

PORTING SCIENTIFIC SOFTWARE TO WINDOWS (OR NOT) ?

NOBUGS October 2004

Andy Hammersley

ESRF Grenoble France

(An Aside)

- Software engineering is about GOOD DESIGN (which meets the requirements)
- If the design is good new features can be added easily within the existing framework
- If the design is bad adding new features will “bend” and “break” the framework
- Every framework has its limits
- Unfortunately good design is not simple

OVERVIEW

(Relevant to all User Controlled Programs)

- The Rise of Windows (and alternatives)
- What is a ported Windows application
- Porting experience
- Partial porting and alternative approaches
- Conclusion

WHY CONSIDER WINDOWS ?

(~10 Years ago we didn't)

1. PC HARDWARE

- Extremely Cheap (Remember workstation prices)
- Powerful (Only High end systems such as the Alpha and G5 significantly more)
- Versatile (Reasonably configurable)
- Convenient (Buy at Supermarket)
- Modular
- Portable (Even wearable !)
- Universal
- Becoming Scalable (64 bit, multi-core, Blade systems)

WHY CONSIDER WINDOWS ?

2. WINDOWS OPERATING SYSTEMS

- User Friendly (well much more than Unix)
- Enormous range of user software; both commercial and free
- Universally available
- Little System Administration
- Native on Latest Hardware e.g. Lap-tops
- Very high degree of compatibility between versions

IS LINUX GOOD ENOUGH ON THE DESKTOP ?

- “Not for my grandfather” (or IBM)
- Are dual boot machines a sufficient answer ?
- When will there be a single agreed Linux window manager style ?
- What is happening with commercial Linux ?
- (Can Bill Gates and others use IP’s to kill free software ?)

SCIENTIFIC SOFTWARE HISTORY

- Some dates back to mainframe era
- Much developed on VAX/VMS
- Ported to many different Unix systems
- Graphics developed using X-11 Window System
- Often ported to Linux
- Many different types of graphical and command line interfaces

Keyboard and Graphical User Interfaces

```

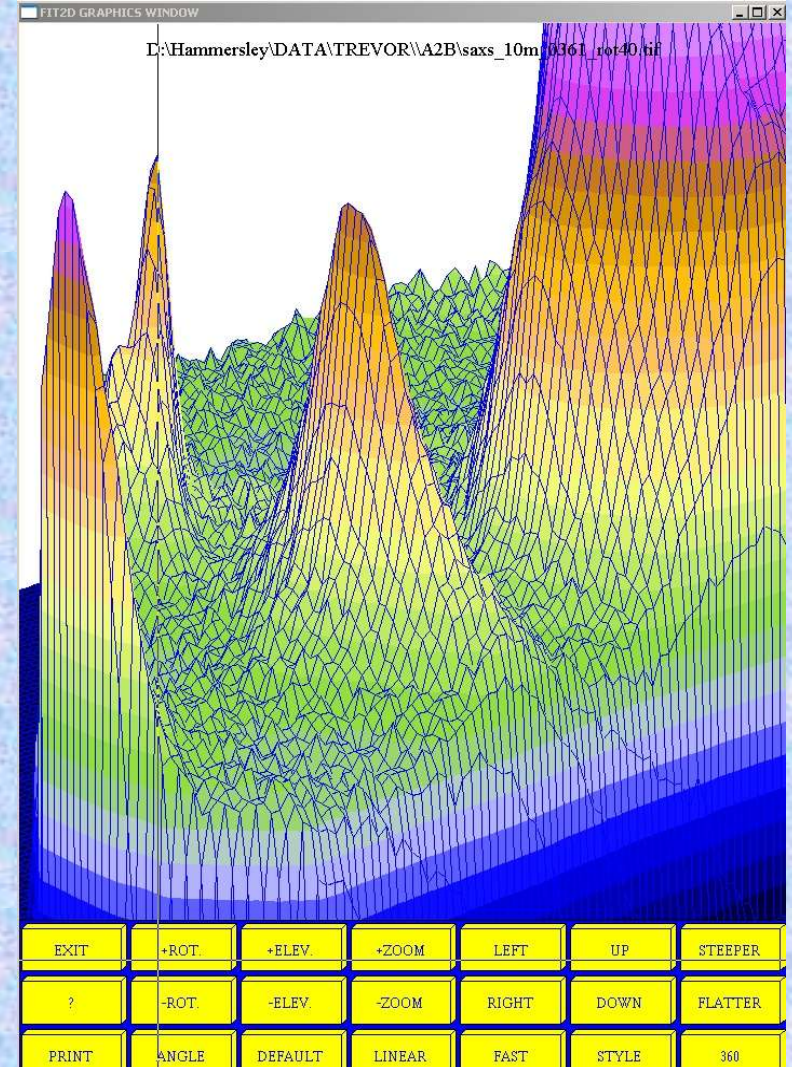
ex Fit2d
PAUSE: Wait for a user return (including within macros)
PEEP: Look at pixel coordinates and values
PIXEL REGION: Change region of interest using pixel limits
PLOT DATA: Plot data as 2-D image (or X-Y graph if 1-D)
POISSONIAN NOISE: Add Poissonian noise to data
POLARISATION EFFECT: Apply polarisation effect to intensities
POSTSCRIPT OPTIONS: Controls options affecting PostScript
POWER SPECTRUM: Calculate power spectrum of ROI
PREDICTOR: Applied a predictor algorithm to image
PRINT GRAPHICS: Output graphics to a file for printing
PUBLICATION QUALITY: Set attributes for high quality output
QUESTION: Ask an interactive question during a macro
QUIT: Exit from program
RAISE TO A POWER: Raise elements in ROI to specified power
--60z--(This is NOT more)--- <RETURN> = Next "page" --- ? = help -----:

REBIN: Rebin data, output in memory (same as "RE-BIN")
RE-BIN: Re-bin data, output in memory
RECALL: Recall data set from internal memory
REFLECT: Reflect data about input line, output in memory
REGION: Change region of interest
RING (ADD POWDER RING): Add powder diffraction ring to data
ROI: Defined Region Of Interest (Region Of Interest) (pixels)
ROTATE LUT: Interactive rotation of the colour table
RUN MACRO: Run previously saved macro definition file
SELECT PIXEL OPERATION: Operation on defined pixel value range
SEQUENCE: Run macro for a sequence of files (or not)
SET ANNOTATION STYLE: Set style of annotation label text
SET ARROW STYLE: Set style of an arrow, style, colour, etc.
SET AXES STYLE: Set style of axes, line width colour
SET BACKGROUND STYLE: Define background colour
SET COLOURS: Set colour of graph lines, text, and markers
--72z--(This is NOT more)--- <RETURN> = Next "page" --- ? = help -----:

SET CURVE STYLES: Set attributes for curve representations
SET ENUMERATION STYLE: Set style for axis numbering
SET FONT: Select text font to be used for all roman text
SET GRID STYLE: line type, colour, and width for grid
SET LAYOUT STYLE: Graphics title, and axis label distances
SET PIXEL VALUE: Set value of defined pixel
SET TICK POSITIONS: Positions of axes large tick marks
SET TITLE STYLE: text font, colour, character width, etc.
SET X-LABEL STYLE: text font, colour, character width, etc.
SET Y-LABEL STYLE: text font, colour, character width, etc.
SLEEP: Pause for a defined number of seconds
SMOOTH: Top-hat convolution smoothing of user input size
SPATIAL FILTERING: Filtering in Spatial domain
START MACRO: Save commands in macro definition file
STATISTICS: Calculate parameters of a region of the data
STOP MACRO: Close previously opened macro definition file
--84z--(This is NOT more)--- <RETURN> = Next "page" --- ? = help -----:

STORE: Store present data set in internal memory
SUBTRACT: Subtract region of interest of memory from data
SURFACE INTERPOLATION: User defined surface in memory
SYMBOL: Same as "VARIABLE"
SYMMETRIC FUNCTION: Add circularly symmetric function to data
TITLE: Input new text for title
THRESHOLD: Set minimum and/or maximum values in ROI
TRANSPOSE: Transpose data arrays and variance arrays
UN-DEFINE VARIABLE: Remove variable from translation table
UNIT CELL PARAMETERS: Convert between real and reciprocal
VARIABLE: Define a named variable its type and value
VARIANCES DEFINITION: Define variances from current data
V2C: Variance to current and vice versa (special command)
WAIT: Wait for a user return (including within macros)
WEIGHTED AVERAGE: data and memory weighted by variances
X-AXIS LABEL: Enter X-axis for graphics
--96z--(This is NOT more)--- <RETURN> = Next "page" --- ? = help -----:

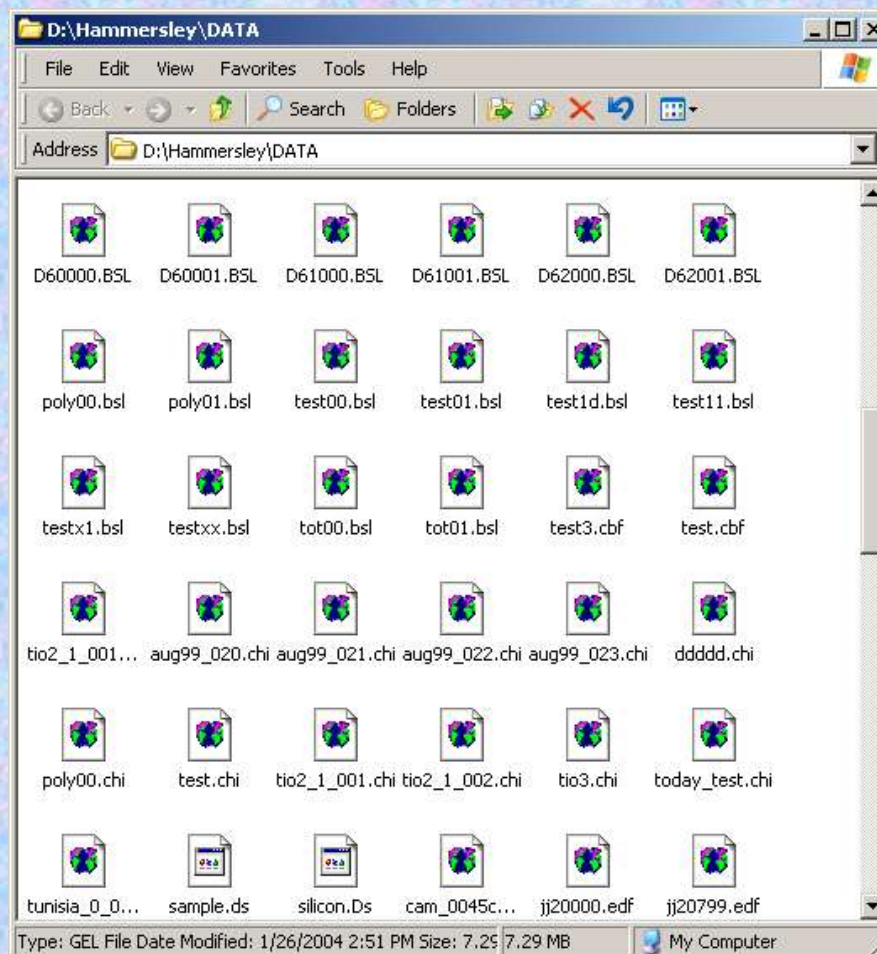
Y-AXIS LABEL: Enter Y-axis label for graphics
Z-AXIS LABEL: Enter Z-axis label (intensity) for graphics
Z-SCALE: Image scaling mode, automatic, minimum/maximum
1-D INTERPOLATION: Correct 1-D values by interpolation
3-D SURFACE PLOT: Graphics view of region of interest
Main menu: ENTER COMMAND [HELP]:input
  
```



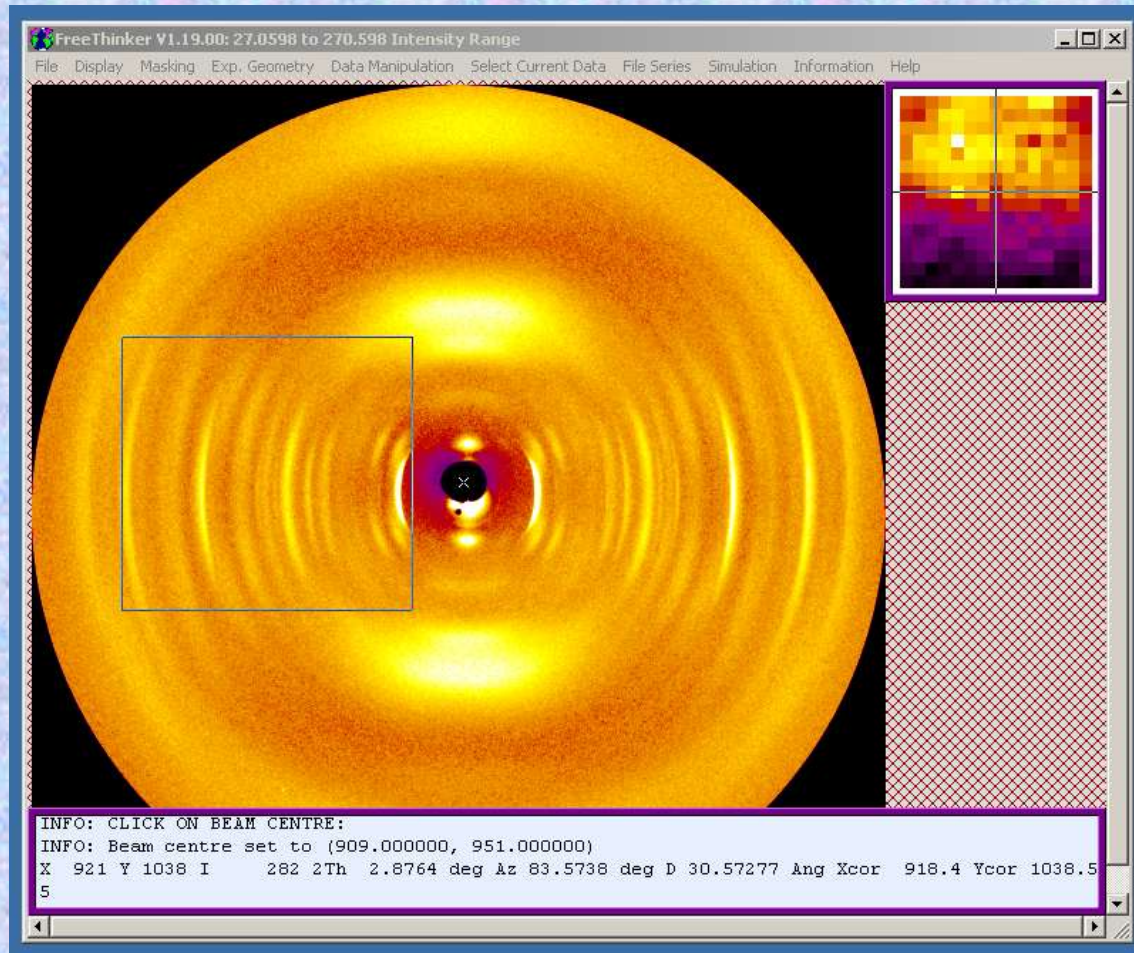
WHAT ARE THE CHARACTERISTICS OF A WINDOWS APPLICATION ?

- Associate program with file type(s)
- Ability to drop files onto program icon
- Standard menu bar options (plus others)
- Standard “File” menu options (plus others)
- Standard Short-cut keys (e.g. Print: CTRL-P)
- “Click and Drag” interaction

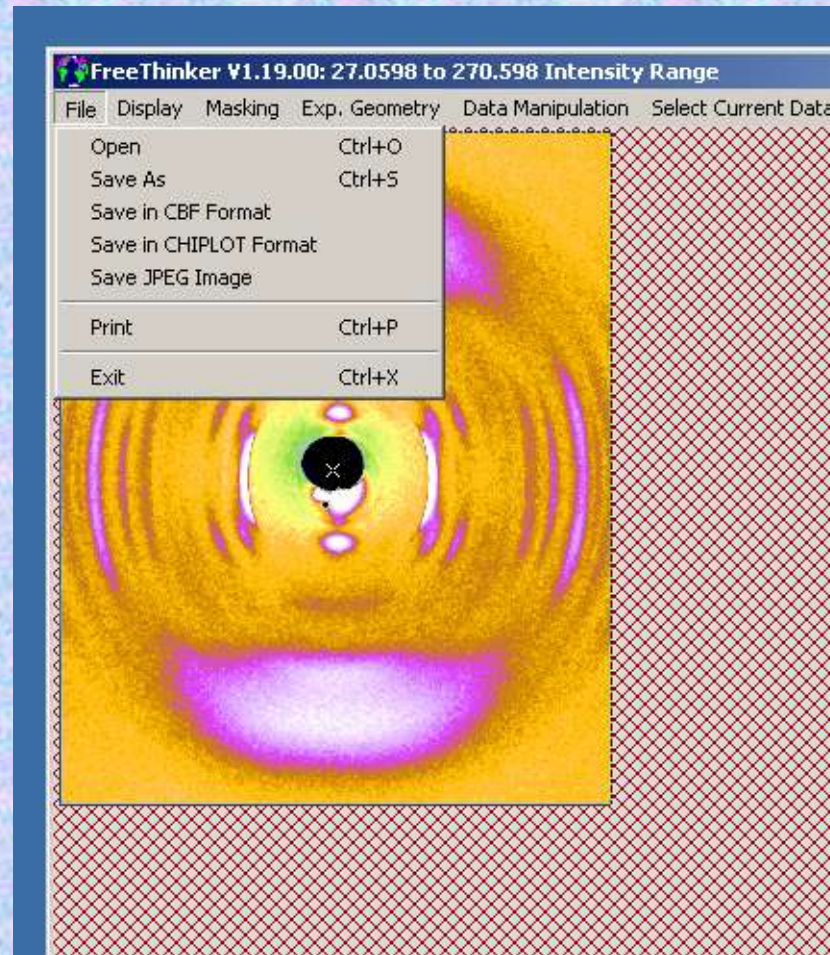
DATA FILES ASSOCIATED WITH FreeThinker



WINDOWS APPLICATION MENU BAR AND DISPLAY



WINDOWS “FILE” MENU



GOOD POINTS OF WINDOWS API

- Consistency
- Documentation (Generally)
- Menu-bar and menus
- Bitmap hardware display independence
- Rotated text
- Form / Dialogue editor and wizards
- Fortran 90 better on Windows than many sub-standard Unix products

BAD POINTS OF WINDOWS API

- Where's Posix (.1) Support ?
- Creating your own device independent bitmap and manipulating it is not trivial and only documented on the web
- Rotated text may not appear as requested, and without an error message

PARTIAL PORTING OF APPLICATIONS

- i.e. Get program running on Windows, but with the user interface as before e.g. FIT2D
- Much easier / faster approach
- Cygnus, etc. provide useful tools, but with drawbacks
- Command Window / Gnuplot approach

ALTERNATIVE APPROACHES ?

- Scripting languages: TOO SLOW, compatibility and support problems
- Java: Too slow, at best a hybrid solution would be necessary with its inherent problems
- High level platform independent component set (a pipe dream)

CONCLUSIONS

- (Porting) Applications to Windows will be more and more demanded
- Partial porting is possible and relatively easy, but results in a Unix application running on Windows
- Full porting may require complete re-write of applications and is a lot of work.
- Well written modular code can be re-used, but a lot of re-arrangement is necessary