DRAL
Daresbury Laboratory
Rutherford Appleton Laboratory

RAL Report
RAL-95-017

The ENGIN Mini-Manual

J S Wright

February 1995

Rutherford Appleton Laboratory Chilton DIDCOT Oxfordshire OX11 0QX

The ENGIN *mini*-manual

J.S.Wright
January 1995

Version 1.5

1. Introduction

This document is designed to get you started planning and executing your experiment and analysing your data. You will also find useful the PUNCH mini-guide which describes the operation of the ISIS VAX cluster and other computing facilities, and the GENIE manual which describes the functions of GENIE, a language for display and manipulation of ISIS histogram data. These are both obtainable from the computer support office. If you want to go more deeply into the refinement of your data, you should obtain a copy of "Profile Analysis of Neutron Powder Diffraction Data at ISIS", RAL report 92-032, by W.I.F.David, R.M.Ibberson and J.C.Matthewman, usually available from Richard Ibberson.

- Before you arrive to start your experiment, please make sure that you have pre-registered with ULS using the form sent to the original contact applicant for the experiment. This will also enable arrangements to be made for the issue of a visitors swipe card to allow access to the ISIS experimental hall.

- On arrival at ISIS, you must:

- Report to ULS for the issue of the access pass, and your film badge. These must be carried with you at all times during your visit.

- Meet up with your local contact.

- You or your local contact should then pick up a sample record sheet from the DAC.

- Before mounting your samples on the ENGIN positioner, you should:

- Thoroughly understand the safety rules for the ENGIN positioner (see section 3 in this manual), and its operating modes.

- Thoroughly understand the operation of the safety interlock system. Your local contact will explain this to you.

- On every occasion you enter and leave the ISIS experimental hall, except in the case of an emergency evacuation, you should use your visitors card at the swipe points to register your entry and exit. If two or more people are entering together, they must all swipe their cards.

In the event of any problems with the instrument, your first point of contact is your local contact. The contact numbers are listed below:

Name	RAL Extension	Home	Bleep
John Wright	5768	01235 812848 0181 902 4691	251

To key the RAL extension from outside the lab, key 01235 445768. To bleep, key 70. On hearing the high pitched tone, key the bleep number, followed by the number of the extension you are keying from. Wait for a further tone, then replace the receiver.

2. Controlling the Instrument

2.1 General Description of ENGIN

The ENGIN instrument consists of six major components. Firstly there are two nominally identical radial collimators, together with their associated detector banks. The collimators each have 40 vanes made from gadolinium oxide coated mylar, which allow the detectors to receive neutrons from a small volume (about 2mm in size, along the beam direction). For very large samples, one of the collimators can be rotated through 90° to give a larger sample space than the normal 300 mm cube.

The detectors have three horizontal rows, each of 45 elements, giving 135 elements per bank. Thus there are a total of 270 detectors altogether.

The fifth component is the positioner, described in detail later. This is capable of repeatable movements to 0.01 mm in x and y, 0.001 in z and to .01° in θ. It is capable of carrying up to 200kg at full extension of both the x and y axes without measurable deflection.

The final component is the incoming beam collimation system. This consists of a pair of slits machined from sintered boron carbide, mounted in a frame suspended from a linear translation table. The slits are made in a range of sizes (1, 2, 3, 4, 5, 10, 25 mm). The translation table can be driven over a range of 300 mm from its "home" position to a software limited position close to the sample. The software limit can be set interactively - see section 3.

In addition, a small class 2 semiconductor laser is suspended above the centre of rotation of the positioner, which is the main fixed reference point, and a TV camera looks down at this point, its output being visible on two monitors, one in the blockhouse and one in the control cabin.

2.2 Computing facilities

There are two terminals, an x-terminal and a Pericom terminal, in the cabin which are both directly connected to the TEB FEM. The x-terminal normally displays three DECterm windows, plus a clock. A beam current monitor may also be displayed by typing pppmon on any of the windows.

One window normally displays the Status Dashboard, showing run number and instrument status. This window is conventionally used as the supervisor window, for starting and ending runs and for executing command files. **Do not use this window for editing files.** Although not fatal, it can cause you to lose the dashboard.

The other windows are used for general purpose editing of command files and for a GENIE session on TEB. This is inclined to be very slow however and is generally only used for interactively focusing from the DAE on runs in progress.

The Pericom terminal is frequently and advantageously used as a terminal on ISISE, or some other cluster member, and will successfully display GENIE graphics.

2.3 The CHANGE command

The change command allows the user to edit the dashboard information and to modify its parameters.

Typing the command

$ **change** (can be abbreviated to **cha**)

will initiate the dashboard DEFT editor. This can be done on the supervisor window, as the dashboard will re-appear when the DEFT editor is exited. Move between fields using the cursor up and down keys. A whole page can be skipped by using keypad "7". When you are asked to toggle a field, the keypad "." key is used.

The only fields which you should need to change will be the title, your RB No, your name and the name of your university. These are all on the first page. You will perhaps need to change the binning of the data. This is on the last page.

Exit the change command using Gold (PF1) "e".

To change only the title, you should use a command like:

cha title """Iron Rod 3.2 mm dia, x=1.5, y=2.7, z=33"""

The triple quotes enable you to use spaces and lower case letters in the title. The title can be changed after a run has started. Since a run can take a few minutes to start, if using fine binning over the full spectral range, it is worth putting the change title command as above in your command files after the BEGIN command.

2.4 Data Collection Commands

All data files are stored in a directory called TEB$DUA0:[TEBMGR.DATA]. This is defined as a symbol in the TEBMGR account, and in yours if it has been set up for use on ENGIN, as teb_data.

All the following instrument control commands can be abbreviated to three letters.

begin Starts a run

update Stores the data collected so far in the current run parameter table (crpt)

store Stores the data collected up to the last update in a file named teb_data:teb*****.sav;* The store command should always be preceded by an update command. The update and store commands are the only ones in this group which can be issued from a window other than the supervisor.

pause Pauses Data collection

resume Resumes data collection

abort Aborts the current run without saving the data

end Ends the current run and stores the data in a file named teb_data:teb*****.raw;1

2.5 Using Command Files

Command files can be used to control the instrument. An example command file is shown below

```
$ set noon              ensures continuation of the file on error detection
$ motor x 15            moves the x axis to +15 mm from the datum
$ begin                 starts the data collection
$ cha title """x at 15 mm"""
$ waitfor 200 uamps     waits until the total current reaches 200 µamp.hr
$ update                updates the crpt
$ store                 stores the contents of the crpt in a .sav file
$ waitfor 400 uamps
$ end                   stops the run and stores the data in a .raw file
$ motor x 16            moves the x axis 1 mm to +16 mm
$ begin
$ cha title """x at 16 mm"""
$ waitfor 200 uamps
$ update
$ store
$ waitfor 400 uamps
$ end
```

The file should be saved using the .com extension - see the VMS manuals for help on use of the edt and eve editors. The TEB computer will default to the eve editor but you may also use edt by typing:

```
$ edit/edt filename.com
```

Command files can be run by typing:

```
$ @filename
```

Command files should be run from the supervisor window. If this is not done, then the begin and end commands may not be executed properly.

3. The Sample Positioning System

3.1 The ENGIN positioner

The ENGIN positioner consists of an x-y table with a number of mounting holes in it to enable samples to be firmly fixed to it (see appendix for dimensions). Each axis can move 125mm on either side of its normal datum point. This is mounted on a column which can move in the z direction over a total range of 300 mm. **Note that the maximum height of the x-y table is limited by the need to avoid collisions with the collimator supports**. The maximum height of the table is **310 mm** below the neutron beam. You will therefore need to arrange that your sample(s) can be brought to the correct height beforehand. The column sits on a direct drive motor which rotates the whole assembly in the θ direction through slightly less than 360°. The normal datum point of the θ axis is about 180° from the extremes of its travel. In this position, by convention, the x axis is along the incident neutron beam, with the positive direction being towards the moderator. The y axis is perpendicular to this, with the positive direction being to the right when viewed from above and facing towards the moderator.

There are two methods of communicating with the controllers: by use of the RP240 hand held controllers located in the blockhouse ("LOCAL" mode) and by means of an RS-232 connection to the μVAX-II instrument computer ("REMOTE" mode). The three controllers are "daisy-chained", so that only one terminal connection is needed to address them both.

The controllers and associated amplifiers and transformers are located in a rack next to the detector electronics rack, identifiable because it has a metal door with a very large key. On opening the door, the two Parker Compumotor 6250 servo controllers are seen at the top of the cabinet. Below them are the two Digiplan BL75 drives for the x and y axes, and below them is a further BL75 drive for the z axis, and a Dynaserv drive for the θ axis. Below these units is a small switch panel with green and red LEDs, an on/off switch and a reset button. At the bottom of the cabinet are the transformers providing power to the BL75 drives, the Dynaserv having its own built in power supply. The third controller, a Parker 6200 stepper controller, which drives the incoming beam collimation system, sits in a box on top of the rack.

3.2 Safety Rules

The drives are **extremely powerful** and could easily crush arms/legs/head if such items were in the way when the drives are activated. For this reason, a few simple rules are set out to minimise the risk of such an occurrence:

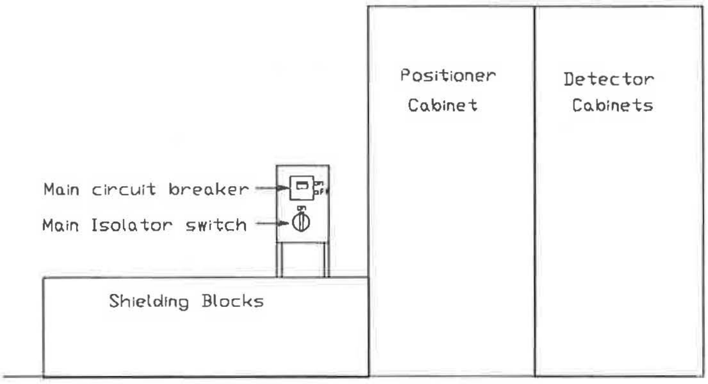
1. **Ensure you are familiar with the modes of operation of the positioner - if in doubt, ask.**
2. **Exercise extreme caution whenever any part of the body is within the movement range of the positioner, especially when changing samples.**
3. **Be aware of the position of the emergency stop button.**
4. **The positioner must be under local (RP240) control, and two persons should be present whenever samples are being changed.**
5. **NO-ONE should enter the area under the decking while power is applied to the positioner.**

3.3 Incoming Beam Collimation System

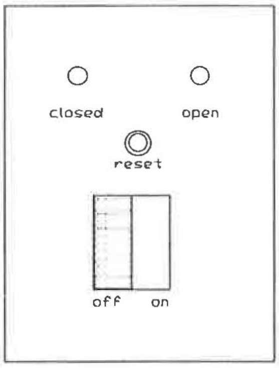
This is briefly described at this point as its control system is similar to that of the positioner. It is automatically powered up at the same time as the positioner - similarly, if the emergency stop button is pressed, the slits will stop moving as well. The carrier for the slits is driven on a linear table limited by software normally to a 300 mm travel. There is a remote control pod in the blockhouse which can be used to move the slits in and out, although no display of the actual position is possible. Measurement of its' position can be done via the RS232 link to the controller. The pod has a 4 buttons and two switches, although only two buttons and one switch are currently in use. The buttons are labelled "in" and "out", and the switch is labelled "fast" and "slow".

3.4 System Start-up

To start the system, first ensure the emergency off button located next to the left hand telescope mount in the Test Beam blockhouse is released. This requires the use of a key, normally only held by John Wright, R3 2.07. Once the emergency off button is released, switch on the main isolator and main circuit breaker next to the controller cabinet. Open the controller cabinet and check that the green "CLOSED" LED is alight. If the red "OPEN" LED is alight, press the reset button. Pick up the loose switch and simultaneously press this switch and the reset button below the green LED. The red and green LEDs on the front panels of the 6250 controllers should be alight, and "0. " should be visible on the Dynaserv amplifier. If this fails, a fuse may have blown. Contact John Wright for assistance.



View of cabinet and main switches



SWITCH PANEL INSIDE CABINET

Inspection of the RP240 local controllers will now show the message "Main Setup Complete", but no message will be shown on the µVAX-II. Closer inspection of the RP240's will show a vertical line of LED indicators, the one indicating remote mode will now be lit.

Since the next steps after switching on are likely to be initialising the system and alignment of samples using the telescopes, the normal means of control will be the RP240's. Use of these will thus be discussed first.

3.5 RP240 or Local Control

The RP240 (somewhat illogically the name means Remote Panel with a 2 line display of 40 characters) is a menu driven terminal connected to the 6250 controller via a dedicated RS232 line. Unlike the use of the remote RS232 line, these devices are not daisy chained. Thus they are completely independent. It is thus quite possible for one 6250 to be under local control and the other to be under remote control.

3.5.1 Description

The RP240 is a rectangular box with a grey front panel. On this panel are Dymo labels indicating which axes are controlled, and the maximum speed of the motors in revolutions per second. Below this is the display, and below the display are seven buttons labelled "Menu Recall" and F1-F6. Down the left side are 8 LEDs with appropriate legends, and on the right is a numeric keypad including, as well as number keys, a +/- button, a C/E (clear entry) and an enter button. Dymo labels are also placed close to these buttons to indicate the movement which will be activated in JOG mode. Most importantly, on the bottom left corner is a STOP button which if pressed will terminate motion. Be aware of the fact that the STOP button will only stop motion on the axes controlled by that RP240!

3.5.2 Operation of the RP240

As mentioned above, On first switching on the message "MAIN SETUP COMPLETE" appears on the RP240 display. Pressing Menu Recall now brings up the first of the menus:

```
        COMPUMOTOR 6250 SERVO CONTROLLER
 RUN JOG   STATUS      DRIVE        DISPLAY     ETC
```

Each of the commands on the second line sits above one of the function buttons. To bring the positioner under local control, we press the button under RUN, that is F1. A secondary menu now appears:

```
     Program to run is:
     FIND   ALPHA      <--     -->     TRACE      STEP
```

Pressing FIND now steps through the available options: MAIN, RP240, PARMS, GHOMEX, GHOMEY, and MOVE. The only one of real relevance is RP240, as the other options appear on later menus.

Press FIND until RP240 appears, then press ENTER on the numeric keypad. A new menu now appears:

```
LOCAL CONTROL                      MENU RECALL EXITS
     JOG    PARAMS  GO   HOMEX     HOMEY       POSN
```

This is the normal menu for most operations, so each operation will be covered in some detail. HOMEX and HOMEY are specific to the x,y controller. On the z,θ controller these are replaced by HOMEZ and HOMET.

3.5.3 HOMEX, HOMEY, HOMEZ and HOMET

Pressing any of these options will cause the relevant axis to search for its home position. The home positions are defined by magnetic proximity switches on the x,y and z axes, and by a known distance from the limit switch on the θ axis. On the x and y axes these are so positioned as to bring the centre of the x y table to the centre of rotation of the machine to within a millimetre or so. This is designed to speed up the alignment procedure, not to replace it. On the completion of the homing manoeuvre the encoder counter is set to zero for that axis.

CAUTION: when the machine has just been switched on, or for any other reason has lost its position references, use of the home commands may cause that axis to move to the extremes of its available travel. THIS IS PERFECTLY SAFE WHEN NO SAMPLE OR OTHER SAMPLE MOUNTING JIG IS MOUNTED ON THE PLATFORM. Mounting samples or jigs on the platform then performing a home operation may cause a collision to occur between the sample and the collimators or the incoming beam collimation assembly. The collimators are protected by pressure sensitive pads, but this will terminate power to the positioner and thus position information will be lost. The slit assembly is not so protected and may be damaged in a collision.

3.5.4 JOG

Pressing JOG brings up another menu:

```
AXIS 1: L/R          AXIS 2: U/D
LO 0.3000            LO 0.3000            EDIT            JOG*
```

This menu is similar on both the x,y controller and the z,θ controller, except that the default numerical values are different. AXIS 1 LO sits above F1, and AXIS 2 LO sits above F3. Pressing F1 or F3 changes the LO to HI. The values shown next to LO or HI can be altered by pressing F5, which sits below EDIT. When this is done, the asterisk moves from the right hand of JOG to the right hand of EDIT. The function button under the axis it is desired to change should now be pressed. The existing value can be cleared using C/E, and re-entered terminating with ENTER. Pressing F6 returns the asterisk to JOG. Motion will only occur when JOG is active rather than EDIT.

3.5.5 Movement of the X and Y Axes

On the numeric keypad there are arrows superimposed on four of the keys pointing left (4), right (6), up (2) and down (8). The direction of the arrows on the keypad corresponds to L/R and U/D on the display. It will be noted that when one of the arrow keys is pressed, the appropriate symbol on the display is underlined. In the case of the Left and Right arrows, motion corresponds to movement of the x axis. If the θ axis is at or near its home position, pressing the Left facing arrow will move the table away from the moderator, (the conventional negative direction) and the Right facing key will move it towards the moderator (the conventional positive direction). Similarly, the Upwards facing arrow (2 key) moves the y axis in a conventionally negative direction, to the left when facing the moderator or looking down from above, and the Down facing arrow moves the y axis in a conventionally positive manner.

3.5.6 Movement of the Z and Θ Axes

The correspondence between the numeric keys and the display is the same as above, but the actual motion is somewhat counter intuitive. Pressing the Left arrow moves the Z axis UPWARDS, and the Right arrow moves Z DOWNWARDS. Pressing the Down arrow moves the θ axis clockwise, the Up arrow moves θ anti-clockwise. Refer to the Dymo labels on the panel.

3.5.7 PARAMS

PARAMS is used to set the velocity and distance for the pre-determined movement of an axis. Pressing PARAMS brings up another menu:

```
LOCAL CONTROL                  SET MOVE PARAMETERS
      X            Y           BOTH
```

It is normally recommended that only one axis at a time is moved and thus the BOTH option is not normally used. Pressing X or Y brings up a further level:

```
LOCAL CONTROL                  SET MOVE PARAMETERS
ENTER X AXIS VELOCITY  ?
```

The value to be input here corresponds to the rotation speed of the motor in revs/second, and limits in the software do not allow them to exceed certain values, these being 4 rev.s^{-1} for the x and y axes, 20 rev.s^{-1} for the z axis and 0.05 rev.s^{-1} for the θ axis. Values larger than these will be rejected and the same question will reappear.

```
LOCAL CONTROL                  SET MOVE PARAMETERS
ENTER X AXIS POSITION ?
```

The value entered here not only sets the distance, but also the direction, so the values can be positive or negative. The distances moved are absolute (not incremental or relative). For the x, y and z axes, the values entered are in millimetres. For the θ axis the value entered is in degrees.

The sign of the numbers input is important. A positive number will drive the z-axis in the upwards direction. Positive numbers will drive the θ axis in a clockwise direction.

On pressing enter after keying the number, the previous menu is seen again. Pressing menu recall will now bring back the top layer menu:

```
LOCAL CONTROL                      MENU RECALL EXITS
      JOG   PARAMS  GO  HOMEX   HOMEY       POSN
```

3.5.8 GO

Pressing GO will now execute the move programmed using PARAMS. At the end of the move, the display will show the position of both axes.

```
X POS IS +00100.36                Y POS IS -00000.00
             PRESS MENU RECALL TO EXIT
```

The values displayed will be scaled, within the limits of accuracy imposed by the ballscrews and the fact that the servos may not have completely reached their target value. The display above was obtained after moving the x axis 100 mm. Five minutes later, after moving the y axis a similar amount, the display showed:

```
X POS IS +00099.99                Y POS IS +00100.34
             PRESS MENU RECALL TO EXIT
```

This shows how the value can change after a few minutes. To minimise this problem, the program waits for five seconds after the nominal completion of the move before displaying the message shown above.

3.5.9 POSN

Pressing POSN will bring up a display similar to the one described above under MOVE.

```
X POS IS +00099.99                    Y POS IS +00100.34
           ZERO X                                ZERO Y
```

This can be displayed at any time, however, not only immediately after a move has been made. It also allows each axis to be zeroed individually, allowing the counters to be reset once a reference point is reached. Pressing zero for either axis will cause the routine to exit. Pressing POSN again will show the new position, and allow the other axis to be zeroed if required.

3.6 Remote Mode Operation

3.6.1 General

In remote mode the positioner is under the control of its RS232 port, which is normally connected to the μVAX-II instrument computer. In this mode, all three controllers are daisy chained, and thus under most circumstances any commands are prefixed by "0_" for the x,y controller, "1_" for the z,θ controller and "2_" for the incoming beam collimation system. Leaving out the prefix causes a command to be executed by all three controllers. This can be used to send a command like !K, which stops all motion. Although the controllers may be in local mode, it is possible to force them to obey commands by prefixing the command with a "!" as shown above.

3.6.2 μVAX-II control

When under the control of the μVAX-II, a special set of routines passes commands down the RS232 link and displays the responses. These routines are called by using the logical name XYZT, which may be shortened to XY. There is also a higher level command accessed by the logical name MOTOR which simplifies the control of the positioner.

3.6.3 The XYZT Command

Because the argument of the command is expected to be a single string, and commands issued to the positioner may contain commas, spaces, underscores etc., it is always advisable to enclose the command string in quotation marks. Thus a command dialogue by the μVAX-II might have the following appearance:

```
$ xyzt "1_!tpe"
1_!TPE

 *-3485.7,-11265.0
COMMAND COMPLETED SUCCESSFULLY
DONE
$
```

This shows a call to the handling routine xyzt, followed by an argument in quotes. This command instructs the z,θ controller to pass back down the RS232 line the values stored in the encoder register. On sending the command, xyzt echoes the command to the screen. The 6250 then sends back the required information. This is followed by the sign off messages of the routine xyzt, "command completed successfully" and "done". These messages are not issued until all three controllers have been polled and have confirmed that movement has ceased.

3.6.4 The MOTOR Command

Motor is a higher level handling routine which applies scaling factors, and then calls the xyzt routine. This is the easiest form of command when using the positioner under μVAX-II control. The syntax is as follows:

```
MOTOR <motor symbol (x,y,z,t,s)> <position>
```

x,y,z,t describe the axis to be moved - or s for the incoming beam collimation slits, and position is a value in millimetres (or degrees for the θ axis). This value can be positive or negative, except for s, where conventionally the end stop furthest from the sample is zero. The motor command scales the distances entered, and alters the sense of the z motion, compared to use of the xyzt command. It is therefore recommended that only the MOTOR command is used to move the positioner. The command file sends commands to the controllers using the xyzt routine described above, and automatically sets the speed direction and axis to be moved.

3.6.5 Command Syntax

A small subset only of the rich (i.e. complex) command language used for the 6250 controllers is presented here. For further reference the reader is referred to the 6000 series software reference manual. Listed here are the commands which may be found useful most of the time

All commands in the 6000 series language are mnemonic in structure, and thus can be easily learned by those with a logical turn of mind. The important groups of commands are those which transfer information, and those which give us the current values of parameters or variables.

TPE - Transfer Position of Encoder

Gives the current value of the encoder registers in the order x,y (controller 0) or z,θ (controller 1). The values returned are in encoder steps (x,y), or scaled values (z,θ). For technical reasons, these scaled values are about 1-2% out.

TPM - Transfer Position of Motor

The stepper motor which drives the incoming beam collimation does not have an encoder, so TPM is used in place of TPE. The value of TPM is encoded within the 6200 controller to read directly in millimetres.

TPC - Transfer Position Commanded

Gives the current commanded position for each axis, as listed above.

TPER - Transfer Position Error

Displays the difference between the encoder value (see above) and the commanded encoder value. Order is the same as above.

The commands above are only informational. Following commands can be use to read information from the controllers or to input values to the controllers.

PSET - Establish absolute position.

This command is most usefully used for zeroing the encoder of one axis of the positioner - similar to the use of the RP240 command POSN, except that values other than zero can be used. The syntax is as shown for the D command below. The value must be in steps or scaled values.

D

This register holds the number of units to be moved on both axes, and the direction of movement. The values are in encoder steps (x,y) or in scaled values (z,θ). The movement will take place as an absolute motion rather than incremental steps. The format of the command varies according to the function. Typical syntaxes are shown below:

```
1_d
*+1000.0,+100.0
```

Sending D without any further argument returns the values in the register, in the order x,y or z,θ.

```
0_d5000,
```

Alters the value of D for the x (or z) registers, without affecting the value of y (or θ)

```
0_d,-5000
```

Alters the value of D for the y (or θ) registers, without affecting the value of x (or z)

```
0_d5000,-5000
```

Alters the values of D for both x and y (or z and θ).

V

V defines the speed at which the motor will run during the move. (the direction is defined by D, thus numbers in V are always positive). The syntax is exactly the same as for D.

VAR47

This is a variable used in the programs to define which axis will respond to a move command. When var47=1.0 then the x or z axes will be moved. Any other value of var47 means that the y or θ axes will move. Normally var47 is set explicitly to 0 or 1.0.

```
1_var47
*+0.0
```

Sending var47 with no argument returns the current value.

```
1_var47=1
```

Sending var47 with an argument as above changes the value of var47.

ZERO

Zero is a program in the 6200 controller which drives the incoming beam collimation system. The command format is xyzt 2_zero. This command moves the incoming beam collimation slit to its outer (away from the sample) end stop, then zeroes the register which holds the value of TPM. This allows compensation for any loss of steps in the stepper motor.

MOVE

Move is a program that performs a move on an axis. The axis to be moved is defined by var47, the speed by v and the distance and direction by d. Hence all these parameters must be set up before a move can be successfully executed.

3.7 Software Limits

In order to set software limits, the following commands must be issued. Normally the software limits are disabled on the x,y,z and θ axes and enabled on the incoming beam collimation system. To enable soft limits, the command:

LSn,m is issued. n and m can take the values 0 (disable both) 1 (disable clockwise) 2 (disable counter clockwise) or 3 (enable both). n represents the values for the x and z axes and the incoming beam slits, while m represents the values for the y and θ axes.

Limits are set using **LSCWn,m** (clockwise) and **LSCCWn,m** (counterclockwise) where n and m are the displacement values from the zero points as defined using RP240 POSN or PSET. Values are in steps (x,y) scaled values (z,θ) and millimetres (incoming beam collimation). These commands can be issued using the xy routine, using the appropriate 0_, 1_, or 2_ prefix.

3.8 Suggested Strategy For Use of the Positioner

If the positioner is already in use, then it may be acceptable to start from step 5

1. Switch on the positioner as described in section 3.4.

2. Set the controllers to RP240 (Local) control.

3. Remove all samples and jigs from the positioner.

4. Home the x, y, and θ axes using the RP240 commands.

5. Mount and align your sample. If the sample is aligned with the edges in line with the scribed lines on the positioner face, then moving the θ axis to **+52.82, -127.18, -37.2,** or **142.8** degrees using the RP240 command "Params" will line the sample up with the telescope line of sight.

6. Move the positioner using the RP240 "Jog" commands to bring a reference point on the sample to the centre of the field of view of the telescopes - i.e. to the centre of the sampling volume.

7. Zero the counters using the RP240 command "Posn"

8. Move the incoming beam slits as close to the sample as desired using the pod.

9. Use the MOTOR command for all subsequent displacements, remembering that the positioner moves in absolute mode.

10.Use the xy "n_tpe" or "n_!tpe" command to ensure that the positioner has gone where you want it to go. Use xy “2_tpm” to find the position of the incoming beam collimation.

3.9 Summary of RP240 Commands

MENU RECALL - always moves up one layer in the menu structure. Exits from RP240 mode to remote mode at the top level.

FIND - allows selection of a program to run, mostly used for RP240.

RP240 - sets up local control.

MAIN - initialisation routine, sets up variables and default values.

JOG - allows free movement of any axis, at one of two editable speeds.

PARAMS - prompts for entry of a: axis to move. b: speed of movement, c: distance and direction of movement.

GO - executes move according to parameters entered under PARAMS

HOMEX, HOMEY, HOMEZ, HOMET - sends axis to its home position. CAUTION - do not execute when samples or jigs are present on the table.

POSN - allows the position of the both axes to be viewed, and each axis to be zeroed individually.

3.10 Summary of Essential Remote Control Commands

MOTOR - logical used to call MOVE_MOTOR.COM command routine.

XYZT - logical used to call VMS command handling routines. May be shortened to xy

0_, 1_ 2_- address command to the appropriate controller 0 = x,y 1=z,θ, 2=s

! - forces execution of a remote command even when axis is under RP240 local control.

TPE - transfer position of encoder. Tells you where a pair of axes are.

TPC - transfer position commanded. Tells you where a pair of axes should be.

TPER - transfer position error. The difference between TPE and TPC

TPM - used with the incoming beam collimation system to read position of motor.

LS - Enables/disables software limits

LSCW - Soft limit clockwise range

LSCCW - Soft limit counterclockwise range

PSET - sets motor encoder value to 0 or to a desired value.

D - tells you the currently stored value of or assigns new values to the distance and direction to be moved for a pair of axes

V - tells you the currently stored value of or assigns new values to the speed of a pair of axes.

VAR47 - determines which of a pair of axes will move, VAR47=1 moves x or z axes, VAR47≠1 (normally set to 0) moves y or θ axes.

MOVE - executes a move set up by VAR47, D and V.

ZERO - sends the incoming beam slits to their zero position and zeroes the counter.

EXAMPLE OF COMMAND SYNTAX used when transmitting commands interactively or via a command file from the µVAX II:

```
$ xyzt "0_!tpe"
```

4.0 Instructions for Data Reduction on ENGIN

4.1 Objective

The objective of the routines described here is to simplify the data reduction to strain values for experiments on ENGIN. In addition, the output files include the peak width parameters and the anisotropic shift parameter. Since calibrations are performed at the start of each ENGIN run, the current version includes the facility to include the use of a user defined instrument parameter file. This enables the same command structure to be used with files recorded at different times. A list is kept of all the focusing runs and instrument parameter files, together with the dates and run numbers for which the various files are valid. Current copies of this may be obtained from John Wright.

4.2 Preliminary Notes

For these routines to work you must have a copy of GENIEINIT.COM extracted from TEB$DUA0:[TEBMGR]. You also must define the logical GENIEINIT in your LOGIN.COM in a part that is executed in both interactive and batch mode. It is best if you define the logical GENIEINIT to point to this file so that a copy does not have to be in your default directory. For example, if you use the GENIEINIT.COM in the TEBMGR directory, you should define the logical GENIEINIT using a statement as below:

```
$ DEFINE GENIEINIT TEB$DUA0:[TEBMGR]GENIEINIT.COM
```

To define the logicals and symbols described later, for use of these routines, you should include the execution of ISISDEFN.COM, found in INST$DISK:[TEBMGR.PREM.ANALYSIS.NEW] in your LOGIN.COM file in a part that is executed in both interactive and batch mode. An example command line is shown:

```
$ @ INST$DISK:[TEBMGR.PREM.ANALYSIS.NEW] ISISDEFN.COM
```

The logical UTILITYN will then point to this directory, where the analysis files are stored. To distinguish this implementation from the original single detector bank implementations, the letter N has been added to the symbols to access the command files and programs. For example, the symbol to activate the refinement routine, ISISREF, has become NISISREF. Note that these programs do not run on the Alpha VAXen at the moment.

You must wait for one submission of BINn, NSAVBIN or NISISREF to finish before starting or submitting the next one. See section 4.6 for an example of how to string together multiple executions of NISISREF.

There are a lot of files generated during this analysis process. Some management of excess files is done by the process, however, it is recommended that you to run these processes on your SCRATCH$DISK, where a larger default disc quota is available. If you want to keep the *.HIS files for future use, you should keep a copy in your USER$DISK area or on the data archives. See section 4.12 to find out more information on archiving and restoring data.

4.3 Tasks

4.3.1 Create focused data files

The first task is to bin or focus the data from each of the individual detector elements to the single spectrum. All the necessary preliminary work of computing the shifts so that peaks from a standard sample (silicon) of each element are made to lie upon one another has already been done. Also a sample (vanadium) has been run to get the incident spectrum shape for normalising the data. The binning routine uses this information to add the separate detectors together and to divide by the incident spectrum shape. In addition, it may subtract a no sample background. The user can also set a flag to indicate that the data should be rebinned to a logarithmic bin from something less than 0.001 to 0.001 (the usual default for ENGIN). Since there are two detector banks, the spectrum created for each bank is stored separately in the same data file. The convention selected is that left is determined by positioning oneself in the sample location while facing the source. This binning task is accomplished by using the command BINn where n is a number, depending on the version of the focusing routine in use BINn wants a directory in which to store files, and the run numbers of the *.RAW files to bin. The output is a file or files named <run#>.HIS in the supplied directory. The left detector bank is stored in block 1 of this file and the right detector bank is stored in block 2. See the GENIE manual and the example sessions for instructions on how to view the results.

Sometimes the user saves data at intermediate times during a single run. These saves of data produced <run#>.SAV files that the user may see also to examine. In this case, the utility NSAVBIN should be used to focus the data files. NSAVBIN works in a similar manner to BINn.COM with two modifications. For each run number, NSAVBIN looks for all the versions of<run#>.SAV and focuses each version. Due to the limitation in the refinement program of 6 characters for the file name, in place of the run number as the name of the focused *.HIS file, a coded number is used. This code consists of the last two digits of the run number and two digits to indicate the version number of the *.SAV file. For example, for run 4578 with four *.SAV files, four *.HIS files will be created. The names of these files will be 7801.HIS, 7802.HIS, 7803.HIS, and 7804.HIS. You need to be careful as this may lead to confusion if you have so many runs as to wrap around. N.B., The input file for NSAVBIN cannot be used with NISISREF.

4.3.2 Select regions to fit

The second task is to determine the region or regions of interest for fitting. One can select a wide range of time-of-flight (note: must be time-of-flight and not d-spacing ranges) or several bands of time-of-flight. The best way to select regions is to look at the focused data files (*.HIS). You can do this using GENIE. To enter GENIE, type GENIE. Now you want to read the .HIS file into a workspace n. You do this by executing the following command in GENIE:

```
READ W<n> <FILE SPECIFICATION> <block number>
```

For example:

```
>READ Wl 1234.HIS 1
>READ W2 1234.HIS 2
>READ/CLOSE
```

This reads block l(left detector) of the file 1234.HIS into workspace 1 and block 2 (right detector) into W2. You can read it into any workspace you wish. (The GENIEINIT.COM sets up 25 workspaces.) Using the display capability you can decide what region or regions you wish to fit. You can select regions with individual or multiple peaks. You should get region boundaries in time-of-flight, as that is what the automated analysis program expects. Write down the regions of interest or save cursor values and then exit GENIE. See GENIE manual for more information.

4.3.3 Instrument parameter file

You need to select an instrument parameter file. These are located in UTILITYN and have names similar to si-5122.iparm and si-5623.iparm after the silicon calibration run numbers. A cerium oxide standard has also been used, and files derived from these are called (for example) ce_6966.iparm. These files contain the instrumental constants as determined from calibration runs. The calibration constants are obtained by fitting the silicon data using the standard refinement suite, so that the peak shape parameters are also determined. See "Profile Analysis of Neutron Powder Diffraction Data at ISIS" for details on the standard refinement routines. The user can copy these files and replace the appropriate values and then refer to that instrument parameter file. An example of an instrument parameter file is given later.

4.3.4 Fit regions of interest

NB. Currently the number of fit regions is limited to 10 per invocation of NISISREF. This command file will request the run numbers, the banks 1 (left) and/or 2 (right), the directory holding the *.HIS files, the file specification of the instrument parameter file, the regions of interest, the space group, lattice parameter and title for each run. It is assumed that all runs have the same space group and lattice parameter (i.e.; they are on the same sample). This command file drives a batch file called BATCHREF.COM to actually do the refinements. The refinement uses programs and command files developed by Bill David for this purpose, using a Pawley refinement method which allows the sample to have a higher degree of texture than is normally found in powder samples. The output from these refinements is a lattice parameter even if only one diffraction peak is selected. There are option flags when inputting the regions of interest to select anisotropic width (option flag =1) or select anisotropic width plus anisotropic shift and zero point (option flag =2). A blank or any other number indicates that only through the isotropic widths will be refined.

At this point it becomes necessary to distinguish between the two banks of data. Thus the output files contain either an L or R to indicate Left or Right detector banks. This is carried through the rest of the fitting programs, except for temporary files.

The routines will fit all the runs with each of the entered regions of interest and then extract the lattice parameter from the *.OUT files and the (hkl) and intensity from the *.HKL files for each region of interest. Finally it will combine the information for all regions of interest for a given run number into a summary files. One (*.SUM) contains the hkl, the d-spacing and its error, and the intensity and its error. The other (*.LSUM) contains the lattice parameter values and the region of interest limits.

If an error occurs during the fitting procedure, most likely due to the low intensity of a peak, then an error file (*.ERR) is generated. You should look for the presence of these files so that suspect fit data can be eliminated from the final files. Currently there is no automatic method for removing suspect fit information from the files.

4.3.5 Merging run information

You must merge runs (even if there is only one run) prior to calculating strains.

In general with residual strain measurements you have several runs that are related - such as different depths into a sample. These runs can be merged into one file by executing NMERGERUNS. This command will ask for a directory, the runs to merge, and an output file name. All the *.SUM and *.LSUM files created by the refinement process for the named runs will be merged together, keeping the two banks separate. The peak-by-peak information is output into a *.PKS file and a *.PKSRT file. The latter is more interesting as it contains the information sorted by decreasing d-spacing. Similarly, the lattice constants are merged and sorted and output to a *.LATSRT file. The *.PKSRT and *.LATSRT files are tab separated so they can be more easily read into a spreadsheet (e.g. Excel) or graphics (Sigma Plot or KaleidaGraph) programs on a PC or Mac.

4.3.6 Calculating Strains

Two programs exist for calculating the strains either relative to a supplied lattice spacing or to one of the runs contained in the *.PKSRT or *.LATSRT files. They are called NSTRAIN, and NLSTRAIN. The strain can be calculated relative to the values of the lattice parameter from a "stress free" sample or to the values of the run contained within the input file. The output is *.STRAIN or *.LSTRAIN, respectively. These files are also tab separated.

4.4 Options

One does not have to run NISISBIN, NSAVBIN, or NISISREF interactively. NISISREF actually creates a file named RUN.INPUT in the selected directory. This file contains all the information needed to drive BINn or NISISREF. So you can edit this file and use it again and again and avoid typing in redundant information. See example for more information. This option can be used with NMERGERUNS; however the user must supply an output file name on the command line as the third parameter.

You can do everything interactively in GENIE by issuing the commands included in the command files. You can clean up the unwanted files created by the refinements by running NCLEANUP and supplying the directory and run number. Some purging and clean up is done along the way in the refinements, but the files deleted by NCLEANUP include those necessary to graphically look at the fits to the data.

As a final comment, copies of all the necessary files are stored in UTILITYN together with an ISISDEFN.COM file that will set up the logicals to point to this directory and symbols for the command files when run as part of your LOGIN.COM.

4.5 Useful Logicals and Symbols

Get a copy of GENIEINIT.COM from TEB::TEB$DUAO:[TEBMGR]. You must have this for the binning and refinements to work. You also need a copy of ISISDEFN.COM from UTILITYN to make the following definitions for ease of using these routines. You can either copy these to your user$disk, or use assignments as shown in section 4.2

```
$ DEFINE UTILITYN INST$DISK:[TEBMGR.PREM.ANALYSIS.NEW]
$ NISISBIN == "@UTILITYN:ISISBIN.COM"
$ BIN2 == "@UTILITYN:ISISBIN2.COM"
$ BIN3 == "@UTILITYN:BIN_10_94.COM"
$ BIN4 == "@UTILITYN:BIN_12_94.COM
$ NSAVBIN == "@UTILITYN:SAVBIN.COM"
$ NISISREF == "@UTILITYN:ISISREF.COM"
$ NMERGERUNS == "@UTILITYN:MERGERUNS"
$ NSTRAIN == "RUN UTILITYN:STRAIN"
$ NLSTRAIN == "RUN UTILITYN:LATSTRAIN"
$ NCLEANUP == "@UTILITYN:CLEAN-UP.COM"
$ NPRINTOUT == "RUN UTILITYN:PRINTOUT"
```

4.6 Example of Multiple NISISREF submission

If you wish to run several refinements with different input files, you can create a control command file similar to below. This command file can be run interactively - don't logout- or in a batch queue not used by NISISREF- NISISREF uses HUBWK2$FAST and SYS$SLOW generic queues.

```
$! Command file to run sequential invocations of NISISREF
$!
$ NISISREF FILE FIRST.INPUT              ! Do first input file
$!
$! Pause this command file until the BATCHREF job completes -- you
$! should check to which queue the BATCHREF job is submitted.
$!
$ SYNCHRONIZE /QUEUE=HUBWK2$FAST BATCHREF
$!
$! For this to work, you must put all three parameters on the NMERGERUNS
$! command line.
$!
$ NMERGERUNS FILE FIRST.INPUT FIRST
$!
$! Now on to the next input file
$!
$ NISISREF FILE SECOND.INPUT             !Do second input file
$!
$! Pause this command file until the BATCHREF job completes -- you
$! should check to which queue the BATCHREF job is submitted.
$!
$ SYNCHRONIZE /QUEUE=HUBWK2$FAST BATCHREF
$!
$! For this to work, you must put all three parameters on the NMERGERUNS
$! command line.
$!
$ NMERGERUNS FILE SECOND.INPUT SECOND
$!
```

Repeat this pattern until all input files have run. Be careful about disc space!

4.7 Example of an Instrument Parameter File

This is the instrument parameter file located in UTILITYN called SI_5623.IPARM. The first line reading LEFT indicates that the parameters that follow belong to the left detector bank and similarly for the line RIGHT. All these lines must be present and no additional lines can be present.

```
LEFT
L ZERO    0.2596
L PKCN    15.8149     0.0000
L PKFN TAUF 8.0000    0.0000      4.6745
```

```
L PKFN TAUS 8.0000   28.04597    2.8644
L PKFN SIGM 5.0000   0.0000      50.3749    0.0000
L PKFN GAMM 10.0000              0.0000     3.2789     0.0000
L PKFN SWCH 1.0000   4.4961      0.7223
RIGHT
L ZERO    0.095
L PKCN    15-8089    0.0000
L PKFN TAUF 8.0000   0.0000      4.9892
L PKFN TAUS 8.0000   28.0133     2.8644
L PKFN SIGM 5.0000   0.0000      50.8798    0.0000
L PKFN GAMM 10.0000              0.0000     4.5563     0.0000
L PKFN SWCH 1.0000   4.6117      .7154
```

4.8 Example Session

These examples assume you have executed ISISDEFN.COM.

4.8.1 Example NISISBIN session

```
$ NISISBIN

Enter directory to store binned data file(s).
(Ex: [jag.mit]): scratch$disk:[jag]

Enter the run number(s) (slash separated) of the data to be binned
(Ex: 1218/1219/1220) : 5490/5491/5492/5493/5494

Do you want to change the bin grouping to [0.001]?
(1=YES, anything else including <CR> means no): 1

The binned file(s) will be saved as scratch$disk:[jag]run#.his

Job BIN (queue SYS$SLOW, entry 24) started on ISVS1$SLOW

Job BIN (queue SYS$SLOW, entry 24) completed

$ dir scratch$disk:[jag]*.his

Directory SCRATCH$DISK:[JAG]

5490.HIS;l    5491.HIS;l    5492.HIS;1    5493HIS;l    5494.HIS;l

Total of 5 files.

$ NISISREF

Enter directory of stored binned data file(s).
(Ex: [jag.mit]): scratch$disk:[jag]
```

Enter the file specification of the instrument constants
(Ex:user$disk:[jag.utility.new]si-5623.iparm):user$disk:[jag.utility.new]si-5623.iparm

Enter the run number and banks (space separated) of the data to be refined.
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5490 1 2

Enter the run number and banks (space separated) of the data to be refined.
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5491 1 2

Enter the run number and banks (space separated) of the data to be refined.
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5492 1 2

Enter the run number and banks (space separated) of the data to be refined.
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5493 1 2

Enter the run number and banks (space separated) of the data to be refined.
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5494 1 2

Enter the run number and banks (space separated) of the data to be refined.
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
<CR>

Enter min and max times (space separated) for range 0
and refinement flag (0 or blank = do through isotropic widths,
1=do anisotropic widths, 2=do zero and anisotropic shift)
(e.g.,4000 14000 1), <CR> to end
Data: **3000 12250 2**

Enter min and max times (space separated) for range 1
and refinement flag (0 or blank = do through isotropic widths,
1=do anisotropic widths, 2=do zero and anisotropic shift)
(e.g.,4000 14000 1), <CR> to end
Data: **6000 12250 1**

Enter min and max times (space separated) for range 2
and refinement flag (0 or blank = do through isotropic widths,
1=do anisotropic widths, 2=do zero and anisotropic shift)
(e.g.,4000 14000 1), <CR> to end
Data: **10500 12500**

Enter min and max times (space separated) for range 0
and refinement flag (0 or blank = do through isotropic widths,
1=do anisotropic widths, 2=do zero and anisotropic shift)

```
(e.g.,4000 14000 1), <CR> to end
Data: <CR>

Enter lattice space group (only cubic elements)
with spaces between symbols; e.g., F M 3 M
Data: i m 3 m
Enter lattice parameter (only cubic elements)
Data: 2.86
Enter title for run 5490
Data: Horz r=9, h=3
Enter title for run 5491
Data: Horz r=9, h=12
Enter title for run 5492
Data: Horz r=9, h=15
Enter title for run 5493
Data: Horz r=9, h=21
Enter title for run 5494
Data: Horz r=9, h=27
A summary of the fits for each run will be saved as
SCRATCH$DISK:[JAG]run#<L or R>.sum

Job BATCHREF (queue HUBWK2$SLOW, entry 32) started on
HUBWK2$SLOW

Job BATCHREF (queue HUBWK2$SLOW, entry 32) completed

$ DIR SCRATCH$DISK:[JAG]*.SUM,*.LSUM

Directory              SCRATCH$DISK:[JAG]

5490L.SUM;l       5490L0.SUM;l       5490Ll.SUM;l       5490L2.SUM;l
5490R.SUM;l       5490R0.SUM;l       5490Rl.SUM;l       5490R2.SUM;l
5491L.SUM;1       5491L0.SUM;1       5491Ll.SUM;l       5491L2.SUM;1
5491R.SUM;1       5491R0.SUM;1       5491Rl.SUM;l       5491R2.SUM;1
5492L.SUM;l       5492L0.SUM;l       5492Ll.SUM;l       5492L2.SUM;l
5492R.SUM;l       5492R0.SUM;l       5492Rl.SUM;l       5492R2.SUM;l
5493L.SUM;l       5493L0.SUM;l       5493Ll.SUM;l       5493L2.SUM;l
5493R.SUM;l       5493R0.SUM;l       5493Rl.SUM;l       5493R2.SUM;l
5494L.SUM;l       5494L0.SUM;l       5494Ll.SUM;l        5494L2.SUM;l
5494R.SUM;l       5494R0.SUM;l       5494Rl.SUM;l       5494R2.SUM;l
5490L.LSUM;l      5490L0.LSUM;l      5490Ll.LSUM;l      5490L2.LSUM;l
5490R.LSUM;l      5490R0.LSUM;l      5490Rl.LSUM;l      5490R2.LSUM;l
5491L.LSUM;1      5491L0.LSUM;1      5491Ll.LSUM;l      5491L2.LSUM;1
5491R.LSUM;1      5491R0.LSUM;1      5491R1.LSUM;1      5491R2.LSUM;1
5492L.LSUM;l      5492L0.LSUM;l      5492Ll.LSUM;l      5492L2.LSUM;l
5492R.LSUM;l      5492R0.LSUM;l      5492Rl.LSUM;l      5492R2.LSUM;l
5493L.LSUM;l      5493L0.LSUM;l      5493Ll.LSUM;l      5493L2.LSUM;l
5493R.LSUM;l      5493R0.LSUM;l      5493Rl.LSUM;l      5493R2.LSUM;l
5494L.LSUM;l      5494L0.LSUM;l      5494Ll.LSUM;l      5494L2.LSUM;l
```

```
5494R.LSUM;l        5494R0.LSUM;l        5494Rl.LSUM;l        5494R2.LSUM;l
```

```
$ NMERGERUNS
Enter the run number and banks (space separated) of the fits to be merged
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5490 1 2

Enter the run number and banks (space separated) of the fits to be merged
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5491 1 2

Enter the run number and banks (space separated) of the fits to be merged
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5492 1 2

Enter the run number and banks (space separated) of the fits to be merged
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5493 1 2

Enter the run number and banks (space separated) of the fits to be merged
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
5494 1 2

Enter the run number and banks (space separated) of the fits to be merged
(Ex: 5333 1 2: means run# 5333 and banks l(left) and 2(right)), <CR> to end
<CR>

Enter directory to where files are located.

(Ex: jag.mit]): SCRATCH$DISK:[JAG]

Enter filename only of the output file (no directory/extension specifications).
(Ex: allruns): POWDER_HORZ_

The merged run file by hkl is saved as SCRATCH$DISK:[JAG]POWDER-HORZ-L.pks
The merged run file by regions is saved as SCRATCH$DISK:[JAG]POWDER-HORZ-L.lats
The hkl-sorted merged runs saved as SCRATCH$DISK:[JAG]POWDER-HORZ-L.pksrt
The region-sorted merged runs saved as SCRATCH$DISK:[JAG]POWDER-HORZ-L.latsrt
The merged run file by hkl is saved as SCRATCH$DISK:[JAG]POWDER-HORZ-R.pks
The merged run file by regions is saved as SCRATCH$DISK:[JAG]POWDER-HORZ-R.lats
The hkl-sorted merged runs saved as SCRATCH$DISK:[JAG]POWDER-HORZ-R.pksrt
The region-sorted merged runs saved as SCRATCH$DISK:[JAG]POWDER-HORZ-R.latsrt
```

```
$ NSTRAIN
Enter the name of the *.PKSRT file for which strains are to be computed
(.PKSRT ASSUMED):
SCRATCH$DISK:JAG]POWDER-HORZ-L
Analyse by run(R) comparison or stress-free sample(S)? (R or S):
S
```

```
Enter the stress-free lattice parameter value/error
2.85909 .00036

Output file is: SCRATCH$DISK:[JAG]POWDER-HORZ-L.strain

$ NSTRAIN

Enter the name of the *.PKSRT file for which strains are to be computed
(.PKSRT ASSUMED):
SCRATCH$DISK:[JAG]POWDER-HORZ-R
Analyse by run(R) comparison or stress-free sample(S)? (R or S):
R
Enter the run number for strain comparison:
5490
Output file is: SCRATCH$DISK:[JAG]POWDER-HORZ-L.strain

$ NLSTRAIN
Enter the name of the *.LATSRT file for which strains are to be computed
(.LATSRT ASSUMED):
SCRATCH$DISK:[JAG]POWDER-HORZ-L
Analyse by run(R) comparison or stress-free sample(S)? (R or S):
S
Enter the stress-free lattice parameter value/error
2.85909 .00036
Output file is: SCRATCH$DISK:[JAG]POWDER-HORZ-L.strain

$ NLSTRAIN
Enter the name of the *.LATSRT file for which strains are to be computed
(.LATSRT ASSUMED):
SCRATCH$DISK:[JAG]POWDER-HORZ-L
Analyse by run(R) comparison or stress-free sample(S)? (R or S):
R
Enter the run number for strain comparison:
5490
Output file is: SCRATCH$DISK:[JAG]POWDER-HORZ-L.strain
```

4.8.2 Example of Using NISISBIN and NISISREF with an Input File

```
$ NISISBIN file SCRATCH$DISK:[JAG]run.input
The binned file(s) will be saved as SCRATCH$DISK:[JAG]run#.his

Job BIN (queue SYS$SLOW, entry 855) started on ISVS1$SLOW

$ NISISREF file SCRATCH$DISK:[JAG]run.input 1

%DCL-W-UNDFIL, file has not been opened by DCL - check logical name

A summary of the fits of each run will be saved as
SCRATCH$DISK:[JAG]run#<Lor R>.sum
```

```
Job BATCHREF (queue HUBWK2$SLOW, entry 859) started on
HUBWK2$SLOW
```

```
$ NMERGERUNS file SCRATCH$DISK:[JAG]run.input allruns
```

At this point go back to calculation of strains using STRAIN and/or LSTRAIN and continue.

4.8.3 Example Input File For Above Mode of Operation

```
SCRATCH$DISK:[JAG]
USER$DISK:[JAG.UTILITY.NEW]SI-5623.IPARM
5490 1 2
5491 1 2
5492 1 2
5493 1 2
5494 1 2
                              ! Blank line to separate run numbers from time regions
3000 12250  2
6000 12250  1
10500   12500
                              ! Blank line to separate time regions from rest of input
I M 3 M
2.86
HORZ R=9 H=3
HORZ R=9 H=9
HORZ R=9 H=1 5
HORZ R=9 H=21
HORZ R=9 H=27
```

4.9 Output Files of Interest

The main file of interest is the summary file produced at the end of the automatic refinement routine NMERGERUNS and files with the strains values output from either NSTRAIN or NLSTRAIN. These files are *.PKSRT, *.LATSRT, *.STRAIN, and *.LSTRAIN. The columns are tab separated to make it easier for insertion in a spread sheet or graphics program on a PC or Mac

Since the amount of information now stored in these files is greatly increased, it no longer fits easily on a single sheet of paper, so an example of the original file is not given here. The organisation of the file is similar to the original output files, with the following additional columns:

Expanded *.PKSRT File:

RUN#	the run number
CODE	refinement option code, 0,1 or 2
H	h Miller index

K	k Miller index
L	l Miller index
HKL2	h^2+k^2+I^2
D-SPACE	d-spacing value
D-ERR	error in the d-spacing value
INTENSITY	intensity of the peak
I-ERR	error in the intensity value
SIGMA2	Gaussian isotropic width
S2-ERR	error in Gaussian isotropic width
GAMMA2	Lorentzian isotropic width
G2-ERR	error in Lorentzian isotropic width
GAMMA1	anisotropic peak width
GL-ERR	error in anisotropic peak width
PKCN2	anisotropic shift
P2-ERR	error in anisotropic shift
ZERO	zero offset
Z-ERR	error in the zero offset

Currently the d-spacing is not correct by the anisotropic shift parameter nor is the GAMMA 1 anisotropic width adjusted for the reflection. However, the isotropic peak width are given for the d-spacing -- that is the sigma-2 value from the refinement is adjusted by (d-spacing)A2 and the gamma-2 is adjusted by the d-spacing.

The *.STRAIN is nearly identical to the *.PKSRT file, the only difference being that the USTRAIN (microstrain) and S-ERR (error in the microstrain) replace the D-SPACE and D-ERR values and column headings.

Expanded *.LATSRT File:

RUN#	the run number
CODE	refinement option code, 0,l or 2
TMIN	time minimum of the fitted region
TMAX	time maximum of the fitted region
LATTICE	lattice parameter value
L-ERR	error in the lattice parameter value
SIGMA2	Gaussian isotropic width
S2-ERR	error in Gaussian isotropic width
GAMMA2	Lorentzian isotropic width
S2-ERR	error in Lorentzian isotropic width
GAMMA1	anisotropic peak width
GL-ERR	error in anisotropic peak width
PKCN2	anisotropic shift
P2-ERR	error in anisotropic shift
ZERO	zero offset
Z-ERR	error in the zero offset

The *.LSTRAIN file is nearly identical to the *.PKSRT file, the only difference being that the USTRAIN (microstrain) and S-ERR (error in the microstrain) replace the LATTICE and L-ERR values and column headings.

4.10 Printing or Looking at Data Files

You can look at these files in the VMS EVE editor. However they extend beyond the 132 character screen. This is due to the tabs in the file, whose lines are only 132 characters long. You can use the SHIFT RIGHT nn and SHIFT LEFT nn to see all the file.

To get an 80 column printout of this information, execute NPRINTOUT. This wraps around the information to fit in 80 columns. Below are short examples of the output of this program. You have the option of sending the file to the printer directly to a printer or creating a new file in 80 column format.

Example of the output of PRINTOUT for the PKSRT files:

```
RUN# CODE H  K  L HKL2   D_SPACE    D_ERR  SIGMA2     S2_ERR  GAMMA1     GL_ERR
                         INTENSITY I_ERR   GAMMA2     G2_ERR  PKCN2      P2_ERR
                                                              ZERO       Z_ERR
5489   0    1  1  0  2   2.02657    0.00013 7.12E+02  1.7E+02 0.00E+00   0.0E+00
                         1241.60    23.00   7.50E+00  3.4E+00 0.00E+00   0.0E+00
                                                              0.00E+00   0.0E+00
5490   0    1  1  0  2   2.02663    0.00022 8.42E+02  3.3E+02 0.00E+00   0.0E+00
                         1027.70    42.40  -6.23E+00  6.9E+00 0.00E+00   0.0E+00
                                                              0.00E+00   0.0E+00
5491   0    1  1  0  2   2.02653    0.0001  9 6.68E+02 2.5E+02 0.00E+00  0.0E+00
                         1439.90    45.00   3.71 E+00 4.5E+00 0.00E+00   0.0E+00
                                                              0.00E+00   0.0E+00
```

Example of the output of PRINTOUT for the LATSRT files:

```
RUN#   CODE  TMIN   TMAX  LATTICE  L_ERR   SIGMA2    S2_ERR   GAMMA1   GL_ERR
                                           GAMMA2    G2_ERR   PKCN2    P2_ERR
                                                              ZERO     Z_ERR
5489   0     3000   12500 2.86605  0.00011 3.65E+01  2.9E+00  0.00E+00 0.0E+00
                                           2.64E+00  4.0E-01  0.00E+00 0.0E+00
                                                              0.00E+00 0.0E+00
5490   0     3000   12500 2.86581  0.00016 3.06E+01  4.2E+00  0.00E+00 0.0E+00
                                           1.56E+00  6.3E-01  0.00E+00 0.0E+00
                                                              0.00E+00 0.0E+00
```

This output stacks the information about a fit. So for the PKSRT file, the d-spacing and intensity are in the same column but adjacent rows. So he d-spacing and its error for run 5490 are 2.02657 and 0.00013, respectively and the intensity and its error are 1241.60 and 23.00, respectively. And so forth for the widths and anisotropic shift.

4.11 Displaying Refined Data in GENIE

Several routines exist to display the .PIC and .TIC files created in the refinement. If you have defined GENIEINIT as explained above, then these routines are defined as GENIE symbols for you. PLS and PLSD are the first routines to use. These display the observed data points as a histogram, the fitted curve produced by the refinement, plus in a separate graph below the main one, the difference between the two. PLS displays in time-of-flight units, while PLSD displays in d-spacing. The data file name you will be asked for is of the form 5677L0, or 5677R0, where the last character is the number of the refinement (0-9) as defined in RUN.INPUT. PLC will plot the same data file using different ranges of time-of-flight or d-spacing. In addition, typing TIC will display the positions of the Bragg reflections as determined by the refinement. Typing TIC alone autoranges the positions of these lines. You can also type TIC n1 n2 where n1 and n2 are appropriate numbers within the range of the vertical axis to display these lines where you want them.

4.12 Archiving and Restoring Data

Data is archived on the optical disk - see PUNCH manual - normally within about 10 minutes of the end of a run. This means that data is normally deleted by the instrument scientists at regular intervals to make space for new data files. Full details of the archiving system can be obtained by typing:

```
$ help optical
```

at the VMS prompt. Data may be restored to your scratch$disk area using two principal methods.

4.12.1 Retrieve.com

This method is used for retrieving small numbers of files to your scratch$disk area. You should copy the file retrieve.com from user$disk:[jsw.progs], then edit it to point at your own scratch$disk area, changing the line:

```
$ p3=" /out=scratch$disk:[jsw]"+p1
```

to

```
$ p3=" /out=scratch$disk:[ano01]"+p1
```

where ano01 is your user name. When this command file is run, it will ask you for the number of the file and the catalogue number. The dialogue will look something like this:

```
$ @retrieve
Filename: 7173
Catalogue number: 943
SCRATCH$DISK:[ANO01]TEB07173.RAW;1
  -- End of restore --
```

The catalogue number is the number of the cycle that the run was measured in, this can be found on the list of focusing runs and instrument parameter files mentioned in section 4.1.

The invocation of this command file can be shortened from @retrieve to ret by including the line:

```
$  RET*RIEVE:== @USER$DISK:[ANO01]RETRIEVE.COM
```

in your login.com.

4.12.2 Listodsr

This second method can be used for the restoration of a number of files. Generally it will only allow you to restore files to your scratch$disk area. The most useful mode of this command is to use a command line as follows:

```
$ listodsr/ed/cat=inst$disk:[tebmgr]catalogue.943
```

Here the extension of the catalogue file is the same as the cycle number. Keying PF2 on the numeric keypad will bring up a display of available commands. DO NOT use ctrl-D, or your life may be in danger! To use the refinement programs with files restored to your scratch$disk, Copy the genieinit.com you are using to your user$disk area. Edit this so that the appropriate lines look like:

```
set disk scratch$disk
set dir [ano01]
```

Then edit your login.com, making sure that 'genieinit' points to your user$disk. The binning routine will then automatically look for your scratch$disk.

4.12.3 Archiving Files

It is recommended that you archive focused data files as .HIS files, so that they are available for future analysis. This also avoids having to re-focus files after the seven day limit on files in the scratch$disk area causes them to be deleted. Assuming that they are resident in your scratch$disk area and that this is also your default directory, this can be done by typing the following:

```
$ archodsr/cat=user$disk:[ano01]catalogue.dat filename.his
```

Wildcards may be used to archive groups of files. A message will appear on the screen to acknowledge that each file has been flagged for archiving. A file called catalogue.dat will now appear in your user$disk area, which you should not delete! Files can retrieved to your scratch$disk area using the listodsr command as above, except that /cat=user$disk:[ano01]catalogue.dat is used in the same way as when you archive files.

5. Eating and Drinking

5.1 R55

There are two coffee machines in R55, one by the μon beam line with a chocolate machine next to it, and one behind the data acquisition centre, where is also a cold drink machine, a drinking water cooler, two snack machines and a microwave oven. Beware that the coffee machines will not accept the debased magnetic 1 and 2 pence pieces at the time of writing.

5.2 Other Facilities in RAL

There is another coffee machine and snack machine on the ground floor of R2. There are also coffee lounges adjacent to the library in R61 and adjacent to the restaurant in R22.These are open, **Monday to Friday only**, at times listed below:

R61	9.30 a.m.	11.30 a.m.
	12.30p.m.	15.30 p.m.
R22	12.45p.m.	13.45 p.m.

5.3 RAL Restaurant

The RAL restaurant is open for breakfast lunch and evening meals 7 days a week at times listed below:

	Monday - Friday	Saturday - Sunday
Breakfast	7.30 a.m. - 8.30 a.m.	8.00 a.m. - 9.00 a.m.
Lunch	11.45 a.m. - 1.45 p.m.	12.00 p.m. - 1.00 p.m.
Evening Meal	5.15 p.m. - 7.15 p.m.	6.00 p.m. - 7.00 p.m.

5.4 Local Pubs

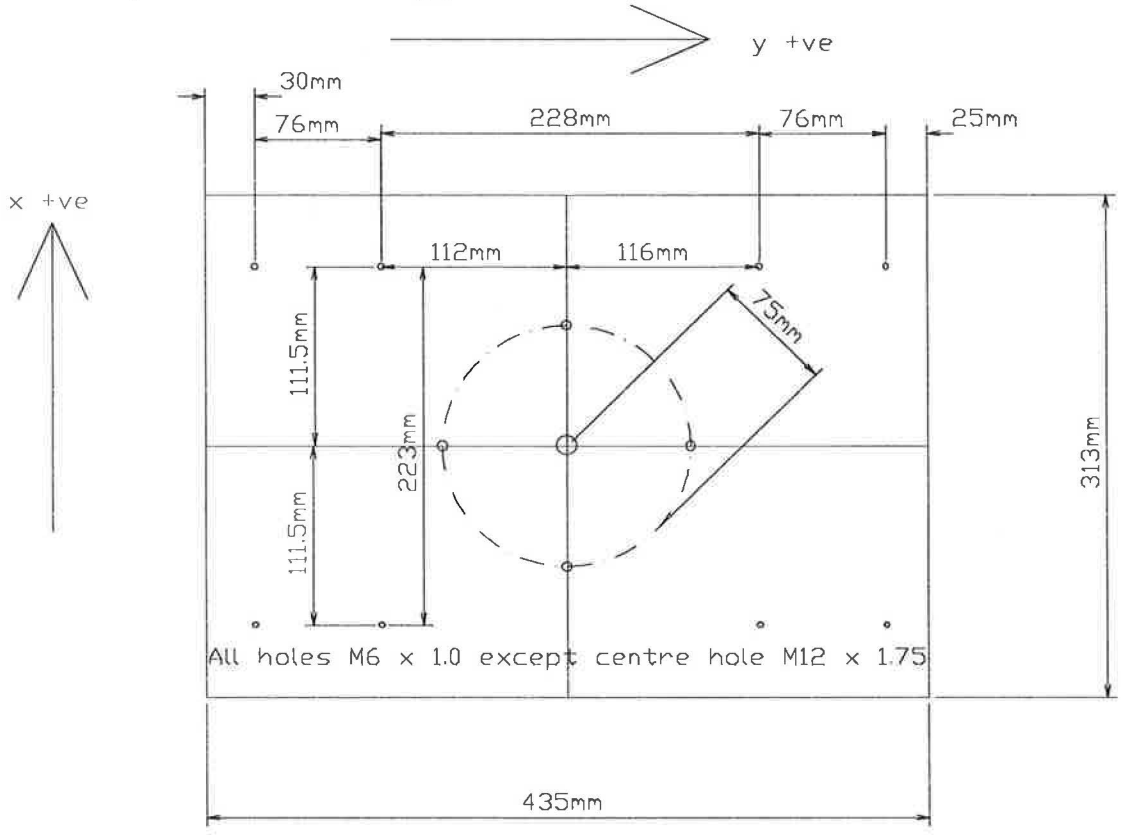
Most of the local pubs serve food - though some do not on Sundays!

Blewbury	The Red Lion
Chilton	Rose and Crown
East Hendred	The Plough
	The Wheatsheaf
East Ilsley	The Crown and Horns
	The Swan
Harwell	The White Hart
Steventon	The Cherry Tree
Wantage	The Lamb
	The Swan

West Hendred	The Hare
West Ilsley	The Harrow

Appendix

ENGIN positioner mounting plate



Index