)
```

Note that CUTID=$\text{S0}$ means all cuts except with the 'G' option. When the latter option is selected, graphical cuts are only operational for plots of the original Ntuple variables, not for expressions of these variables.

* HISTOGRAM/OPERATIONS/ADD
* HISTOGRAM/OPERATIONS/SUBTRACT
* HISTOGRAM/OPERATIONS/MULTIPLY
* HISTOGRAM/OPERATIONS/DIVIDE

```
ID1 ID2 ID3 [C1 C2 OPTION]
```

A supplementary sixth parameter of type character has been introduced. When it has a value of 'Y', errors on the resulting histogram ID3 will be calculated correctly, assuming that histograms ID1 and ID2 are independent.

* HISTOGRAM/OPERATIONS/SMOOTH ID

```
[ OPTION SENSIT SMOOTH ]
```

```
ID C 'Histogram or Ntuple Identifier'
OPTION C 'Options'
SENSIT R 'Sensitivity parameter'
SMOOTH R 'Smoothness parameter'
D=1. R=0.3:3.
D=1. R=0.3:3.
```

Possible values for OPTION are:

0 Replace original histogram by smoothed.
1 Replace original histogram by smoothed.
2 Store values of smoothed function and its parameters without replacing the original histogram (but see note below) - the smoothed function can be displayed at editing time - see HISTOGRAM/PLLOT.

M Invoke multiquadric smoothing.
Q Invoke 3DQH algorithm (HBOOK routine HSMOOF).
S Invoke spline smoothing.
V Verbose (default for all except 1-D histogram).
N Do not plot the result of the fit.

Smooth a histogram or "simple" Ntuple. ("simple" = 1, 2, or 3 variables.)

For multiquadric smoothing, parameter SENSIT controls the sensitivity to statistical fluctuations. SMOOTH controls the (radius of) curvature of the multiquadric basis functions.

Notes:

1. The multiquadric basis functions are of the form $\text{SQRT}(R^2 + D^2)$, where $R$ is the distance from the "centre", and $D$ is a scale parameter and also the curvature at the "centre". "Centres" are located at points where the 2nd differential or Laplacian of event density is statistically significant.

2. The data must be statistically independent, i.e. events (weighted or unweighted) drawn randomly from a parent probability distribution or differential cross-section.

For spline smoothing, SENSIT and SMOOTH control the number of knots ($= 10 \times \text{SENSIT}$) and degree of splines ($= \text{SMOOTH}+2$) (thus if SENSIT and SMOOTH are at their default values a 10-knot cubic spline is used).

Notes:

1. The spline option always replaces the contents of a 2-D histogram. (Also the $\chi^2$ is unavailable in this case.)
2. Use the SPLINE command for more flexibility.

* /GRAPHICS/ATTRIBUTES/PALETTE PALNB

```
[ NEL LIST ]
```

```
PALNB I 'Palette number' D=0 R=0:9
NEL I 'Number of palette elements' D=0 R=0:50
LIST I 'List of palette elements' D=0
```

Define a palette of attributes.

The palette number is used with the command SET. The command SET HCOL 0.1 defines the palette number 1 as colour indices used by the command LEDG in case of stacked lego plots and plotting of SURFACE with options 1 or 2. LEDG with option 2 and CONTOUR with option 3.

By default the palettes are initialized with 6 elements (2,3,4,5,6,7).

If the number of elements (NEL) is equal to 0 (default), the palette is filled automatically according to the number of colours defined with the command IGSET HCOL as follows:

If HCOL$\leq$8 the palette is filled with a subset of the 8 basic colours.

If HCOL$>8$ the palette is filled with colours varying continuously from blue to red. This is called a "geographical palette".

Note that after a command IGSET HCOL, the color indices from 8 to HCOL are set with gray levels. The command PALETTE 1 resets the same indices with a "geographical palette" varying continuously from blue to red.
NEW PAW commands

* PICTURE/PRINT [ FILE ]

FILE C 'File name' D=1,

Print the current picture. The current picture is transformed
into a printable file. The file type is defined according to the
extension of the file name i.e.

FILE Type of file generated
FILE = filename.ps PostScript (-111)
FILE = filename.eps Encapsulated PostScript (-113)
FILE = filename.tex \LaTeX{} (-778)

Typing HELP META will show details about metafile types.
Note that a new picture is automatically created for each new
plot if OPTION ZFLI is set.

If FILE=HIGZPRINTER or FILE=\texttt{''}, the PostScript file
paw.ps (-111) is generated and the operating system command
defined by the environment variable HIGZPRINTER is executed.
The latter variable should be defined as follow:

On UNIX systems:

```
setenv HIGZPRINTER 'lp -dprinter_name paw.ps'
```

On VAX/VMS systems:

```
HIGZPRINTER = "XPRINT paw.ps /PRINTER=printer_name"
```

On CERNVM:

```
setenv HIGZPRINTER 'XPRINT PAW PS (PR printer_name)'
```

When the environment variable HIGZPRINTER is undefined,
the file paw.ps is created but not printed.

* NTUPLE/DRAW IDN [ VALUE OPTION ]

IDN C 'Ntuple Identifier'
VALUE C 'Isosurface value for 3-D' D=0'
OPTION C 'Options' D=1,

Draw a simple Ntuple (1, 2 or 3 variables). For simple N-
tuples, with 1, 2 or 3 variables per event, this command will
draw a histogram with H PLOT options. If the Ntuple has
an associated functional representation, as the result, e.g., of
using SMOOTH, it will also draw the function. No selections
are allowed.

For 3-variable Ntuples which have been SMOOTHed, give a
VALUE for the iso-surface of event density. If VALUE=0, an
iso-surface value half way between the minimum and maxi-
mum fitted smoothing function values will be used.

* NTUPLE/WAVE IDN [ LUN ]

IDN C 'Ntuple Identifier'
LUN I 'Logical unit no.' D=-1

Produce a formatted file suitable for Wavefront's Data Visu-
aliser. Only for simple 3-variable Ntuples which have been
SMOOTHed. A file with logical unit number. LUN must previ-
ously have been opened with the FORTRAN/FILE command.

Commands for drawing Feynman diagrams

A new set of commands to produce Feynman diagrams is
now available. This is a first implementation, which may be
changed according to user's needs and suggestions.

* GRAPHICS/PRIMITIVES/HELI

```
[ X1 Y1 X2 Y2 R WI PHI ]
```

X1 R 'X coord. of start of helix' D=0.
Y1 R 'Y coord. of start of helix' D=0.
X2 R 'X coord. of end of helix' D=10.
Y2 R 'Y coord. of end of helix' D=10.
R R 'Radius of helix' D=.3
WI R 'Number of turns' D=1.
PHI R 'Projection angle' D=15.

Plots a helix on a Feynman graph: gluon phi = 30, photon phi = 0.

* GRAPHICS/PRIMITIVES/ARCHHELI

```
[ X1 Y1 X2 Y2 R WI PHI RL ]
```

X1 R 'X coord. of start of helix' D=0.
Y1 R 'Y coord. of start of helix' D=0.
X2 R 'X coord. of end of helix' D=10.
Y2 R 'Y coord. of end of helix' D=10.
R R 'Radius of helix' D=.3
WI R 'Number of turns' D=1.
PHI R 'Projection angle' D=30.
RL R 'Radius of loop' D=15.

Plots a helix arc on a Feynman graph: gluon phi = 30, photon phi = 0.

* GRAPHICS/PRIMITIVES/ARLINE

```
[ X1 Y1 X2 Y2 H ]
```

X1 R 'X coord. of start' D=0.
Y1 R 'Y coord. of start' D=0.
X2 R 'X coord. of end' D=10.
Y2 R 'Y coord. of end' D=10.
H R 'Arrow size' D=.5

Line with arrow (fermion line)

* GRAPHICS/PRIMITIVES/FPOINT [ X Y R ]

X R 'X' D=0.
Y R 'Y' D=0.
R R 'Radius' D=.5

Draw filled point (vertex)
* c-c system
Arl ine 13 8 10 8 .3
Arl ine 10 10 13 10 .3
Arl ine 10 8 10 10 .3
* proton
Arl ine 4 5 8.5 5 .3
Arl ine 4 5 5 8.5 5 5 .3
Arl ine 4 6 8.5 6 .3
Line 8.5 6 13 4.5
Line 8.5 5 13 5
Line 8.5 5.5 13 5.5
* gluon
Helix 10 8 8.5 6 .3 7 30
* lepton
Arl ine 4 13 8 12 .3
Arl ine 8 12 13 13 .3
* photon
Helix 8 12 10 10 .1 4 0
* vertex
Fpoint 8 12 .1
Fpoint 10 10 .1
Fpoint 10 8 .1
Fpoint 8.5 6 .1
Itx 12.5 10.1 'c'
Line 12.5 8.45 12.7 8.45
Itx 12.5 8.1 'c'
Itx 12.5 13.1 'e'='1'
Itx 4.5 13.1 'e'='1'
Itx 4.5 6.2 'P'
Itx 9.3 11.1 '[g]'
Itx 5.5 8.5 'g'

Figure 2.3: Example of a Feynman diagram with fermion, gluon and photon lines

* photon loop
Arl ine 2 3 4 3 .2
archelix 4 3 10 3 .5 11 30 3.01
fpoint 4 3 .1

fpoint 10 3 .1
archelix 10 3 4 3 .5 11 30 3.01
arline 10 3 12 3 .2

* gluon loop
Arl ine 2 11 4 11 .2
archelix 4 11 10 11 .15 6 0 3.01
fpoint 4 11 .1

fpoint 10 11 .1
archelix 10 11 4 11 .15 6 0 3.01
arline 10 11 12 11 .2

Figure 2.4: Example of a Feynman diagram with gluon and photon loops
PIAF is a farm of 5 new HP755 workstations. Each workstation has 128 Mbytes of memory and 8 Gbytes of RAID. This system is currently in the development phase and it will be installed at the CERN computer center in the first quarter of 1993 when high speed connections to the other CERN systems will be possible. It will be available to all PAW users running the standard version of PAW on a workstation with the TCP/IP software.

PIAF will execute in a transparent way all time consuming commands (Ntuple/plot, Ntuple/project, Ntuple/loop, etc.). The software to stage Ntuples files from tapes, disk files, NFS, AFS files is currently under development. The 5 HP755 can process a user query (Ntuple/plot) in parallel using the PVM system.
2.2 KUIP developments

Nicole Cremel, Alfred Nathaniel /CN

A large fraction of KUIP has been rewritten in the C programming language. The new version is (mostly) backwards compatible with the old version, but because of the large amount of new code was involved, we cannot exclude that some incompatibilities remain.

Known incompatibilities

Alias names may contain only letters, digits, @-signs and underscores.
Commands /KUIP/HELP and /KUIP/SET_SHOW/MODE have been removed.

Improvements

There is no limit on the number of aliases anymore. Since KUMAC variables are treated internally as a special kind of alias complex macros could hit the limit of 200 aliases before.
The HELP command recognizes now abbreviations for submenu names. E.g. HELP H/OP works now while before HELP HISTOGRAM/OPERATIONS had to be spelled out.
The HELP command does a simple text formatting. The line length is controlled by /KUIP/SET_SHOW/COLUMNS.
The HELP -EDIT option is now implemented for all systems.
A new command /KUIP/SET_SHOW/HOST_PAGER defines the pager used to display the help text. E.g. on a Unix workstation define HOST_PAGER 'xterm -e view &' to display the help text in a separate window.
A new command /KUIP/SET_SHOW/FILECASE allows now to select the case sensitivity for filenames on Unix systems.
Before this functionality was hidden as STYLE 'FILECASE ON'.
A new command /KUIP/SET_SHOW/RECALL_STYLE to define the command recall and editing style. There is a choice between ksh/emacs-like binding and the key definitions used by DCL command line editing on VAX/VMS.

The KUIP/Motif interface

This new version of KUIP also features an interface to Motif to adapt a KUIP based application with minimal effort to that new powerful graphical user interface. This will allow you to get:

- multiple instances of a browser for application definable objects;
- optional HIGZ graphics window(s) with object identification;
- hooks for binding your own Motif code to buttons or pulldown menus in the Executive window (see PAW++);
- will be supported on Unix and VMS workstations with X11R4 and Motif1.1.

In order to use these new features, you will have to do the following:

- you do not have to learn programming with Motif;
- you have to change a few KUIP initialization calls;
- to incorporate your own application specific objects into the browser you have to:
  - write "small" routines to scan through the list of objects;
  - describe in the CDF the object types ("classes") and action menus for these classes.

KUIP/Motif in a Nutshell

Design Goals

- Any application already using the basic KUIP command line interface can provide a graphical user interface with Motif "look-and-feel". You do not have to get involved in Motif programming yourself!
- An application can easily incorporate its own objects into the browser by adding class definitions to the CDF and providing simple routines for scanning the object lists.
- The necessary changes are only in the initialization phase. Therefore the application needs only one library which can be loaded with different versions of the main program in order to produce a command line or a Motif interface.
- Basic KUIP and KUIP/Motif co-exist in the same PACKLIB file. The command line interface does not have to load any Motif code.
- If the application does not use graphics KUIP/Motif does not load any HIGZ code.
- You can distribute the Motif version of your application as a single executable module. Additional resource files are not mandatory.

Browser Concepts

The browser allows to traverse a hierarchical directory structure of objects. The objects contained in the currently selected directory can be displayed in various forms: big icons, small icons, text only... Operations on these objects can be chosen from popup menus which depend on the object type ("class").
Objects

"Objects" can be any kind of entities handled by an application. For example, the browser is used by PAW for HIGZ pictures and HBOOK objects (1d-histograms, 2d-histograms, Nuples, ZEBRA/RZ directories), by GEANT for its data structures (volumes, materials, particles, ...), and by KUIP/Motif itself for the tree of commands and menus.

To make new object types accessible by the browser the application has to provide a scanning routine which, when called for the first time, returns the first object and subsequently returns the next object until no more objects are left. The browser passes to this routine the identification of the requested subdirectory and expects in return the object and class names together with a short and a long description text.

Classes

The classes are defined in the CDF with the directive

>Class class-name menu-title
[ big-icon small-icon ]

Example:

>Class id '1d histogram' big_id small_id

specifying the class name ("1d"), a title text for menus, and optionally the icons used by the browser. For each class two menus define possible actions if an object of this class is either selected in the browser or identified inside a HIGZ graphics window.

Browsables

All objects are part of container objects ("browsables") constituting the top level directory, e.g. ZEBRA/RZ files. Browsable classes are defined with the CDF directive

>Browse class-name menu-title scan-objects
[ scan-browsables ]

specifying the application routine to scan the objects contained in a directory. The >Browse directive is followed by the menu definition of possible actions, e.g. how to create a new object or delete all objects in the browsable.

There are two kinds of browsable classes: single instance and multiple instance. For a multiple instance class the application has to provide a second scanning routine from which the names of currently connected browsables of a given class can be retrieved. A second menu definition allows to specify the necessary actions to connect new browsables, e.g. how to open a file in which the objects are stored.

If only one object of a browsable class exists, for example the command tree, there is no need for a scan-browsables routine. Single instance browsables are always present in the browser.

Directories

A tree structure of objects can easily be achieved by defining a special class for subdirectories of a browsable. The corresponding CDF directive is the same as for simple object classes, ">Class", except that there must be a "/" in front of the class name. KUIP/Motif only imposes the following conventions:

- The first item in the action menu always means "switch into this subdirectory".
- The new directory path is formed by concatenating the old path, a "/", and the name of the directory object.
- The routine used for scanning the objects in the top level directory must be able to return the objects in any subdirectory with a given path.

Going upwards in the directory hierarchy is done by selecting a subring of the current path displayed in the browser window.

Action menus

Class specific menus are derived from sequences of action definitions

menu-text special-flags action-string
[ action-routine ]

following the >Class or >Browse directive. The menu pops up when pressing the right mouse button and the selected action is performed when the button is released. A double click with the left mouse button executes the first menu item.

The action can be specified as a string of commands and/or a routine which should be called. Constructs of the form "[var]" can insert identifications of the selected object in the command string:

- [this] is replaced by the object name, e.g. histogram "10".
- [that] is replaced by the short description text returned by the scanning routine and can thus be used as an alias name.
- [name] is replaced by the name of the browsable, e.g. "LUW21" for an ZEBRA/RZ file opened on unit 21.
- [root] is replaced by the path of the top level directory, e.g. "/LUW21".
- [path] is replaced by the complete path to the directory in which the object is contained, e.g. "/LUW21/MYDIR". Initially, [path] is set to [root].

The replacement values for [name], [root] and, if necessary, for additional variables have to be returned by the application routine which is called to scan the list of connected browsables. For single instance browsables [name] is substituted by the class name and the field menu-title of the >Browse directive is used as the definition of [root].

By default the commands in the action string are executed immediately, provided that all mandatory arguments are present.
If any mandatory argument is missing the corresponding command panel comes up where they can be filled in. This default behaviour can be modified in several ways:

- Pressing the CTRL-key when popping up the menu forces the command panel even if all mandatory arguments are specified.
- Putting a "-" in front of the command name forces the command panel as well.
- Putting a "+" in front of the command name never produces a command panel, independently of the state of the CTRL-key.

Customization of the Icons

The application can define the icons to represent objects in the browser. Each class definition allows to specify the names of "big" and "small" icons which are looked up in the table of available icon bitmaps.

To create a new icon bitmap you can use the X11 standard bitmap editor. For example, "bitmap small_1d_bm 20x20" allows to design a $20 \times 20$ pixel icon called "small_1d". Then the output file small_1d_bm containing "#define small_1d_width 20 ..." simply needs to be inserted into the CDF following the directive >Icon_bitmaps.

Graphics

KUIP/Motif is interfaced to the X11 version of the HIGZ graphics package. In order for the two packages to co-operate properly, applications using HIGZ have to declare this with the directive >Graphics in the CDF.

The new version of HIGZ allows to store the name and the class for every displayed object by calling the routine TGP1D. From this information KUIP/Motif can identify the object at the mouse pointer position and pop up the second action menu defined with the >Class directive. This mechanism is extensively used in the Motif versions of PAW (figures 2.1 and 2.2) and GEANT (figure 2.6).

Buttons

An application can create new buttons and pulldown menus in the Executive window using the CDF directive

>Button callback-routine button-label
>Button callback-routine menu-text menu-title

The routine can either be in C (with the usual Motif callback calling sequence) or in Fortran (called with the specified text strings in character arguments).

Hooks

If you want to get involved in Motif programming you are free to do so. KUIP/Motif provides hooks that an application can link its own Motif code in. The CDF directive

>Motif_customize fallback-proc widget-proc

defines two routines fallback-proc and widget-proc, which are called at the initialization phase. The first routine has to return the application specific fallbacks while the second routine is called for every top level widget (Executive window, browser instance, graphics window) created by KUIP/Motif.
Figure 2.6: PAW++ display showing various panels (case of an Ntuple)

- The upper left corner shows the Geant++ Executive Window.
- The lower left contains the GEANT++ File Browser, where the MATE (materials) file has been selected. Within that file the data for CO2 have been activated. A "pop-up" menu has been activated for the chosen 1-dimension histogram and The Smooth function has been selected (see below).
- The graphics window is seen top right and shows the results for all physical mechanisms in the material. These mechanisms are selected via the command panel /CONTROL/PLMAT, which is seen on top of the Executive Window.
  - The "Geant ++ Command_Panel5" panels, shown at the lower right, allow ??????
2.3 HBOOK changes and new routines

René Brun /CN

New HBOOK file format

The HBOOK files created by the new version of HBOOK are by default in exchange mode. They can be transported between machines using the standard binary FTP or they can be NFS mounted in a heterogeneous environment. The old HBOOK files can still be processed by HBOOK and PAW. A conversion program called RT2NEW is available to convert old files to the new format. This program also converts automatically old format Ntuples to the new Ntuples described below.

![Say where that program is!!!](image)

Transfer HBOOK files between systems

Between Unix machines with FTP

```
ftp remote
bin
get remote.hbook
```

CERNVM to Unix workstation

On CERNVM type the following:

```
FTP workstation
BIN F 4096 ! block size in bytes
PUT file.hbook
```

On VMS from any machine

```
ftp remote
bin
get remote.hbook
quit
```

```
resize -s 4096 remote.hbook;1
```

The resize command does not copy the file. It simply changes the header information, from 512 bytes record to 4096 bytes. The resize tool is available on request from the CERN Program Library.

Proposed HBOOK file naming convention

Users are encouraged to name their HBOOK files with the suffix .hbook, so that the PAW++ browser will be able to recognize these files automatically.

New Ntuples in HBOOK

With the new version of PAW/HBOOK the new Ntuple routines, as described in the current HBOOK manual, become operational. Below the more important aspects of the new Ntuples are discussed. For more details see section 3.2 of the HBOOK manual.

Data Types

The new Ntuples support the storage of all basic data types: floating point numbers (REAL*4 and REAL*8), integers, bit patterns (unsigned integers), booleans and character strings. The PAW command UWFUNC has been modified to generate the COMIS functions with the corresponding data types.

Data Compression

Floating point numbers, integers and bit patterns can be packed by specifying a range of values or by explicitly specifying the number of bits that should be used to store the data. Booleans are always stored using one bit. Unused trailing array elements will not be stored when an array depends on an index variable. In that case only as many array elements will be stored as specified by the index variable.

For example, the array definition NHITS(NTRACK) defines NHITS to depend on the index variable NTRACK. When NTRACK is 16, the elements NHITS(1..16) are stored, when NTRACK is 3, only elements NHITS(1..3) are stored, etc.

User Routines

A new Ntuple is booked and defined using the routines: HBNT, HBNAMS and HBNAMC. They are filled using the routines: HFNT and HFNTB. Information is retrieved using: HGNT, HGNTB, HGNTV and HGNTF. Note that routine HGNT cannot be used to retrieve information from a new Ntuple. Global Ntuple options (like buffer size) are set using HBSET. The Ntuple definition can be printed using HPRTNT and a user function to access the Ntuple data can be created with HUFWFUN.

Storage Model

To improve data access time and to facilitate the compression mechanism, new Ntuples are stored column wise, as opposed to row wise for old Ntuples. Column wise storage allows direct access to any column in the Ntuple. Histogramming one column from a 300 column Ntuple requires reading only 1/300 of the total data set. However, this storage scheme requires one memory buffer per column as opposed to only one buffer in total for the old Ntuples. By default the buffer
length is 1024 words, in which case a 100 column Ntuple requires 409600 bytes of buffer space. In general, performance increases with increasing buffer size. Therefore, one should tune the buffer size (using routine RBSET) as function of the number of columns and the amount of available memory. Highest efficiency is obtained when setting the buffer size equal to the record length of the RZ HBOOK file (as specified in the call to HROPEN). A further advantage of column wise storage is that Ntuples can easily be extended with one or more new columns.

Columns are logically grouped into blocks (physically, however, all columns are independent). Blocks allow users to extend Ntuples with private columns or to group relevant columns together. New blocks can even be defined after the Ntuple has been filled. The newly created blocks can be filled using the routine HFTNB.

Note that arrays are treated as a single column. This means that when you define only one array of NVAR elements the behaviour of the old Ntuples will be reproduced (with in addition data typing and data compression). It is however not recommended to use this technique, since the direct column access capabilities of the new Ntuples are lost.

Performance

Accessing a relatively small number of the total number of defined columns results in a huge increase in performance compared to the old Ntuples. However, reading a complete Ntuple will take slightly longer than reading an old Ntuple due to the overhead introduced by the type checking and compression mechanisms and because the data is not stored sequentially on disk. This performance increase will most clearly show up when analyzing new Ntuples with PAW where one typically histograms one column with cuts on a few other columns. The advantages of having different data types and data compression generally outweighs the performance penalty incurred when reading complete Ntuples.

New or modified HBOOK routines

New routines

The following routine HQUAD has been added to HBOOK. It was implemented by J. Allison (OPAL/Manchester) and is automatically called in PAW by the existing command SMOOTH (see above).

CALL HQUAD (ID, CHOPT, MODE, SENSIT, SMOOTH, NSIG*, CHISQ*, NDF*, FMIN*, FMAX*, IERR*)

This routine fits multiquadric radial basis functions to the bin contents of a histogram or the event density of an Ntuple. (For Ntuples this is currently limited to “simple” ones, i.e., with 1, 2 or 3 variables; all events are used - no selection mechanism is implemented. Thus the recommended practice at the moment is to create a “simple” Ntuple and fill it from your “master” Ntuple with the NTUPLE/LOOP command and an appropriate SELECT .FOR function.)

Input parameters:

<table>
<thead>
<tr>
<th>ID</th>
<th>Histogram or Ntuple ID.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHOPT</td>
<td>Character variable containing option characters:</td>
</tr>
<tr>
<td>0</td>
<td>Replace original histogram by smoothed.</td>
</tr>
<tr>
<td>3</td>
<td>find significant points and perform unstrained fit.</td>
</tr>
<tr>
<td>1</td>
<td>Same as ‘0’ above.</td>
</tr>
<tr>
<td>2</td>
<td>Do not replace original histogram but store values of smoothed function and its parameters. (The fitted function is regenerates from the values or the parameters with the FUNC option in HISTOGRAM/ PLOT for histograms or with NTUPLE/ DRAW for Ntuples.)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MODE</th>
<th>Mode of operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Same as MODE = 3 (see below).</td>
</tr>
<tr>
<td>3</td>
<td>find significant points and perform unstrained fit.</td>
</tr>
</tbody>
</table>

Output parameters:

| NSIG  | no. of significant points or centres found, i.e., no. of basis functions used. |
| CHISQ | chi-squared (see Remarks). |
| NDF   | no. of degrees of freedom. |
| FMIN  | minimum function value. |
| FMAX  | maximum function value. |
| IERR  | error flag, 0 if all’s OK. (Hopefully helpful error messages are printed where possible.) |

Remarks:

- Empty bins are taken into account. (Poisson statistics are used for the unweighted case.)
- The multiquadric basis functions are \( \sqrt{r^2 + \Delta^2} \), where \( r \) is the radial distance from its “centre”, and \( \Delta \) is a scale parameter and also the curvature at the “centre”.
- “Centres”, also referred to as “significant points”, are located at points where the 2nd differential or Laplacian of event density is statistically significant.
The data must be statistically independent, i.e., events (weighted or unweighted) drawn randomly from a parent probability distribution or differential cross-section, e.g., you cannot further smooth a previously smoothed distribution.

For histograms, the chi-squared (CHISQ) is that of the fit to the original histogram assuming gaussian errors on the original histogram even for low statistics, including the error on zero being 1. It is calculated like this even for a Poisson likelihood fit; in that case the maximum likelihood may not correspond to the minimum chi-squared, but CHISQ can still be used, with NDF (the no. of degrees of freedom), as a goodness-of-fit estimator. For Nuplexes, an internally generated and temporary histogram is used to calculate CHISQ in the same way.

The following routine RindHDIFFB (from the D0 group) has been included into HBOOK. It compares two histograms, bin by bin. The authors are R. J. Genik II, D. Gilliland, J. Linnemann, J. McCampbell and J. McKinley.

CALL HDIFFB(ID1, ID2, TOL, NBAD*, NBINS*, DIFFS*, CHOPT)

Input parameters:

ID1 The first histogram to be compared. The "reference" histogram in options A and C.

ID2 The second histogram to be compared. The "data" histogram in options A and C.

ID1, ID2 are a pair of 1, 2-dim, or profile histograms booked with the same number of bins.

TOL The tolerance for a passing the test. Under options S and C, TOL is a number between 0 and 1 which represents the smallest probability considered as an acceptable match. Thus TOL is the fraction of bins expected to fail by chance if ID1 and ID2 are drawn from the same distribution. Under option A, TOL is the degree of precision of match required for the test to be considered as passed, e.g. TOL=2.0 would mean that the value from the data bin had to be within 2 times the reference error of the reference mean to be considered as compatible.

NBINS* The dimension of the user-supplied DIFFS array.

CHOPT A string allowing specification of the following options: The default (no options selected) is for option S (statistical comparison), ignoring all underflow and overflow bins, and automatically correcting for the difference in events between ID1 and ID2. No such correction is done for profile histograms.

* N Use the absolute contents of each histogram, thus including the normalization of the histogram as well as its shape in the comparison.

D Debug printout, dumps the critical variables in the comparisons, along with indicators of its weight, etc.

0 Overflow, requests that overflow bins be taken into account.

U Underflow, requests that underflow bins be taken into account.

R Right overflow bin. For a 2-D histogram, it includes the X-Axis overflow bin in the comparisons. If the 0 option is used, this is automatic.

L Left underflow bin. Same as above, but the X-Axis underflow is used. The U option uses this automatically.

T Top overflow bin. Same as R but for the Y-Axis

B Bottom underflow bin. Option L for the Y-Axis

S Statistical comparison. For standard 1-D histograms, calculates the probability that both bins were produced from a Poisson distribution with the same mean. For large statistics, R and D greater than 25, the mean for each bin is the average of the bin contents of ID1, ID2, adjusted for scaling (not adjusted for scaling if option N is selected). For small statistics, the unbiased ultimately most powerful comparison is made. This returns the confidence level that the two bins came from a Poisson distribution with the same mean. For a profile histogram, calculates t-test probability that both bin means were produced from a population with the same mean. This probability is referred to in TOL and DIFFS. The S option should be used when comparing two sets of data. Using the S option when comparing data to a function or known reference yields poor results. In this case, the C option should be selected.

C Compatibility test. Calculates the probability that the data (from ID2) was produced from a distribution with the mean and error in the bin of the reference histogram (ID1). The test is for Poisson statistics for 1-D histogram, Gaussian statistics for profile histograms. The C option should be used when comparing data to either a function, or a known reference or calibration distribution.

A Absolute test. Here the test is on the number of standard deviations by which the data from ID2 deviates from the reference histogram (ID1) mean. The standard deviation is taken from ID1. TOL is the number of standard deviations, rather than a probability. An arbitrary tolerance interval may be formed by using BPAK and BPAKE to fill the reference histogram; asymmetric intervals may be implemented by setting TOL to 1.0 and choosing the error and "mean" so that the allowed interval corresponds to ±1.0 standard deviations.

Z Ignore in the comparison any bins with zero contents in ID1. The default action is to consider all bins as significant.
Output parameters:

**NBAD** - The number of bins failing the compatibility test according to the criteria defined by TOL and CHOPT.

**NBINS** - The number of bins in the comparison (the size of the 1-dim histogram, plus 0, 1 or 2, depending on whether the overflow and underflow channels are included). For a 2-dim histogram, this will have the total bins (X*Y) plus room for overflow bins along any of the axes.

**DIFFS** - An array of length the number of bins being compared, which gives the results of the test bin by bin (probabilities for options S and C, deviations for option A). Results are passed back in the form:

1-dim DIFFS(NX) for no over or underflow or DIFFS(0:NX+1), for overflow and/or underflow.

2-dim DIFFS(NX,NY) or DIFFS(0:NX+1, 0:NY+1).

Errors messages of HDIFFB:

**Warning:** Zero tolerance
The passed value TOL is equal to 0.

**Warning:** Only one comparison at a time, please
More than one type of comparison was selected. Only one of options S, C, and A may be used at a time. This is only a warning and the test defaults to the S mode.

**Warning:** Different binning
The XMIN values for a 1-dim histogram or the XMIN and/or YMIN values on a 2-dim histogram are different. This may give inaccurate results.

**Warning:** Weighted or saturated events in 2-dimensions
HBOOK does not compute error bars for two dimensional histograms, thus weighted event are not allowed, and HDIFFB can not compute the correct statistics. An answer is still given, but it is probably not right.

**Integral is zero!**
The sum of the content bins is zero.

Both histograms must be the same dimension
A 1-dim and a 2-dim histogram have been specified. In order for the routine to work, both must be the same dimension.

Both histograms must be the same type
Two different types of histograms have been specified. Both must be profile or non-profile, you can not have a mix.

Not enough bins DIFF to hold result
The parameter NBINS is less that the number of bins in the histograms.

Number of channels is different
The number of channels in the two histograms to compare are different. They must be the same before the routine will process the data.

U/O/L/R/Y/T/B Option with weighted events
HBOOK does not compute an error bar for over-/underflow bins, thus it may not be used with weighted events.

Weighted options and no HBARR
The user had not told HBOOK to figure the error bars

for the histograms. Therefore, the operations will not be valid.

Both histograms must be the same, weighted or unweighted
As it states, the histograms must be of the same type.

Statistical comments:

The methods used for the S and C mode are correct for unweighted events and Poisson statistics for 1 or 2 dimensional histograms. For weighted events, a Gaussian approximation is used, which results in DIFFS values which are too low when there are fewer than 25 or so "equivalent events" (defined under BSTATI) per bin. This is caused by either few entries or by wide fluctuation in weights. The result is that HDIFFB rejects to many bins in this case.

Comparisons for profile histograms assume Gaussian statistics for the S and C mode comparisons of the channel mean. Fewer that 25 or so events will result in DIFFS values which are too large. The result is that HDIFFB rejects too many event in these low statistic cases.

Axis labels and histograms

A new set of routines has been added to associate labels with histogram channels. This association can be made before or after a histogram is filled, but it has the advantage that the label information get stored in the histogram data structure, so that it is available for all future plots in an automatic way.

**CALL HLABEL (ID, NLAB, *CLAB, W, CHOPT)**

Associates alphanumeric labels with a histogram. This routine can be called for a histogram after it has been filled, and then the labels specified will be shown on the respective axes. The routine can also be called before a histogram (Ntuple) is filled, and in this case, when filling, a certain order can be imposed. By default the entries will be automatically ordered.

**ID**
Histogram identifier.

**INLAB**
Number of labels.

**CLAB**
Character variable vector (CHARACTER*(*) with NLAB elements (input and output).

**CHOPT**
Character variable specifying the option desired.

’A’ As ’N’ below.

’Y’ Add NLAB news labels read in CLAB to histogram ID.

’R’ Read NLAB labels into in CLAB from histogram ID.

’X’ X-axis is being treated (default).

’Y’ Y-axis is being treated.

’Z’ Z-axis is being treated.

’S’ If labels exist then they should be sorted according to:

’A’ Alphabetically (default);

’D’ Reverse alphabetical order;
'E' by increasing channel contents (after filling);
'V' by decreasing channel contents (after filling).
'T' Modify (replace) HLAB existing labels read from CLAB in histogram ID.

Notes:
- For one-dimensional histograms HLABEL can be called at any time.
- For two-dimensional histograms one must call HLABEL with option 'N' for each axis between the call to HBOOK2 and the first call to HFC2.

CALL HFC1 (ID, IPOS, CLAB, W, CHOPT)
Fills a channel in a one-dimensional histogram.

ID One-dimensional histogram identifier.
IPOS Number of the bin.
CLAB Character variable (CHARACTER*16) containing the label describing the bin.
CHOPT Character variable specifying the option desired.
   N Normal filling
   S or default Automatically sort
   U If this option is set and the channel does not exist then the underflow channel is incremented, else a new one is created.

Notes:
- If IPOS≠0, then the fill channel described by IPOS is filled.
- 'N' or 'S' add a new channel dynamically.
- Routine HLABEL can be called before or after HFC1.

CALL HFC2 (ID, IPOSX, CLAX, IPOSY, CLABY, W)
Fills for a two-dimensional histogram the channel identified by position IPOSX or label CLAX and position IPOSY or label CLABY with weight W.

ID Two-dimensional histogram identifier.
IPOSX Number of the X-bin.
CLAX Character variable (CHARACTER*16) containing the label describing the X-bin.
IPOSY Number of the Y-bin.
CLABY Character variable (CHARACTER*16) containing the label describing the Y-bin.
W Weight of the event to be entered into the histogram.

Notes:
- Routine HLABEL must be called before HFC2.
- If IPOSX≠0, then the channel described by IPOSX is filled.
- If IPOSY≠0, then the channel described by IPOSY is filled.
- If the channel described by IPOSX or CLAX does not exist, the underflow channel is incremented. Idem for IPOSY and CLABY.

An example of how to sort a 2-dim. histogram:
call HBOOK2
call HLABEL option N for X-axis
call HLABEL option N for Y-axis
(many) calls to HFC2
......
call HLABEL option S to sort

Another possibility is:
call EBOOK2
(many) calls to HF2 or HFILL
......
call ELABEL option N for X-axis
call ELABEL option N for Y-axis
call ELABEL option S to sort

Operations on histogram labels

LOGICAL FUNCTION HLABELQ (ID, CHOPT)

Varifies whether a histogram has labels.
ID Histogram identifier.
CHOPT Character variable specifying the option desired.
   ', return .true. if histogram ID has labels, else return .false.;
   'X' return .true. if X axis has labels, else return .false.;
   'Y' return .true. if Y axis has labels, else return .false.;

INTEGER FUNCTION HLABNB (ID, CHOPT)

Returns the number of labels for the axes of a histogram.
ID Histogram identifier.
CHOPT Character variable specifying the option desired.
   ', like 'X' below;
   'X' return the number of labels for the X axis;
   'Y' return the number of labels for the Y axis;

CALL HLGIN (ID, IPOS, CLAX, CHOPT)

Get channel label from channel number.
ID Histogram identifier.
IPOS Number of the channel.
CLAX Character variable (CHARACTER*16) containing the label corresponding to channel IPOS (output variable).
CHOPT Character variable specifying the axis desired.
   ', As 'X' below;
   'X' Get label of X axis;
Y'  Get label of Y axis;
Y'  Get label of Z axis.

CALL HLPOS (ID, CLAB, IP0S*, CLA9, CHO9T)

Get channel number from channel label.

ID  Histogram identifier.
CLAB  Character variable (CHARACTER*16) specifying
the label one is looking for.
IP0S*  Number of the channel (output variable). If no chan-
    nel has label CLAB a value IP0S=-1 is returned.
CHO9T  Character variable specifying the axis desired.
    'X'  Get label of X axis;
    'Y'  Get label of Y axis;
    'Z'  Get label of Z axis.

2.4 New Version 1.41 of CMZ

RCodemE

The following changes and fixes have been made in the new
version of CMZ. More information can be found in the HELP
pages of each command. A new version of the CMZ docu-
mentation is also available.

- CMZ files are now machine independent. They can be
  transported using the standard binary FTP or similar
tools. CMZ files can also be NFS mounted in a hetero-
genous environment. For example to transfer a CMZ
file from VXCERN to CERNVM:

```
$ FTP VXCERN
FTP > BIN F 512
FTP > GET DISK$3:[USER]FILE.CMZ FILE.CMZ
FTP > QUIT
```

- Deckname length 32 chars.

- Line length 255 chars.

- The following KUI9 variables are available inside a
  macro:

```
$CMZFILE  name of the CMZ file;
$CV9  current directory;
$VSFILE  version of the CMZ file FILE.
```

- Command GREP has a new option -S and decklist (with-
  out wildcards).

- Command READ has the new options -S and -H.

- Command COT9T has a new option -S instead of option
  Y.

Routine modified

CALL HOPERA (ID1, CHOP9R, ID2, ID3, C1, C2)

Input parameters:

ID1, ID2  Operand histogram identifiers
CHOP9R  Character variable specifying the operation kind
    of operation to be performed (+, -, *, /) and 'E',
    whether errors have to be calculated (see below).
ID3  Identifier of the histogram containing the result
    after the operation.
C1, C2  Multiplicative constants.

The option parameter CHOP9R has a new option 'E', which,
when selected, instructs HOPERA to compute the error bars
on the resulting histograms correctly, assuming that the input
histograms ID1 and ID2 are independent.
+KEEP, AAP1.
  Apollo specific code
+KEEP, AAP2.
  HPUX specific code
+KEEP, AAP.
+SEQ, AAP1, IF=APOLLO.
+SEQ, AAP2, IF=HPUX.

Of course it is still possible to have several sequence definitions in one deck. In this case it stays possible to have IF selections on keeps with identical names.

After editing a deck containing sequence definitions CMZ will automatically (re)load all definitions it finds into memory.

To be able to edit a KEEP that has a name identical to one of the KUIP system functions one has to put the keep name between single quotes, e.g., to edit the keep $DATE type: EDIT ‘$DATE’. Whithout quotes the current date will be substituted for $DATE.

- CMZ can be made filecase sensitive by using the command FILECASE. With FILECASE set to KEEP, no file-name case conversion will be done. For the time being the default is FILECASE CONVERT, which will always convert the filenames to lowercase (backward compatible).

- With command MKALL one cannot specify a KEEP name anymore. MKALL checks now for every CDE or SEQ it finds the date of change of the KEEP (stored when loading the sequence definitions in memory). When a number of KEEPs are stored in one deck they will all get the same date and time of change from the decks history records. Due to this fact it may happen, although you changed only one keep in a deck containing several KEEPs, that many more decks will be recompiled than you expected. This is because all KEEPs in the deck get the same, new, modification date.

- Command UPDATE has a new option -REF.
- Command USE_CORR has a new option -I.
- Command YCORR: Issuing the command:
  YCORR //file1 //file2 //file3 //userfile
  will make a correction cradle of all decks in
  //userfile compared to decks with the same name in
  any of the files //file1, //file2 and //file3. Be
  sure that the decks in //userfile appear only once in
  //file1...//file3.
- Command DIFF: DIFF * will compare the highest cycles of all decks in the current directory to the previous cycles.
- Command CFL: CFL -P * is equivalent to CFL -P.
- Command EDIT saves only decks which have been modified. So EDIT * followed by a save, results only in the storage of the decks that have actually been changed.
- Command DELETE: DEL -I * works on all directories when you are at the top level.
- Command TTOC splits decks with more than 10000 lines into separate decks, containing each no more than 10000 lines.
- Command MKALL has a new option -P (pilot).