A survival guide to Radiance on Windows

When you have (almost) no choice :-(

Why am I doing this?

- OS Monopoly
  - PCs come with Windows pre-installed
  - Corporate power (somebody else has decided for you)
- Integration with other Windows programmes (Ecotect, IES, etc.)
- Self inflicted pain

Are there no alternatives?

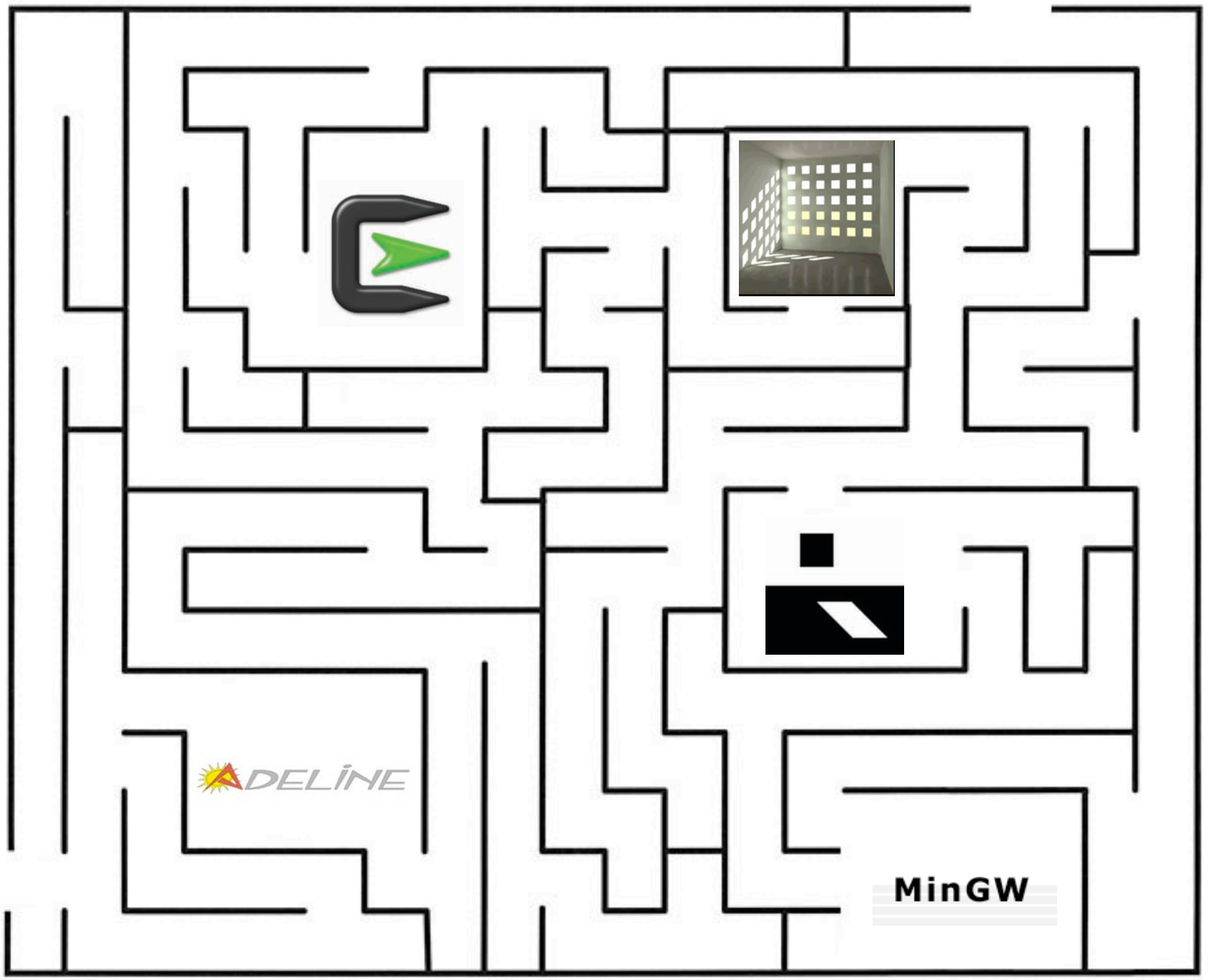
- Linux live distros
  - Learnix includes Radiance!
- Linux distros have become easier to install and use
- Virtualisation
  - VMware, VirtualBox, Qemu, etc.
- Running Radiance from a remote UNIX box, using Putty and Xming for example

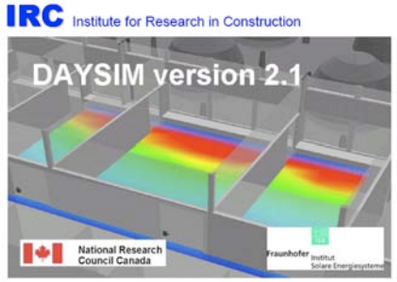
What do I need to use Radiance?

- Radiance "command line" tools
  - Binaries
  - Shell scripts
- CAD / 3D exporters
- HDR image viewers
- Radiance interactive viewers

Radiance distributions for Windows:

Into the maze





Radiance distributions for Windows: Comparison

commercial products

				
rayfront	adeline	desktop radiance	cygwin binaries	mingw binaries
development stopped in 2003	development stopped in 2002	development stopped in 2001	voluntarily supported and (almost) up-to-date	voluntarily supported and (almost) up-to-date

Radiance distributions for Windows: Comparison

commercial products

rayfront	adeline	desktop radiance	cygwin binaries	mingw binaries
includes a GUI, all "windows native" binaries and compiled shell scripts and more (materials, cal files, more up-to-date versions of winrview and winimage)	includes most "windows native" binaries and compiled shell scripts, but I would not recommend the GUI	includes all the "windows native" binaries and winrview and winimage	exactly the same Radiance binaries as in the source distribution, including the X11 programmes, but requires extra installation of the Cygwin infrastructure	only Radiance binaries, no shell scripts (yet) and no X11 programmes, but does not require any extra installation

The Radiance on Windows (free) survival kits

The Radiance on Windows (free) survival kits

- Kit 1: 
  - Cygwin + Cygwin Radiance distribution
- Kit 2: MinGW
  - MinGW Radiance distribution + interactive viewer + HDR image viewer

The Radiance on Windows Survival Kit:
Option 1:
Cygwin + Cygwin Radiance distribution

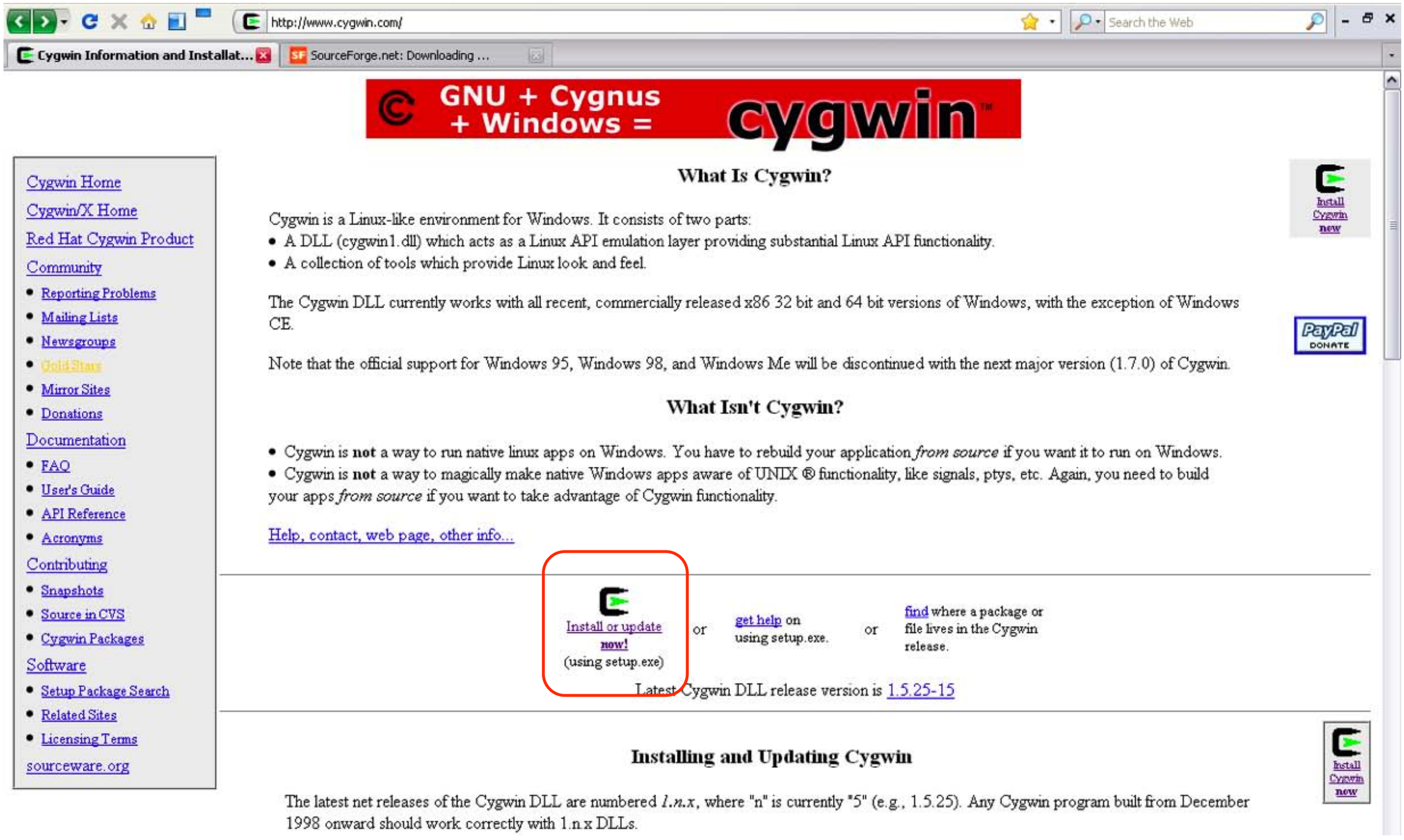
1) Download and install Cygwin
2) Download and install Cygwin Radiance
3) Run Cygwin
4) Run Radiance

- All UNIX programmes must be run from within the Cygwin shell
- X.org packages or Xming must be installed to run ximage, rvu, etc.

Cygwin

- Cygwin is a Linux-like environment for Windows.
- It consists of a DLL (`cygwin1.dll`), which acts as an emulation layer providing POSIX (Portable Operating System Interface) system call functionality, and a collection of tools, which provide a Linux look and feel.

1) Cygwin installation

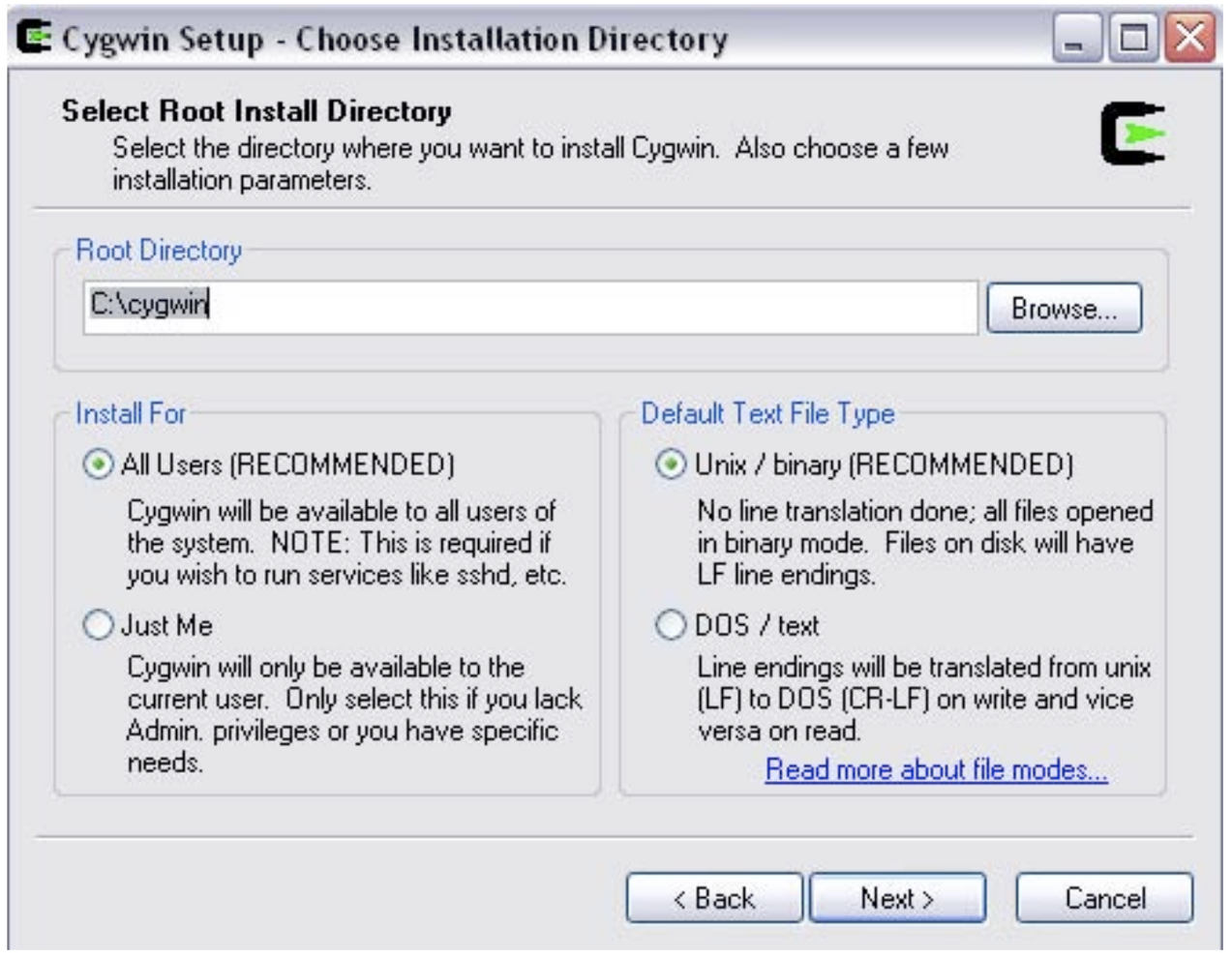


Get the setup programme

1) Cygwin installation

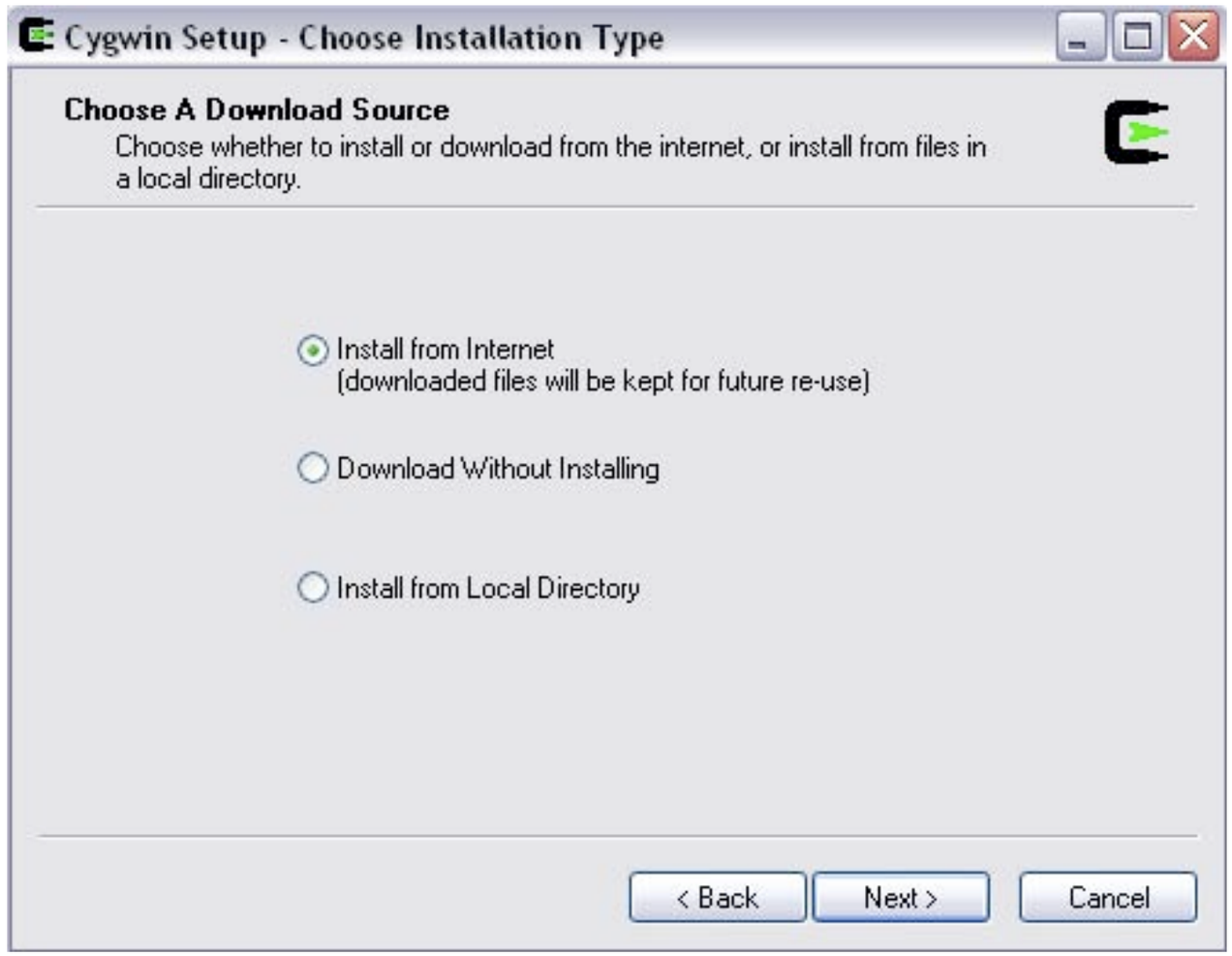
Run it

1) Cygwin installation



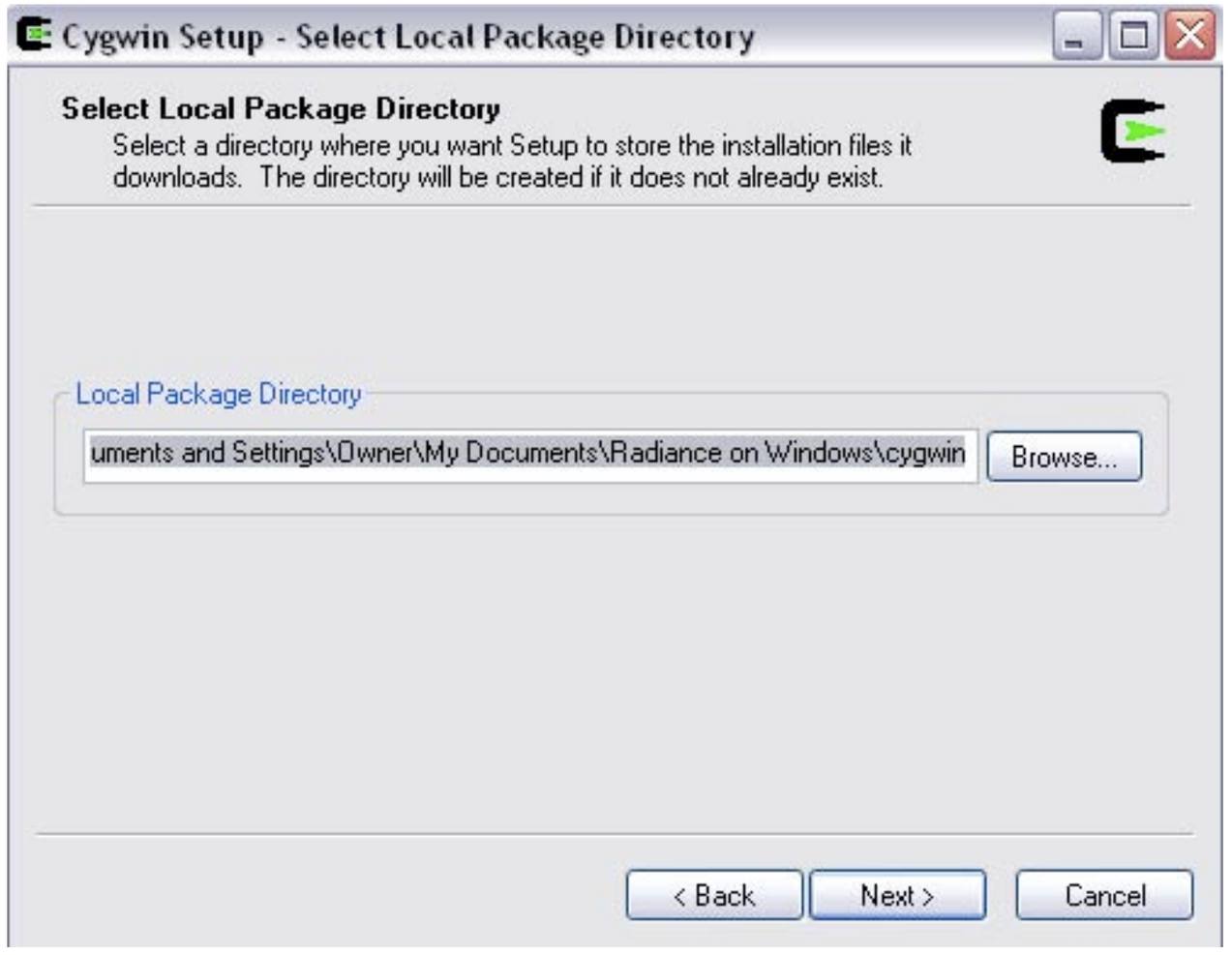
Choose installation directory

1) Cygwin installation



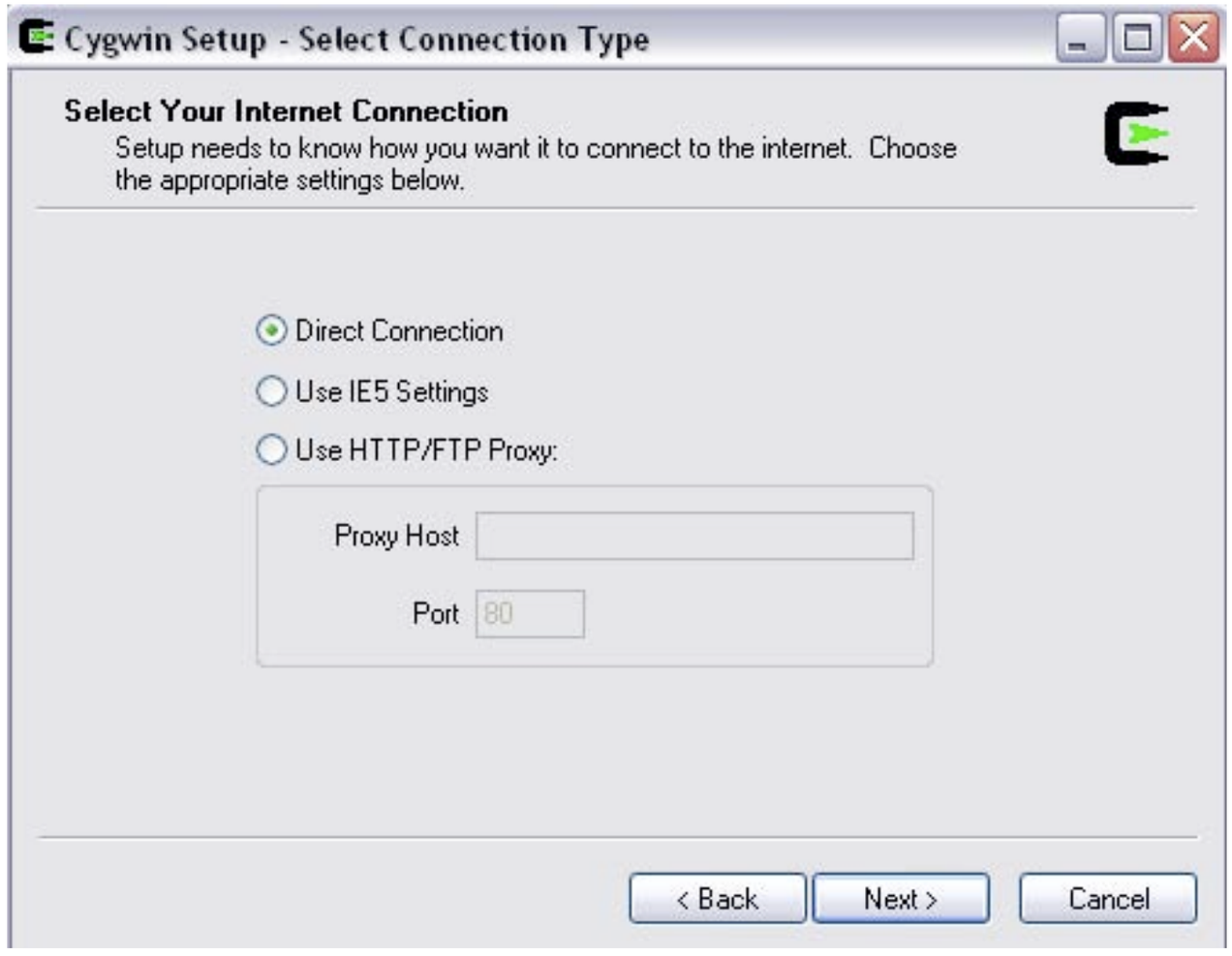
Choose whether to install from the Internet or from a local directory, or to download only

1) Cygwin installation



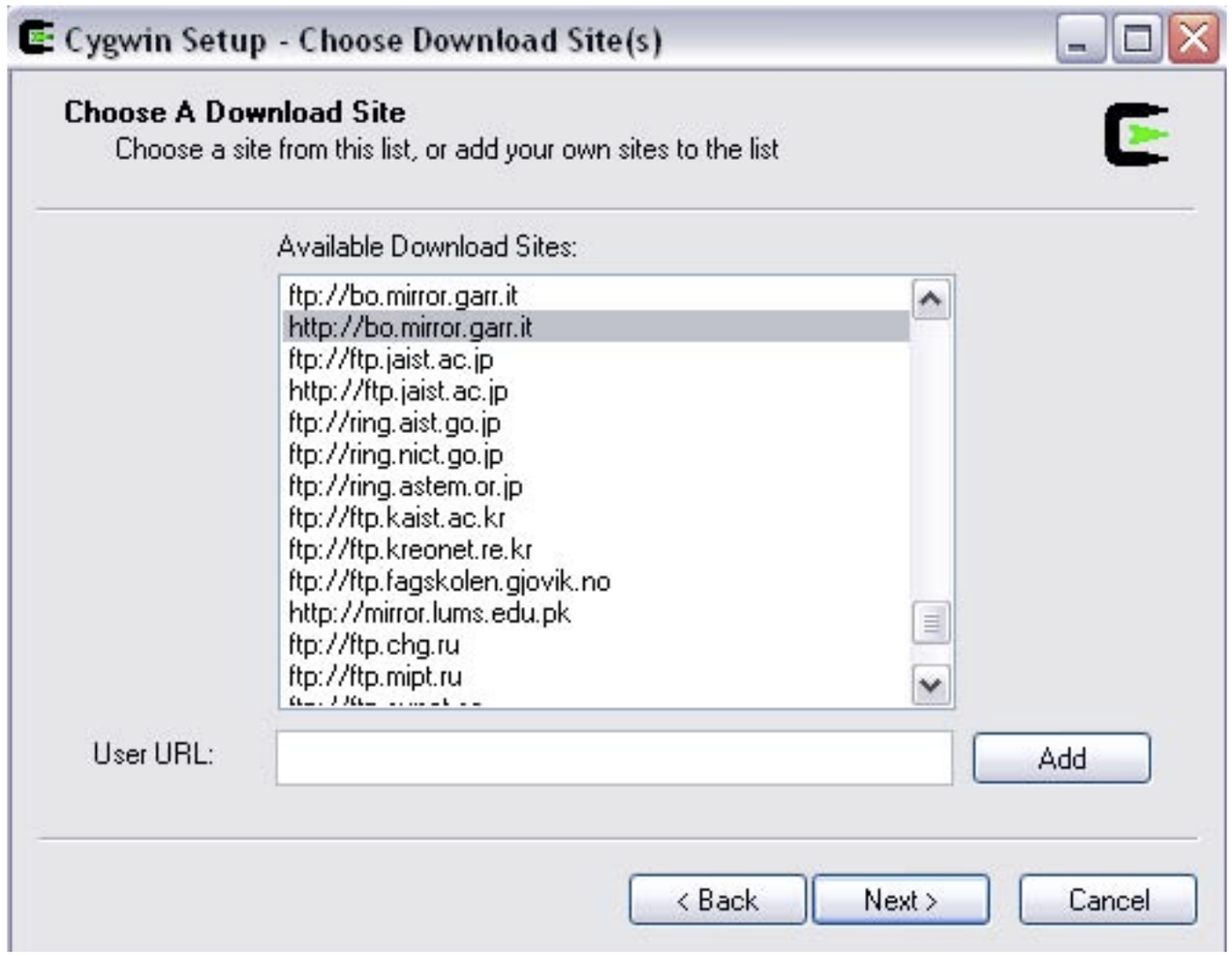
Select where to save downloaded packages

1) Cygwin installation



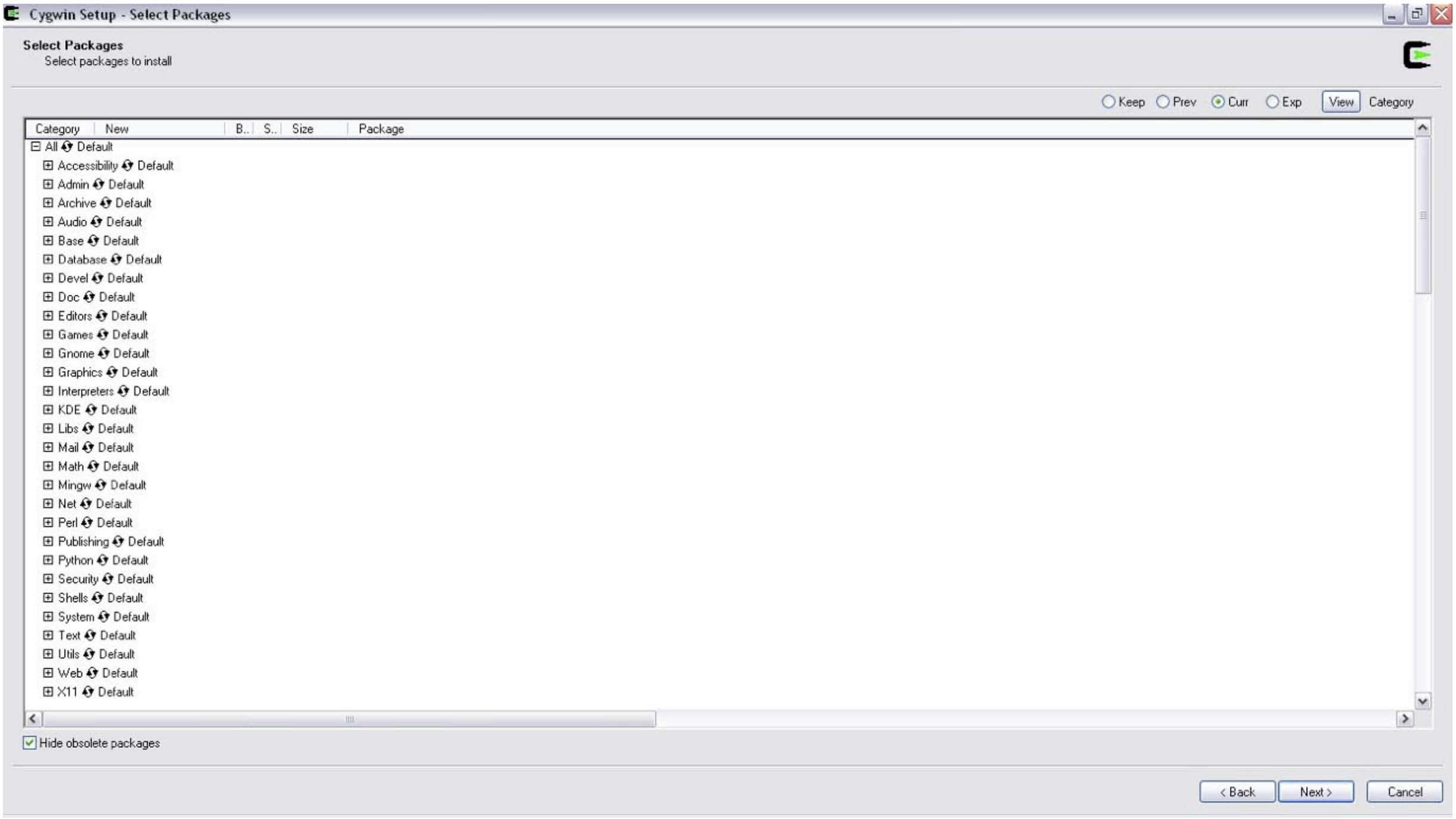
Select the type of connection and proxy settings

1) Cygwin installation



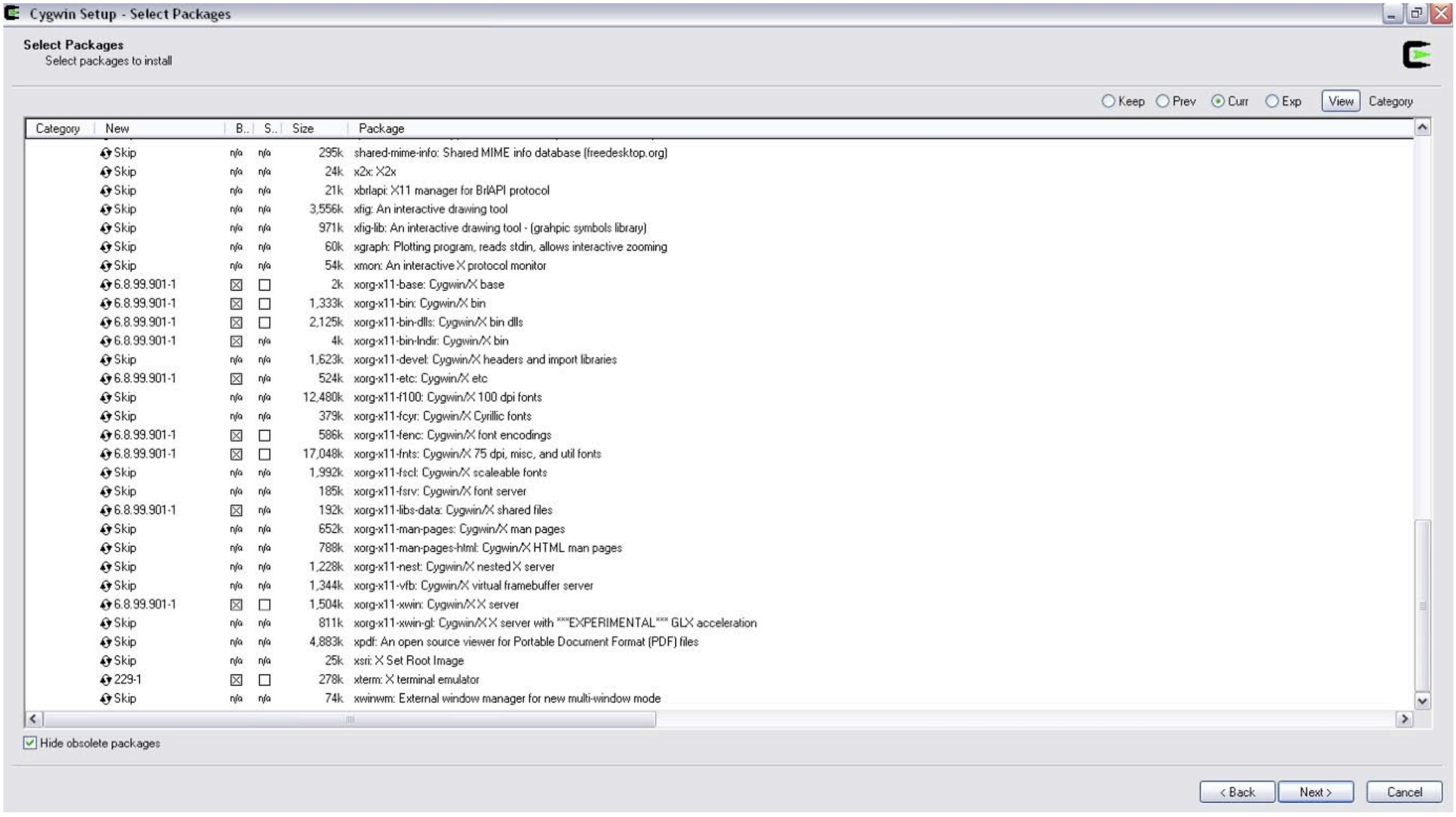
Choose download mirror

1) Cygwin installation



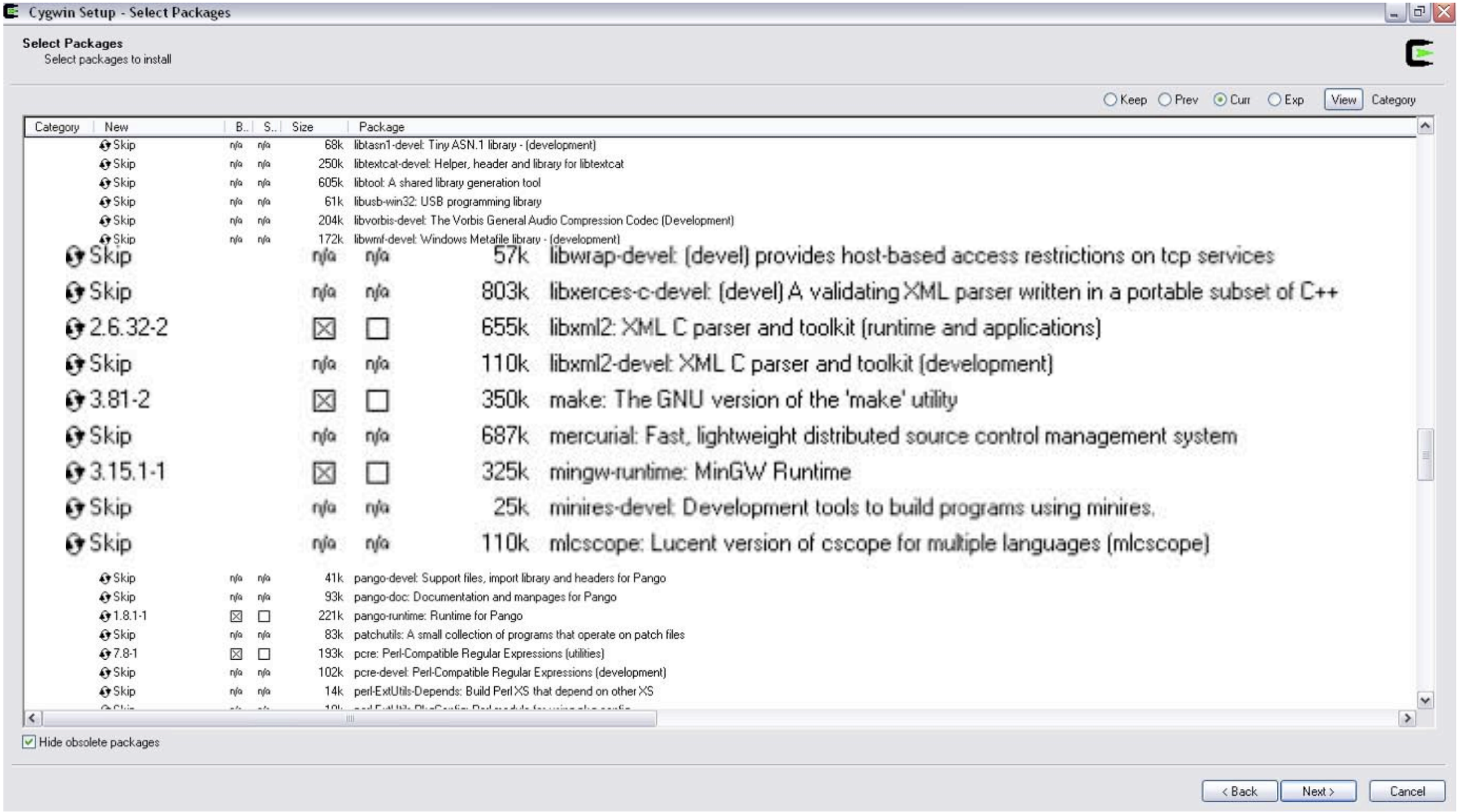
Select packages to install

1) Cygwin installation



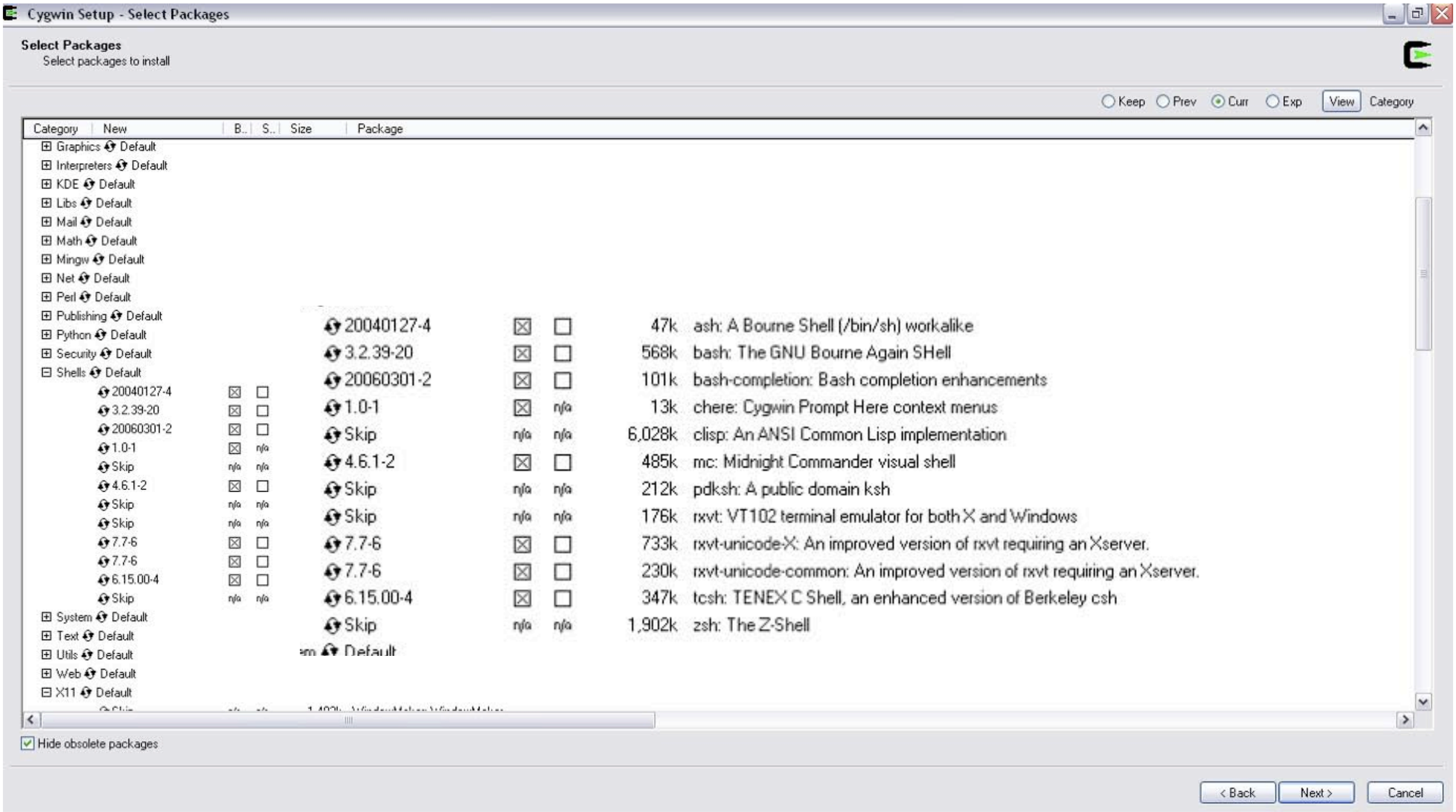
Don’t forget to select X.org packages

1) Cygwin installation



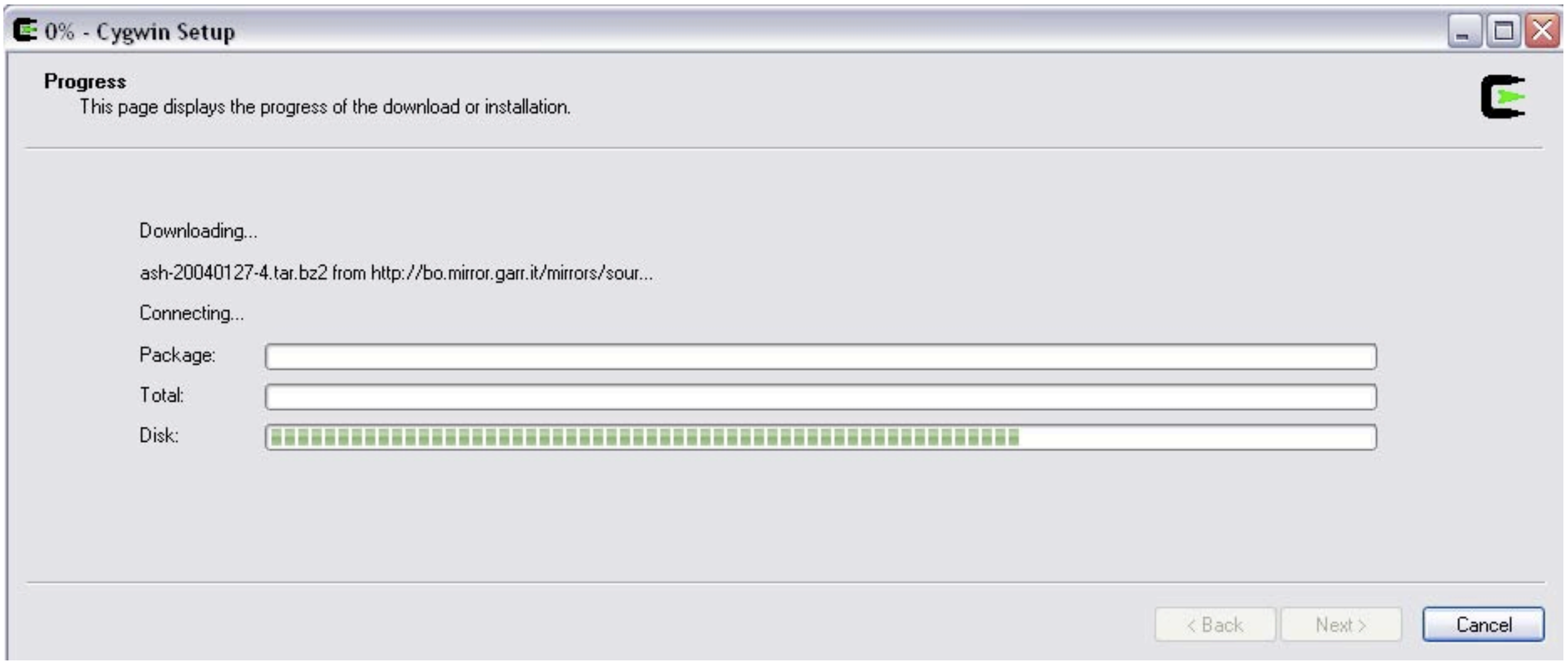
Selecting “make” can be useful

1) Cygwin installation



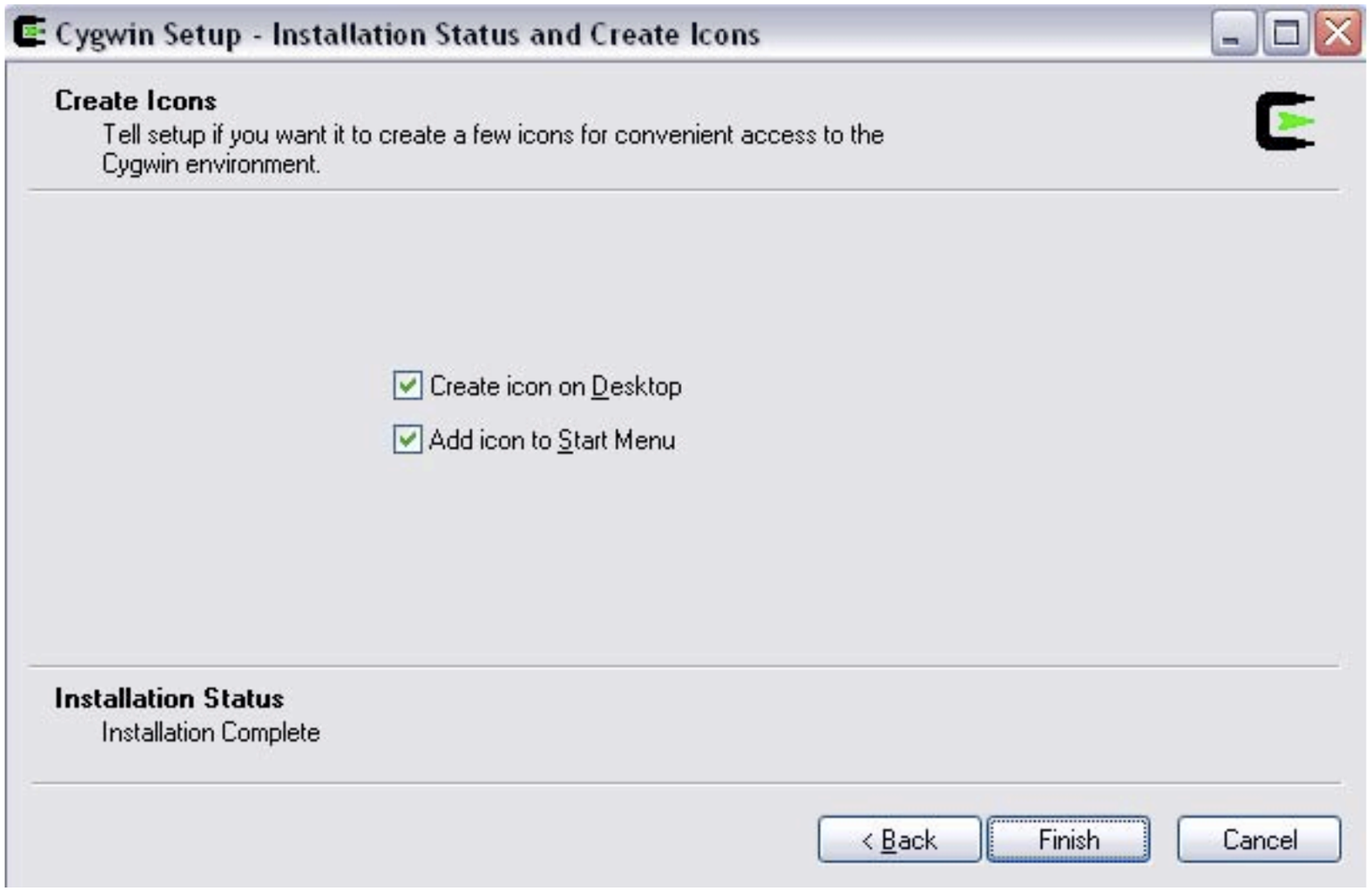
Don’t forget to select the “tcsh” package

1) Cygwin installation



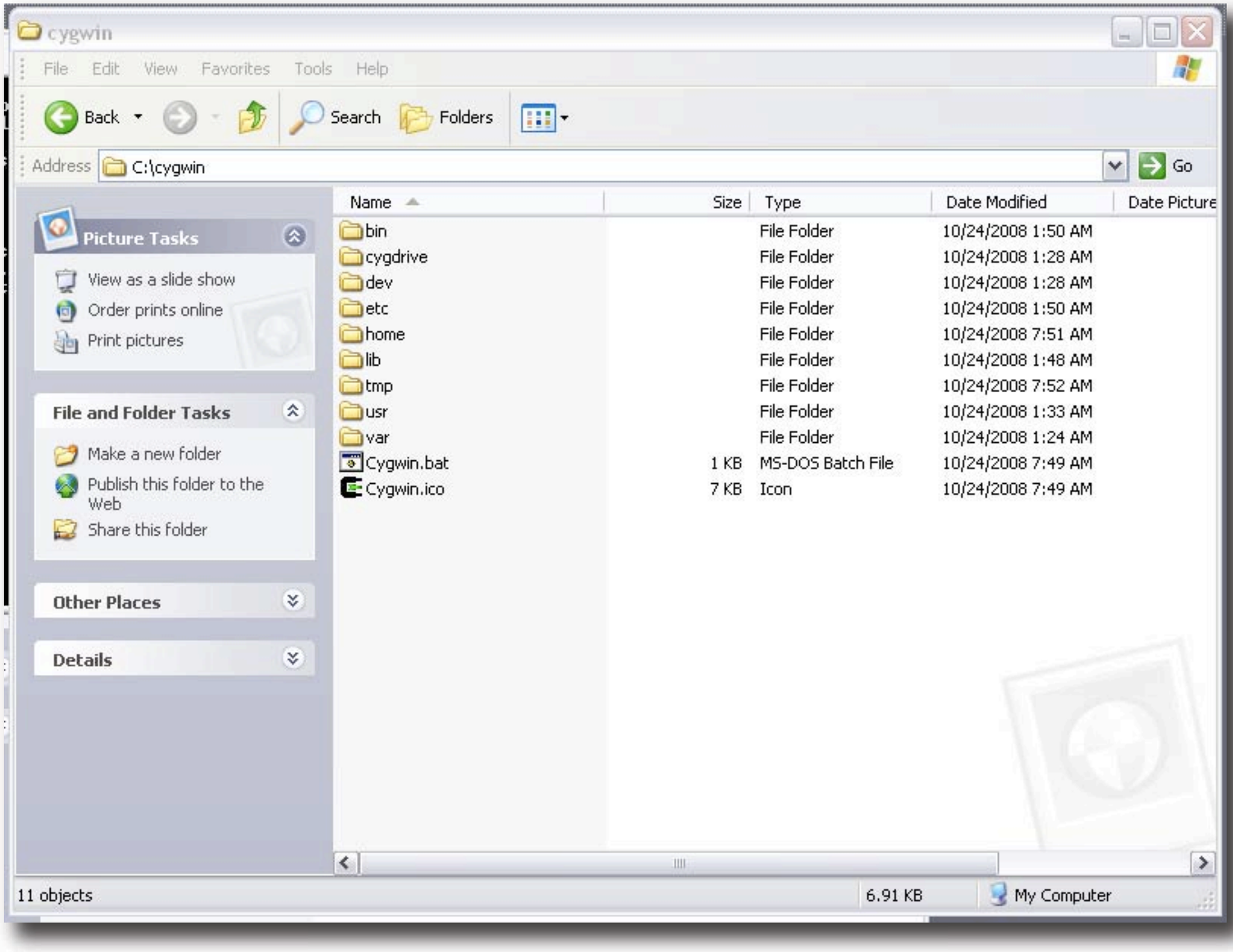
Wait a bit ...

1) Cygwin installation

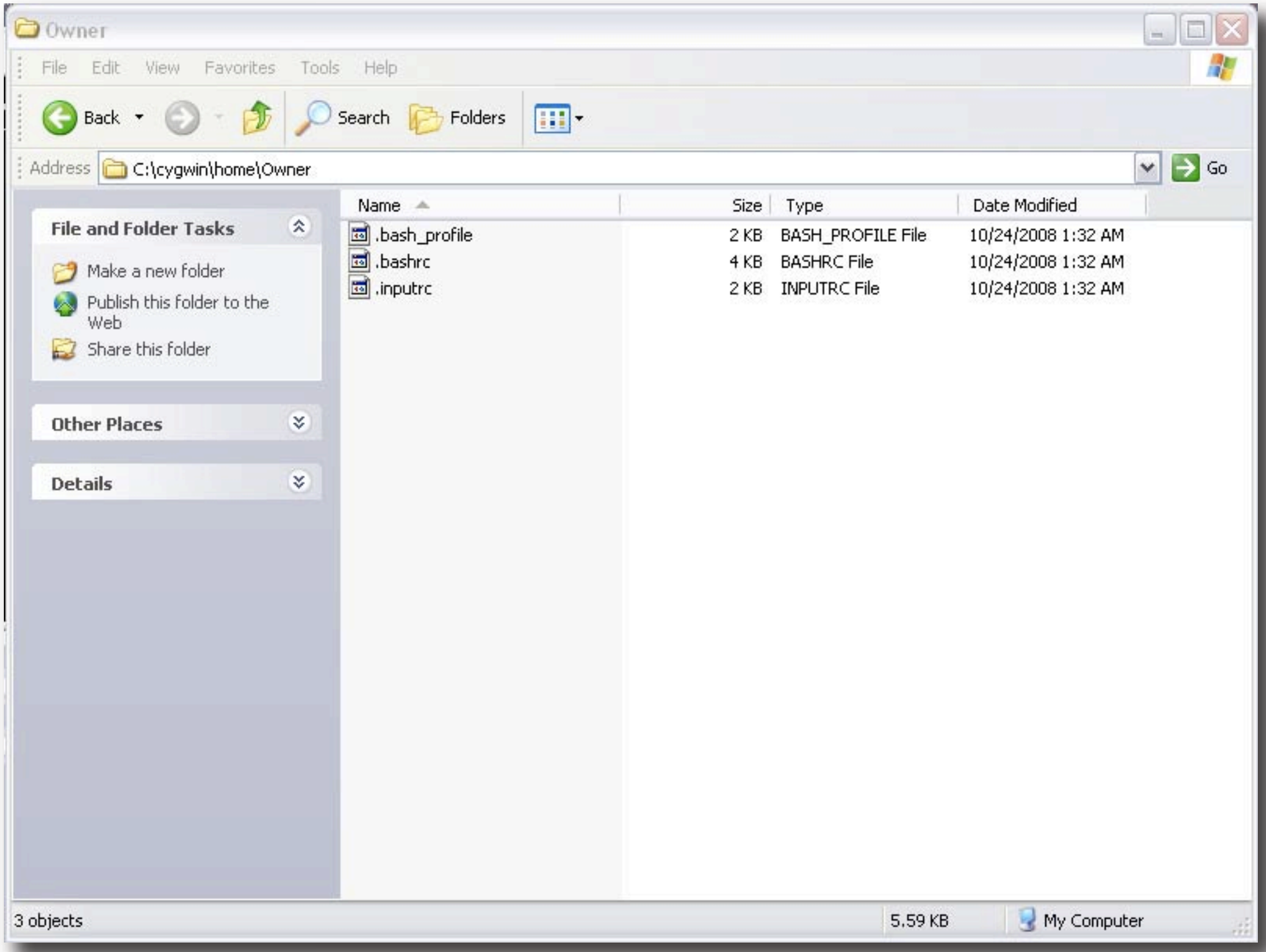


... until the final click

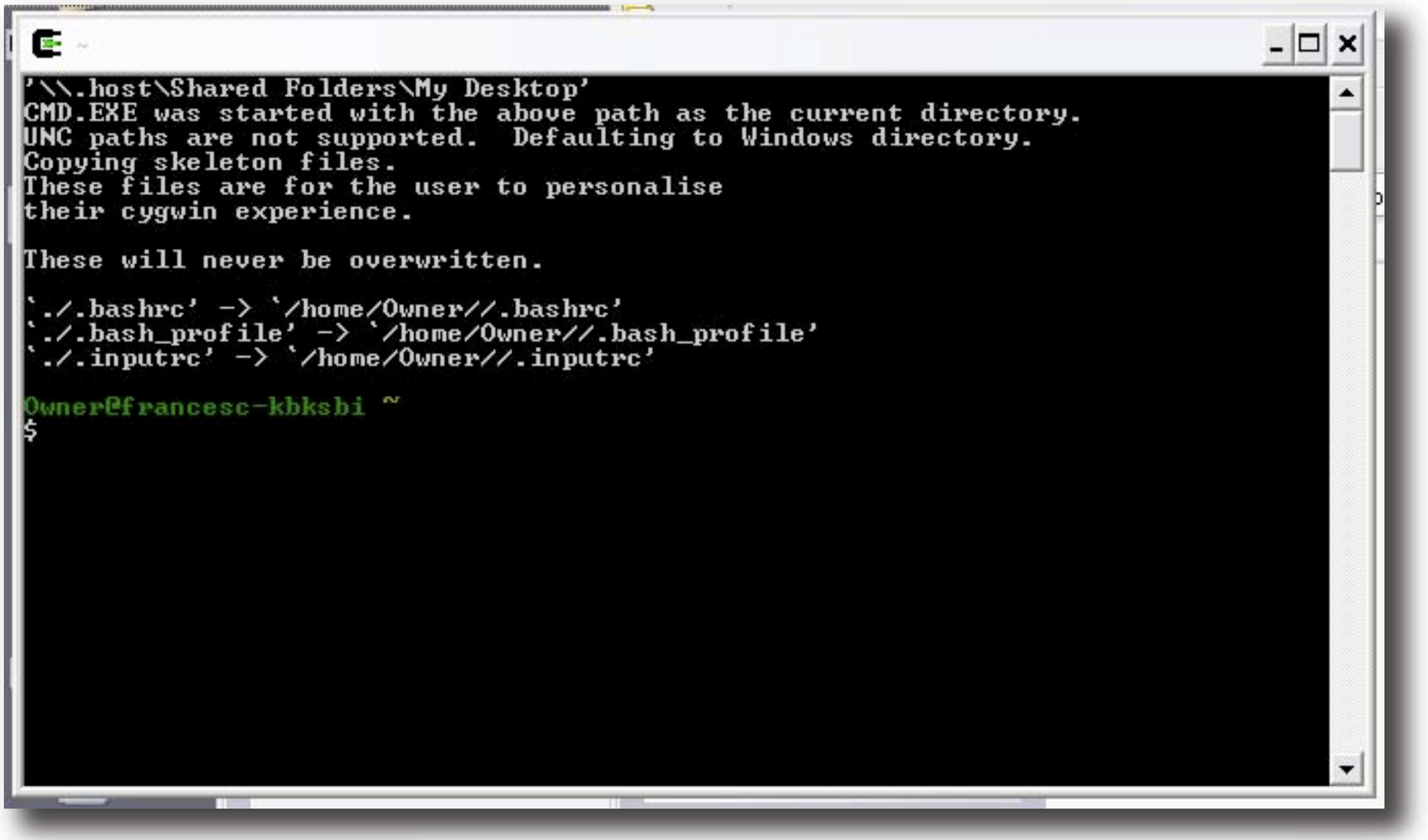
1) Cygwin installation



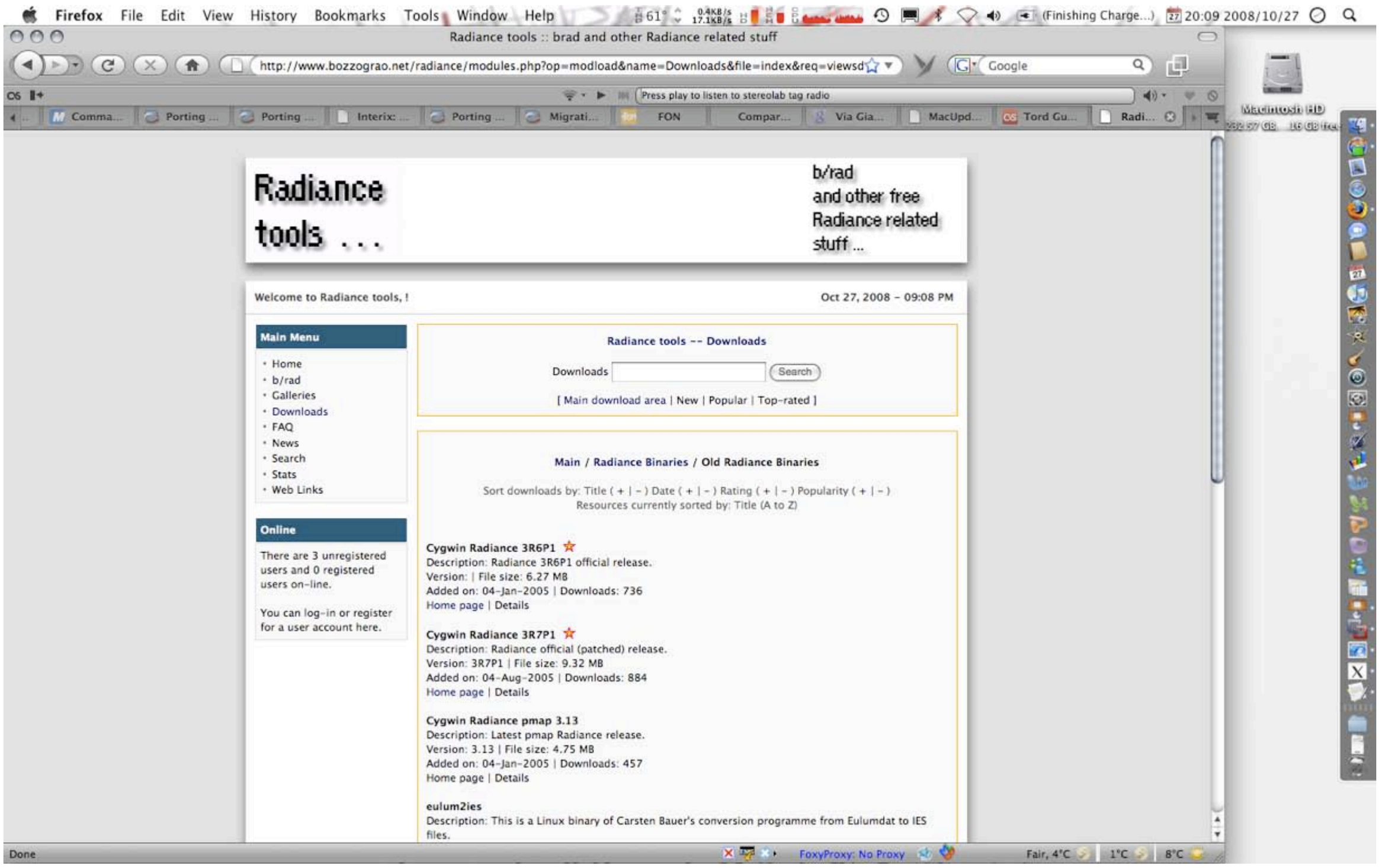
1) Cygwin installation



1) Cygwin installation



2) Radiance installation



Download the Cygwin Radiance package from http://www.bozzograo.net/radiance

2) Radiance installation

1. Launch the Cygwin console (double-click on the Cygwin desktop icon)
2. Change to root directory (don't type the dollar sign):

```
$ cd /
```

3. Create the "opt" directory:

```
$ mkdir /opt
```

4. Create the "radiance" directory inside /opt:

```
$ cd /opt
$ mkdir radiance
```

5. Copy the radiance_cygwin_3R9.tar.gz inside C:\cygwin\opt\radiance
6. Extract the archive:

```
$ cd radiance
$ tar zxvf radiance_cygwin_3R9.tar.gz
```

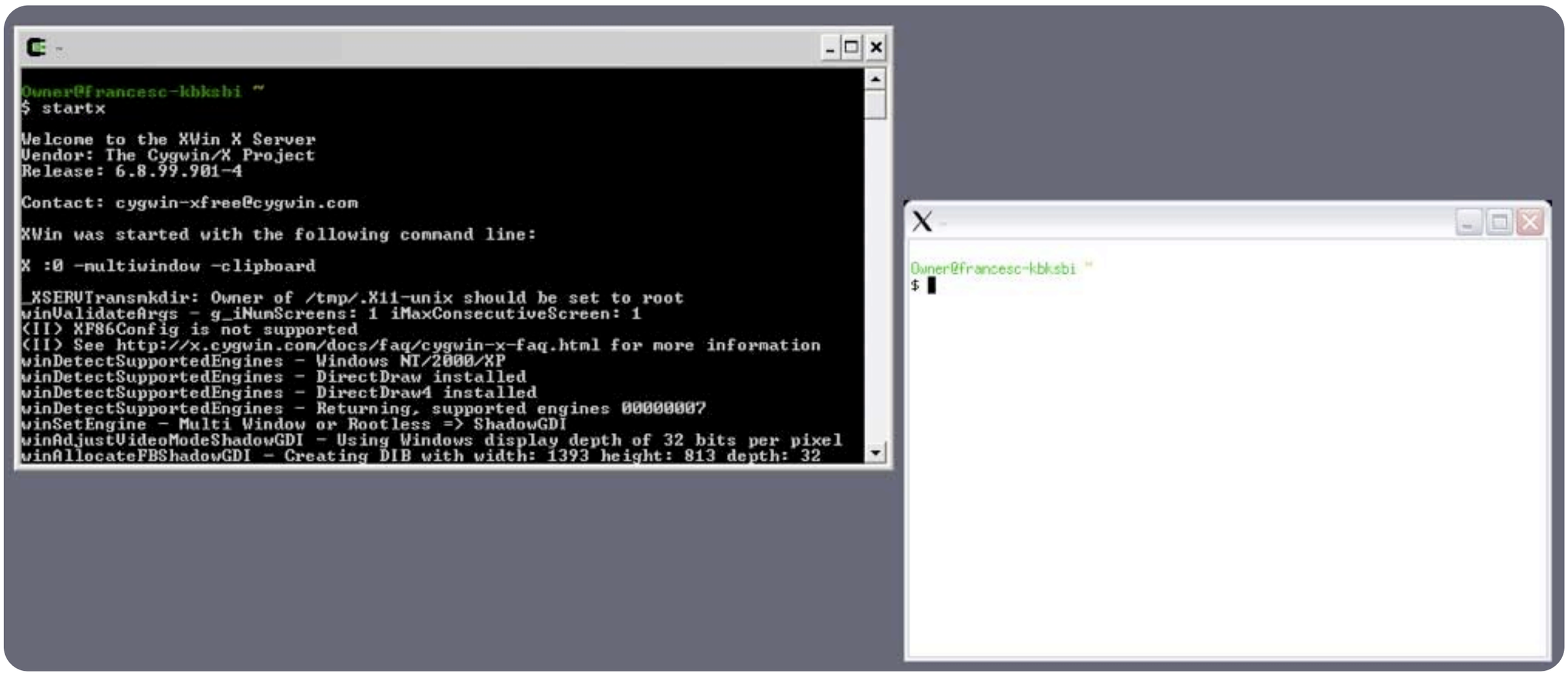
7. Append bash_profile file to the one in the home directory:

```
$ cat bash_profile >> ~/.bash_profile
```

8. Close the Cygwin console.
9. Re-launch the Cygwin console.

3) Run Cygwin

- X11
  - either use the provided X.org (type `startx` from Cygwin command line) ...



3) Run Cygwin

- ... or use Xming

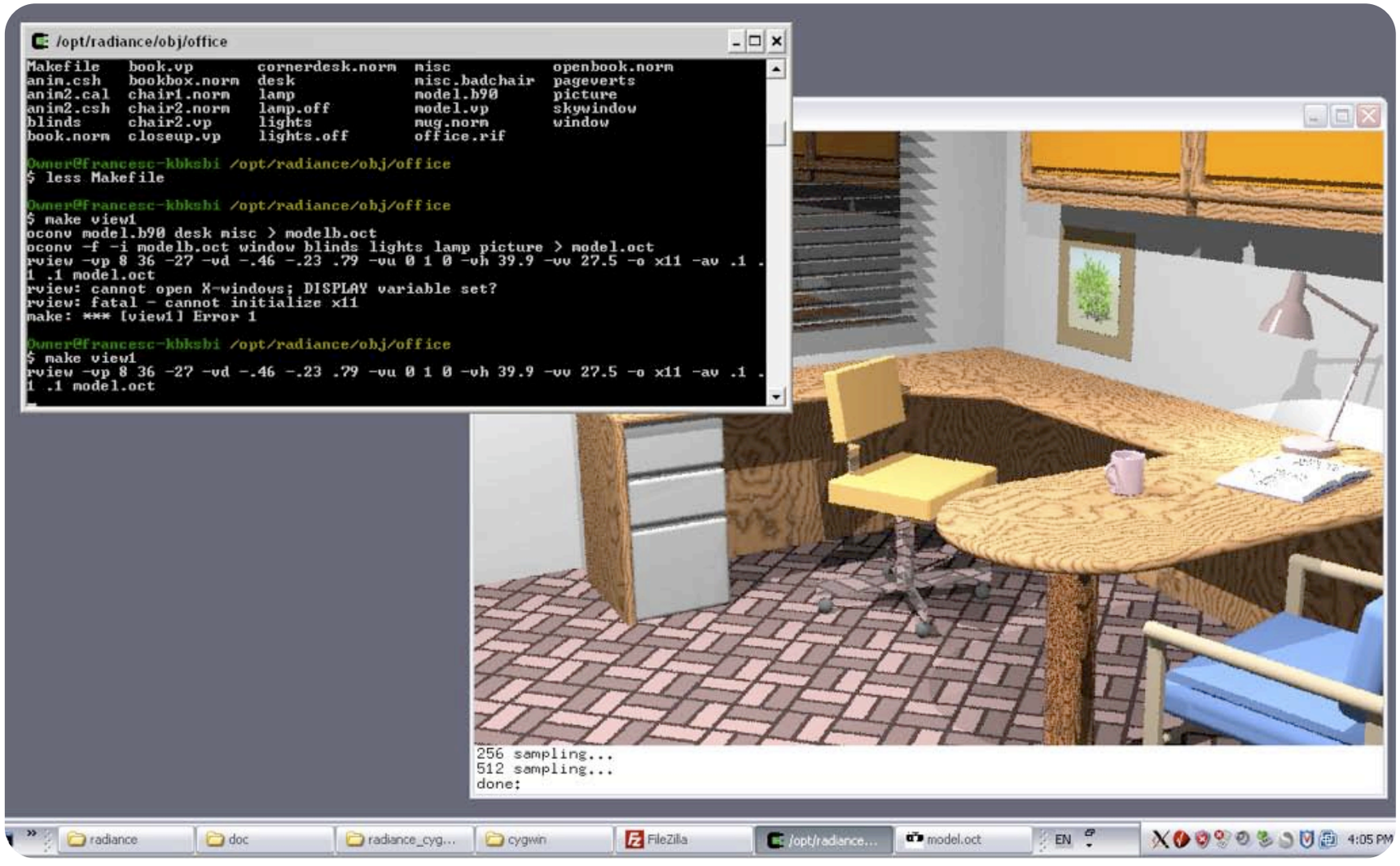
  http://www.straightrunning.com/XmingNotes/)

  first run the Xming server

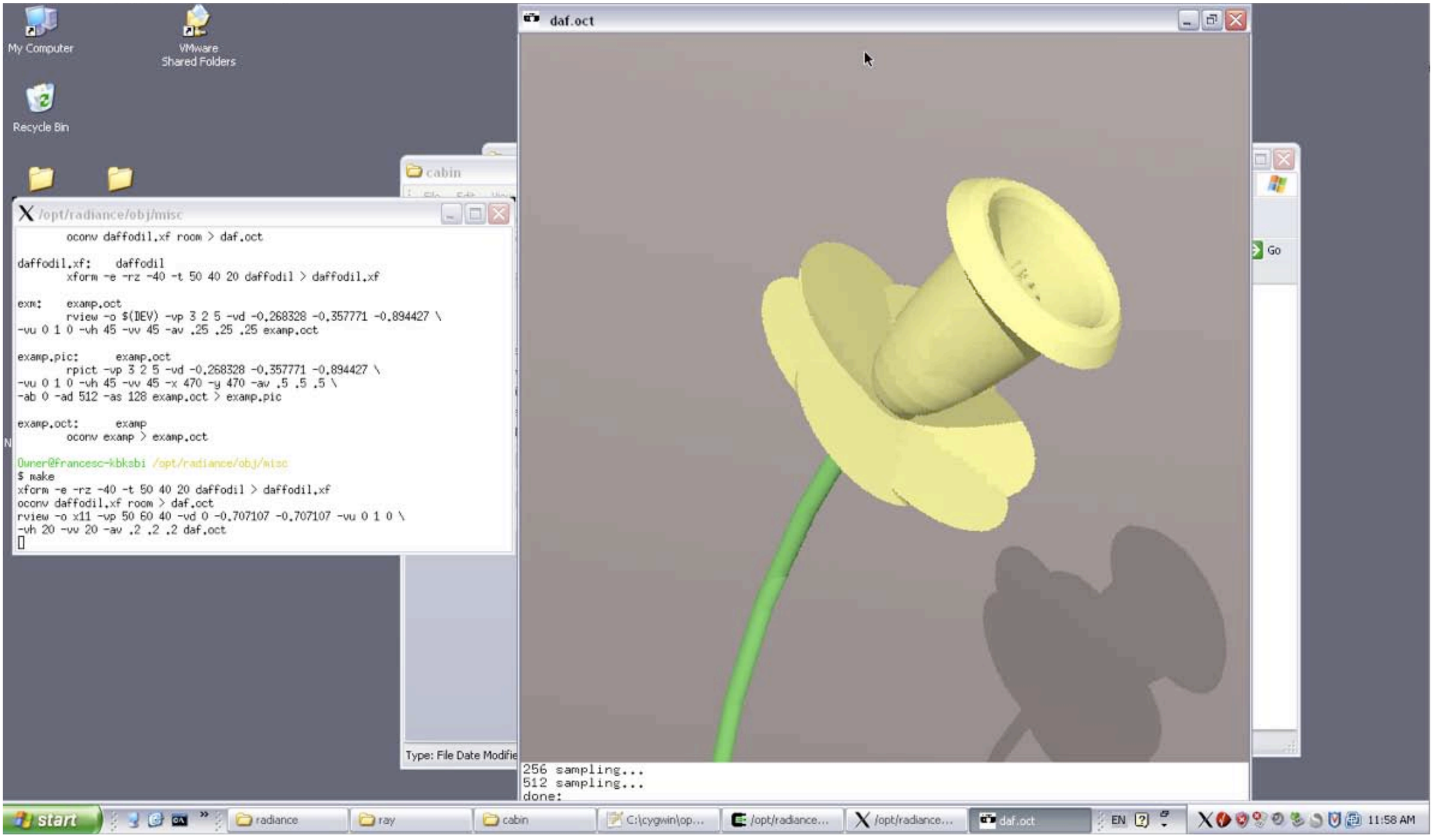
  then export the DISPLAY variable

  ```
  $ export DISPLAY=:0
  ```

  then run the Radiance X11 applications



4) Run Radiance



Cygwin shell and performance

- By default no Cygwin program can allocate more than 384 MB of memory (program+data). To use more real or virtual memory, add an entry in the either the HKEY_LOCAL_MACHINE (to change the limit for all users) or HKEY_CURRENT_USER (for just the current user) section of the registry:
    - Add the DWORD value heap_chunk_in_mb and set it to the desired memory limit in decimal Mb, or
    - using the regtool program included in the Cygwin package, like in this example that sets memory limit to 1024 MB:

```
regtool -i set /HKLM/Software/Cygnus\ Solutions/Cygwin/
heap_chunk_in_mb 1024

regtool -v list /HKLM/Software/Cygnus\ Solutions/Cygwin
```

- Cygwin supports both Win32- and POSIX-style paths, using either forward or back slashes as the directory delimiter. This means that no canges to Makefiles and shell scripts are required.
- UNC pathnames (starting with two slashes) are supported.

Cygwin shell and performance

- The Windows filesystem is accessible from the `/cygdrive` mount point.
- `fork` calls are slower than `spawn` calls.
- Executable program filenames end with .exe but the .exe need not be included in the command, so that traditional UNIX names can be used. For programs that end in .bat and .com, you cannot omit the extension.
- UNIX and Win32 use different end-of-line terminators in text files. Consequently, carriage-return newlines have to be translated on the fly by Cygwin into a single newline when reading in text mode. This seems to slow down command line pipelines. This solution addresses the compatibility requirement at the expense of violating the POSIX standard that states that text and binary mode will be identical. Consequently, processes that attempt to `lseek` through text files can no longer rely on the number of bytes read as an accurate indicator of position in the file. For this reason, the CYGWIN environment variable can be set to override this behavior.
- Refer to the user guide for more information:

  `C:\cygwin\usr\share\doc\cygwin-doc-1.4\cygwin-ug-net.pdf`

Cygwin Radiance To Do

- Package Radiance so that it can be installed directly from the Cygwin installation programme `setup.exe`

The Radiance on Windows Survival Kit:
Option 2:
MinGW Radiance + interactive viewer + HDR viewer

MinGW

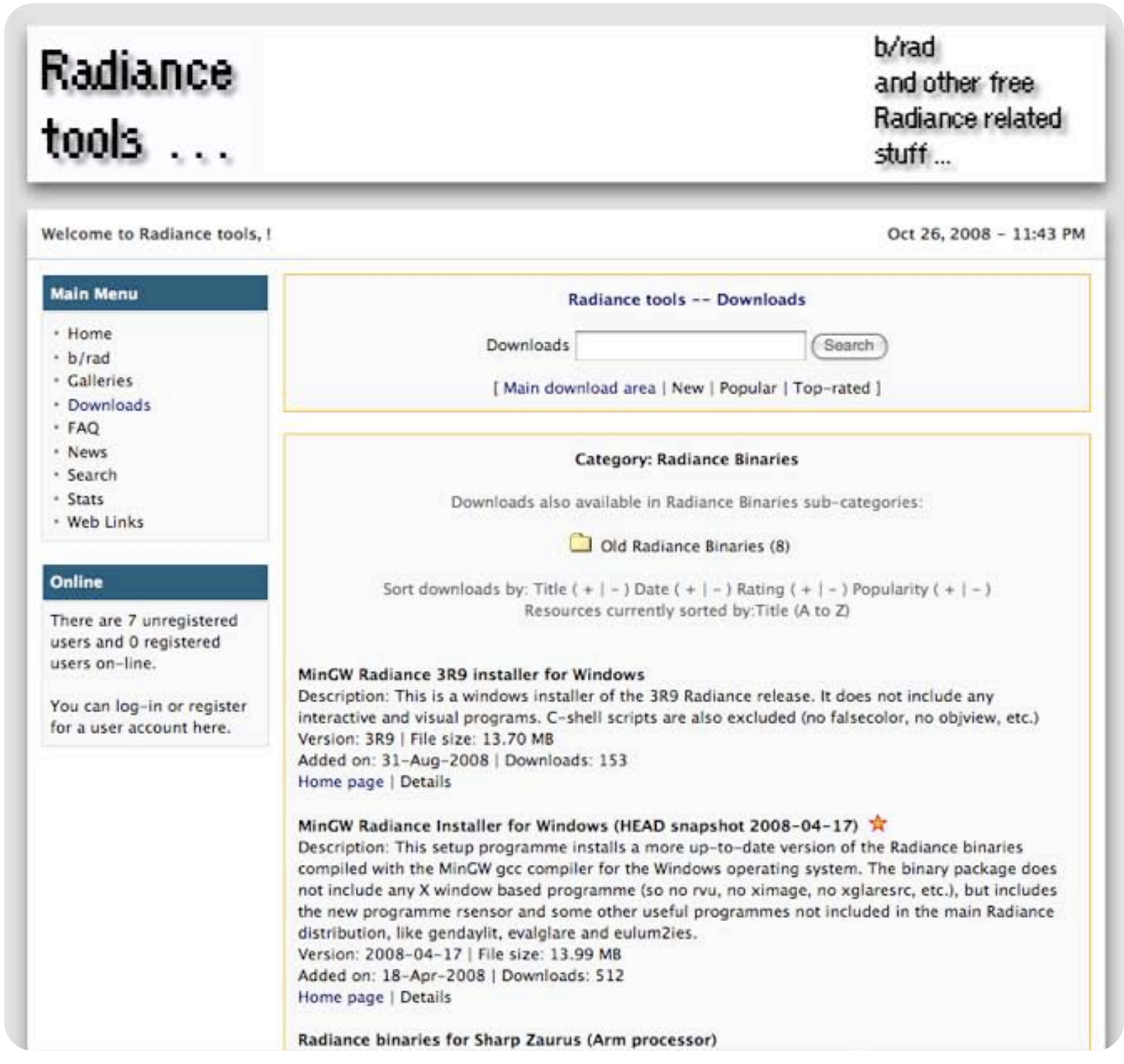
- Download and install MinGW
- Interactive viewer
  - nrv (http://www.aisarquitectura.com/nrv/)
  - winrview (Desktop Radiance http://radsite.lbl.gov/deskrad/)
- HDR image viewer
  - winimage (Desktop Radiance http://radsite.lbl.gov/deskrad/)
  - raddisplay (http://deluminaelab.com/en/raddisplay_details.html)
  - QTpfsGUI (http://qtpfsgui.sourceforge.net/)

MinGW

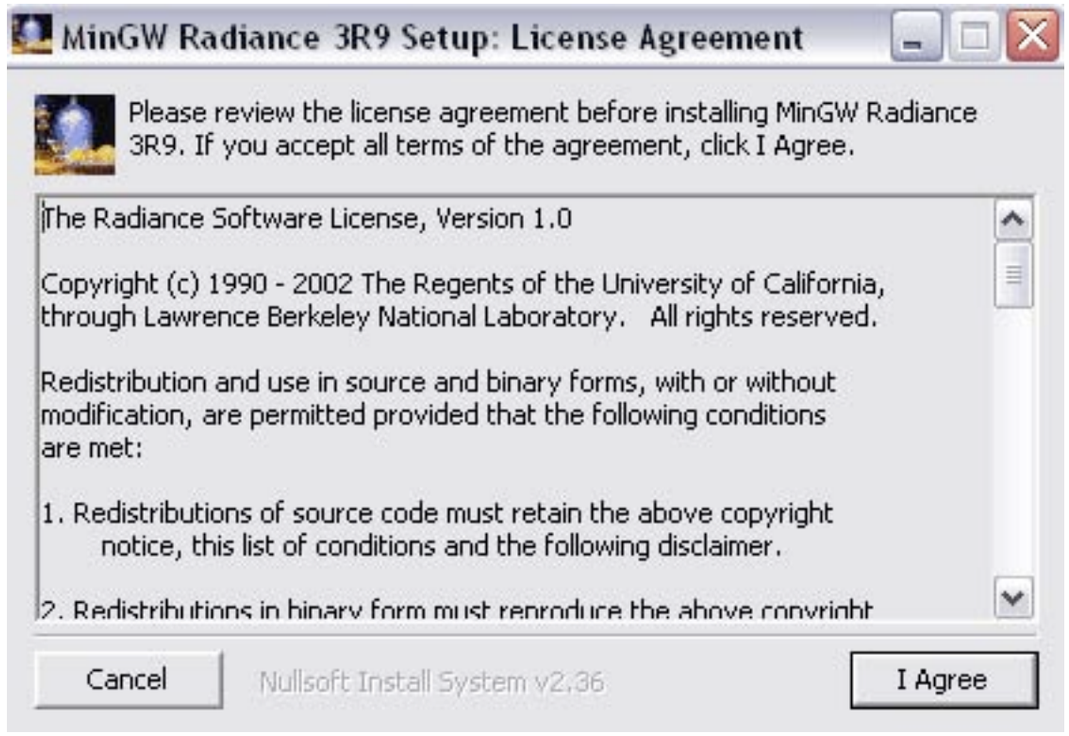
- MinGW: A collection of freely available and freely distributable Windows specific header files and import libraries, additional to the GNU Compiler Collection (GCC) and its associated tools (GNU binutils). MinGW provides a complete Open Source programming tool set which is suitable for the development of native Windows programmes that do not depend on any 3rd-party C runtime DLLs.

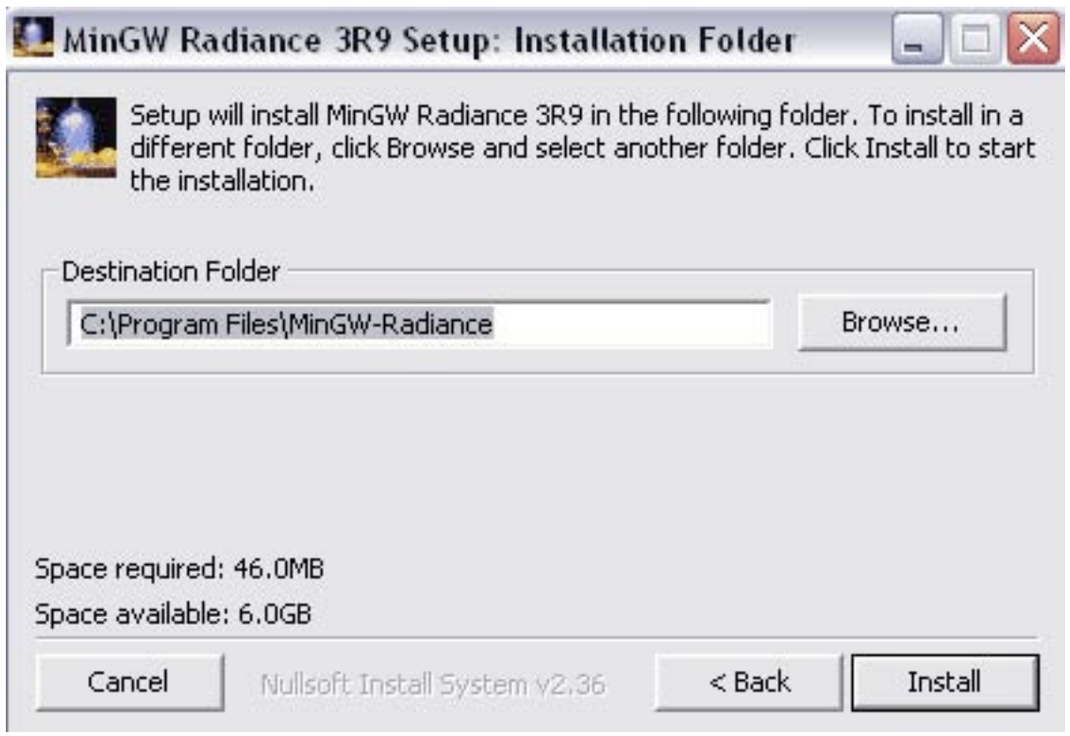
- MSYS: A Minimal SYStem providing a POSIX compatible Bourne shell environment, with a small collection of UNIX command line tools. Primarily developed as a means to execute the configure scripts and Makefiles used to build Open Source software, but also useful as a general purpose command line interface to replace Windows cmd.exe.

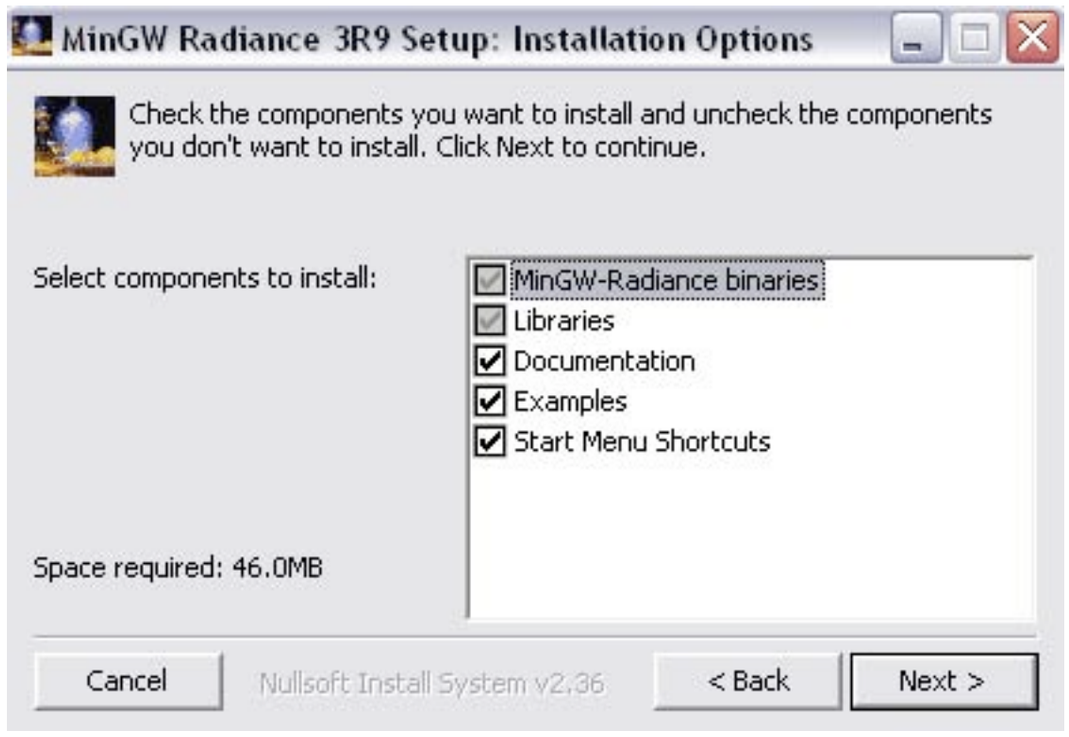
1) Download MinGW Radiance

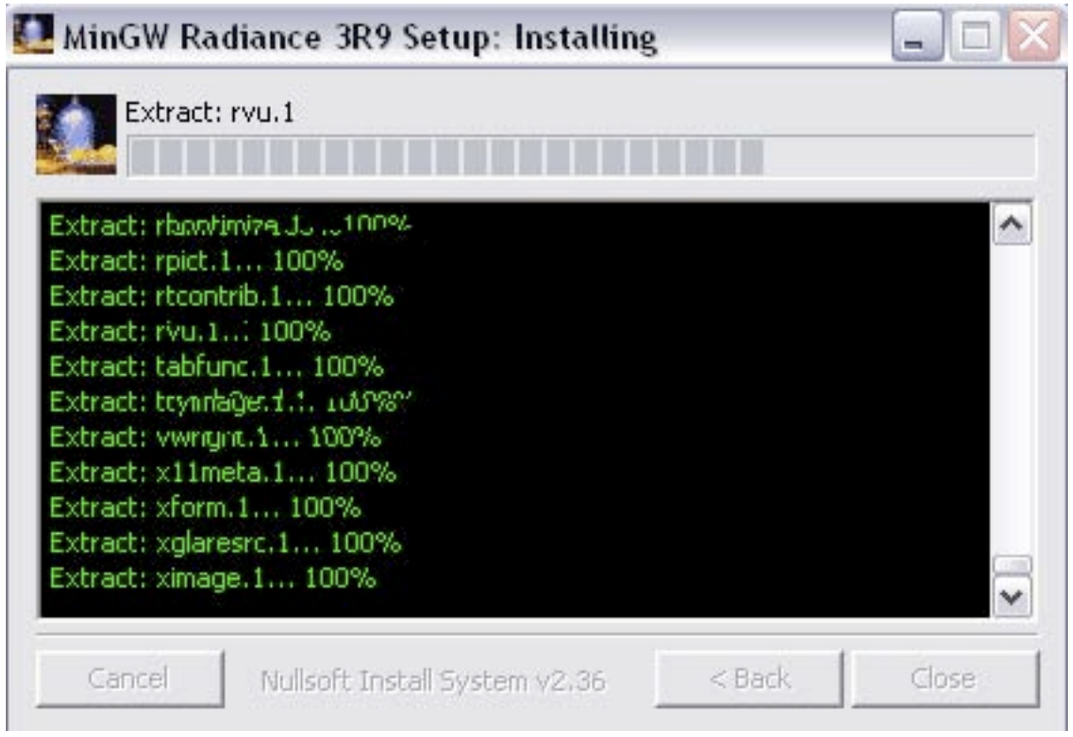


2) Install MinGW Radiance

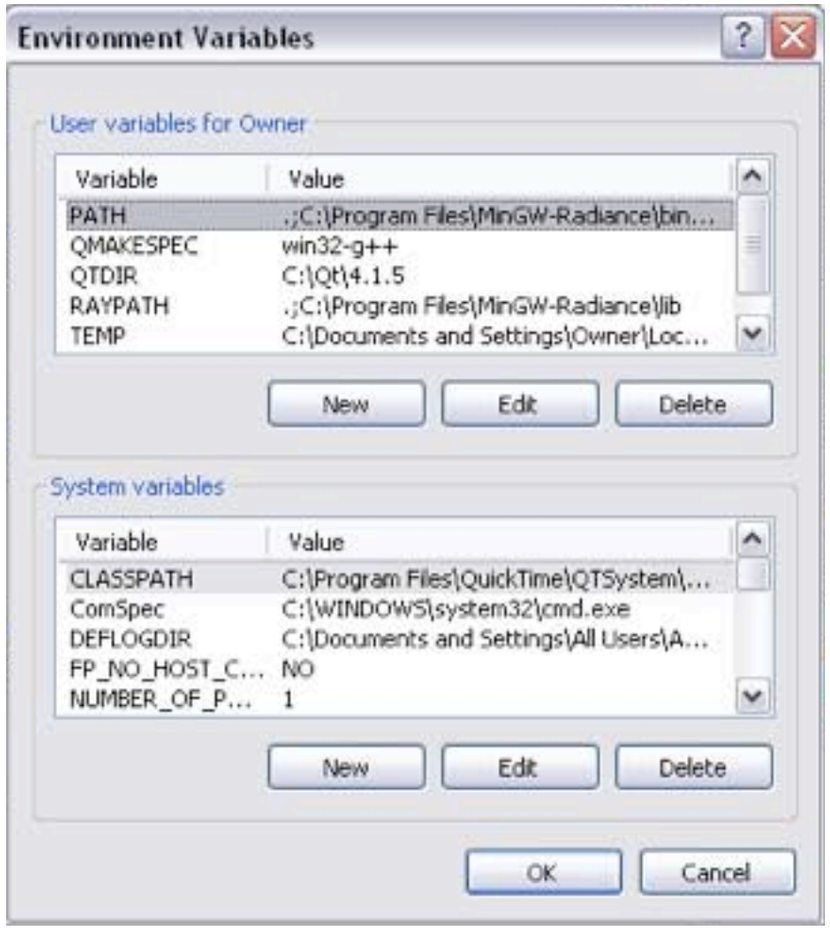




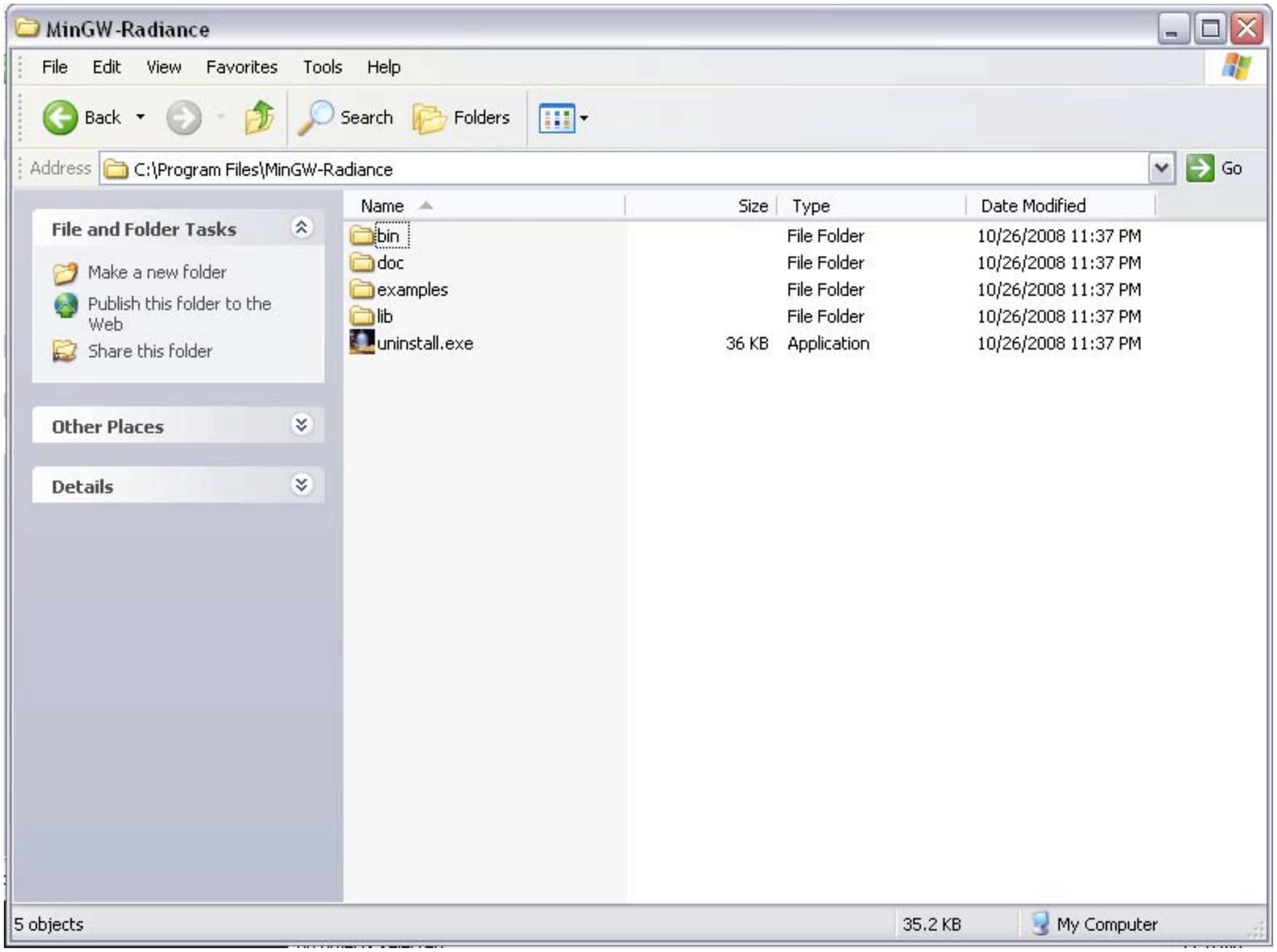




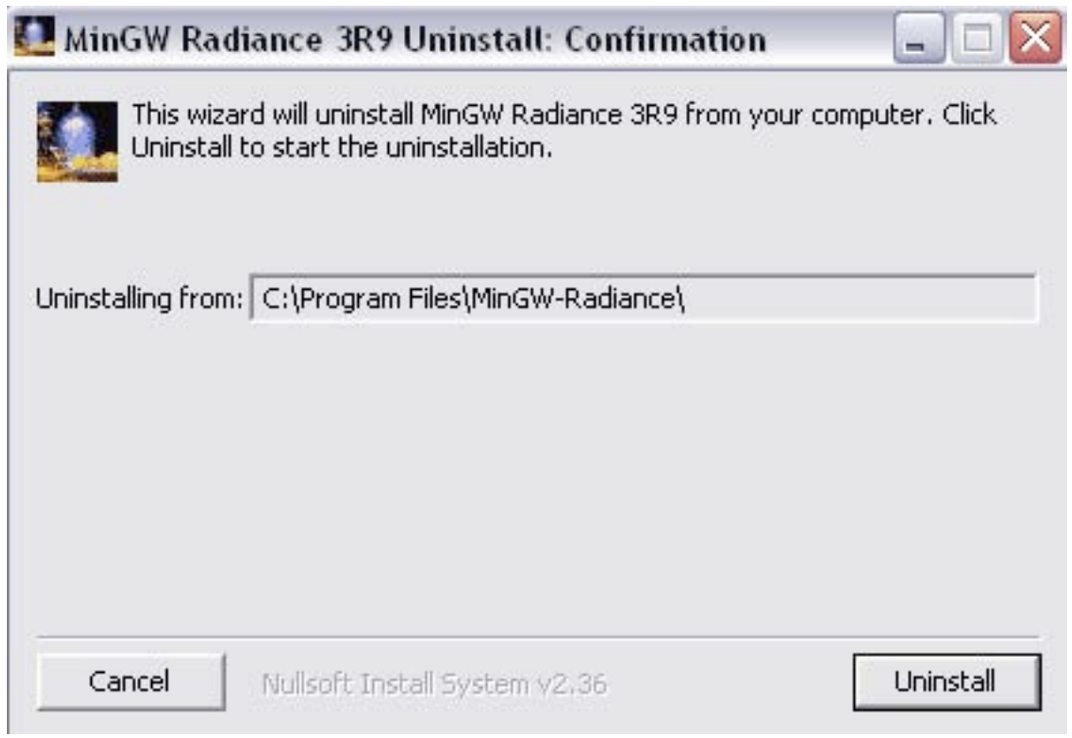
2) Install MinGW Radiance

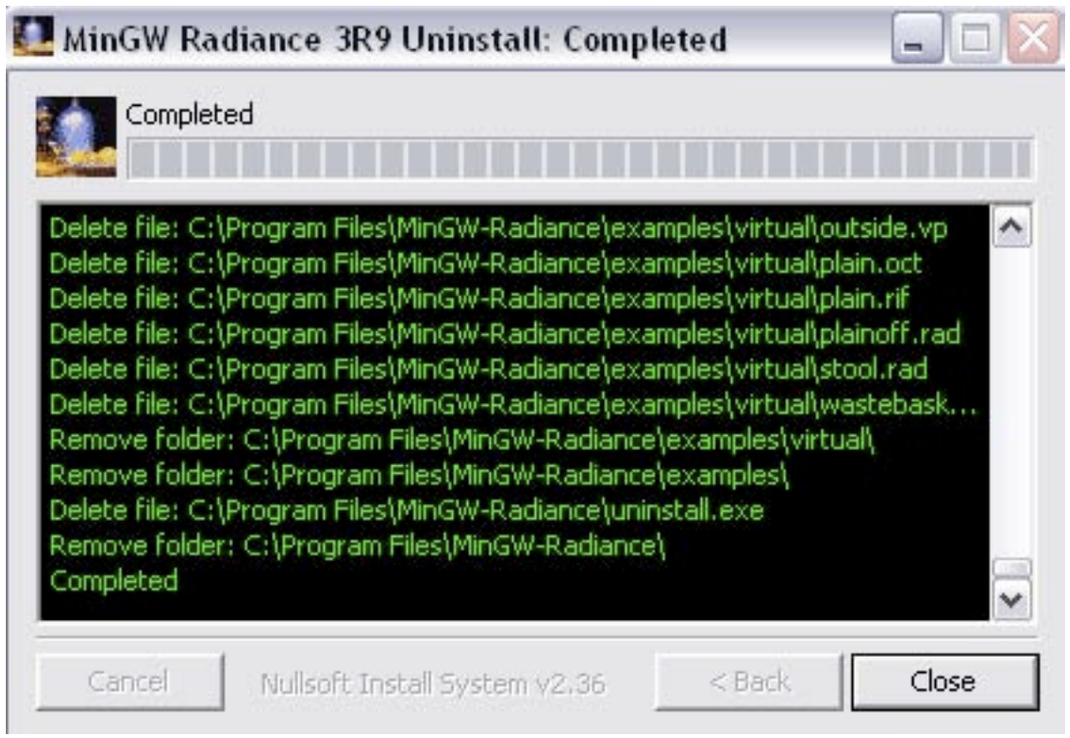


2) Install MinGW Radiance



Uninstall MinGW Radiance





Some Windows command lines oddities

- The command line has a very limited number of characters that can be input (8191 for Windows Xp, 2047 for Microsoft Windows 2000 or Windows NT 4.0). This unfortunately includes the PATH environmental variable, and has many implications on running long pipelines with Radiance commands

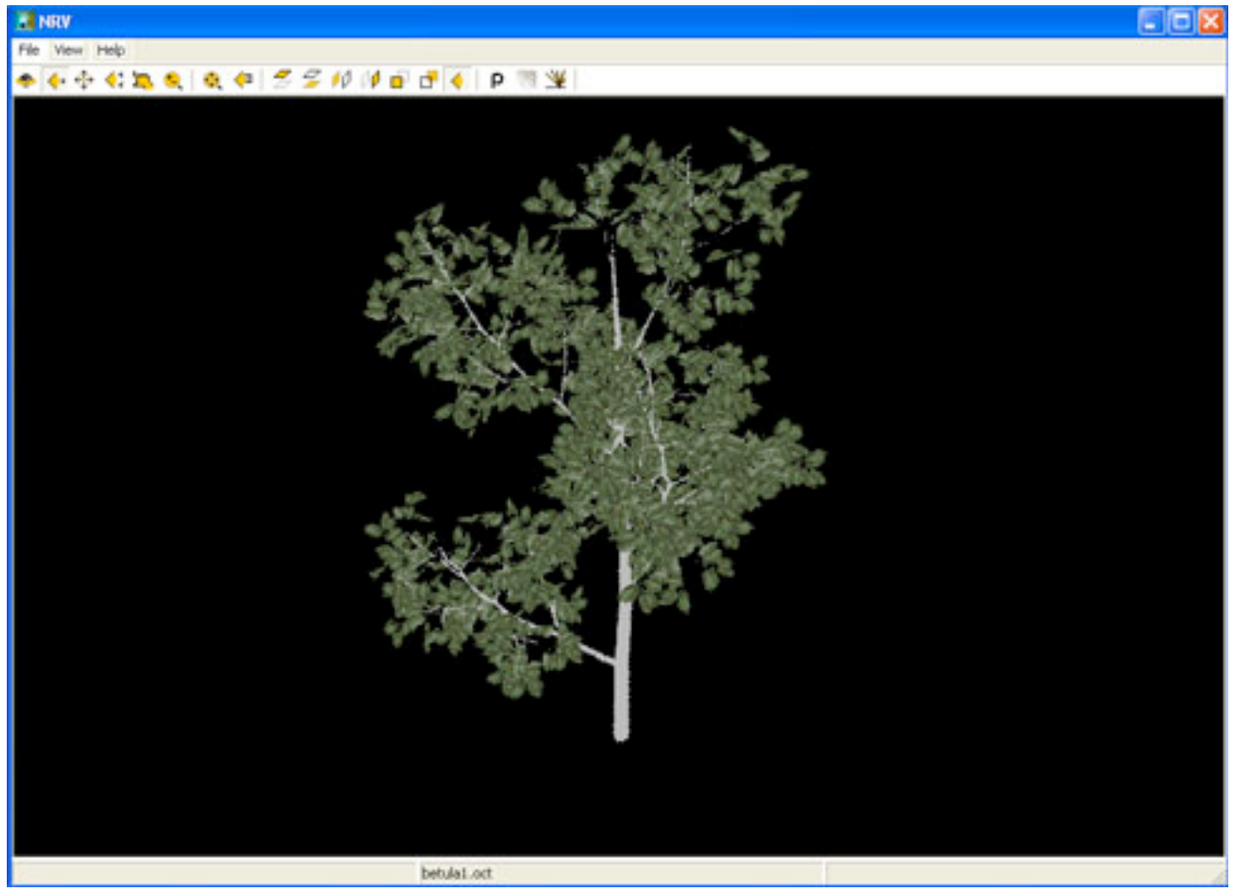
- `rcalc -e '$1=$1'` translates to `rcalc -e "$1=$1"`

- Piping *stdout* of a programme to *stdin* of another seems to be slower than saving *stdout* to a file and then sending this file to the next programme

Interactive viewer

winrview - Desktop Radiance

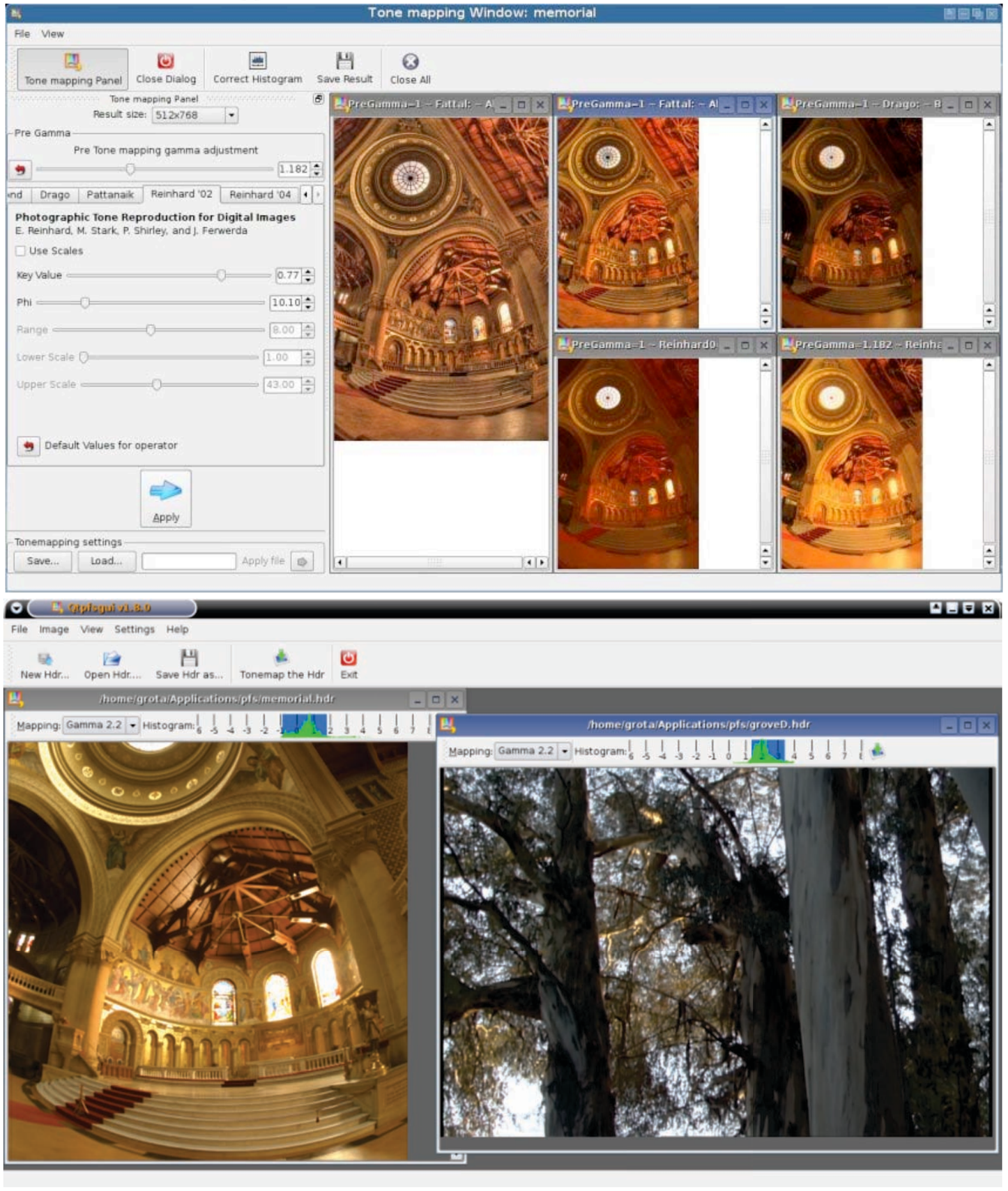
http://radsite.lbl.gov/deskrad/



nrv

http://www.aisarquitectura.com/nrv/

HDR image viewer



QTpfsGUI

http://qtpfsgui.sourceforge.net/

MinGW Radiance To Do

- Provide scripts (perhaps as compiled binaries)
  - Re-write them in Python?
- Compile X11 programmes
  - ideally with Windows native widgets (major GUI rewrite required), or ...
  - ... using Xming or Interix infrastructure (X11 server required), or ...
  - writing totally new applications ...

Radiance scripts example: radfalsecolor


```
Francesco-Anselmo-Laptop:radtools francescoanselmo$ python radfalsecolor.py --help
Usage: radfalsecolor.py [options]

Options:
  -h, --help            show this help message and exit
  -i INPUT, --input=INPUT
                        input picture filename for reading data
  -p PICTURE, --picture=PICTURE
                        background picture filename to use for contour bands
                        or contour lines
  -o OUTPUT, --output=OUTPUT
                        output picture filename; if omitted, the program will
                        add a _fc suffix to the input filename
  --contour-bands       display contour bands
  --contour-lines       display contour lines
  -e, --show-extrema    display pixel extrema
  -s SCALE, --scale=SCALE
                        maximum value for scale; set to "a" to account for all
                        values in the picture
  -l LABEL, --label=LABEL
                        text for label
  --legend-width=LEGEND_WIDTH
                        legend width (number of points)
  --legend-height=LEGEND_HEIGHT
                        legend height (number of points)
  -n DIVISIONS, --number-divisions=DIVISIONS
                        number of divisions
  --log=LOG_DECADES     number decades below the maximum value for logarithmic
                        scale
  -m MULTIPLIER, --multiplier=MULTIPLIER
                        multiplier (standard multiplier is 179 lm/W)
  -r RED, --red=RED     mapping expression of values to colors for red channel
  -g GREEN, --green=GREEN
                        mapping expression of values to colors for green
                        channel
  -b BLUE, --blue=BLUE  mapping expression of values to colors for blue
                        channel
  --spec                use old false color mapping
```

Volunteers?

Ideas for further Radiance development

1. Translate Radiance shell scripts to Python, to make them portable across different OS platforms:
   - I. **addfclegend** (Add a falsecolor legend to a RADIANCE picture - from RADZILLA)
   - II. **compamb** (Compute best ambient value for a scene and append to rad input file)
   - III. **dayfact** (Interactive script to calculate daylight factors)
   - IV. **debugcal** (Script to debug cal files for Radiance)
   - V. **falsecolor** (Create false color image with legend)
   - VI. **fieldcomb** (Combine alternate lines in full frames for field rendering)
   - VII. **genambpos** (Mark ambient locations)
   - VIII. **glare** (Interactive program for calculating glare values)
   - IX. **glaze** (Complex glazing model - goes with glaze1.cal and glaze2.cal)
   - X. **objline** (Create four standard views of scene and present as line drawings)
   - XI. **objpict** (Make a nice multi-view picture of an object)
   - XII. **objview** (Make a nice view of an object)
   - XIII. **optics2rad** (Convert Optics 5 output to correct Radiance input)
   - XIV. **pbilat** (Bilateral Filter)
   - XV. **pdelta** (Compute 1976 CIE Lab deltaE* between two Radiance pictures)
   - XVI. **pdfblur** (Generate views for depth-of-field blurring on picture)
   - XVII. **pgblur** (Apply Gaussian blur without resizing image)
   - XVIII. **phisto** (Compute foveal histogram for picture set)
   - XIX. **pmblur** (Generate views for motion blurring on picture)
   - XX. **pmdblur** (Generate views for motion and depth blurring on picture)
   - XXI. **raddepend** (Find scene dependencies in this directory)
   - XXII. **ran2tiff** (Convert Radiance animation frames to TIFF output)
   - XXIII. **ra_pfm** (Convert to/from Poskanzer Float Map image format using pvalue)
   - XXIV. **rlux** (Compute illuminance from ray origin and direction)
   - XXV. **vinfo** (Edit information header in Radiance file)
   - XXVI. **xyzimage** (Display one or more CIE XYZE pictures using ximage)
2. Advanced image viewer, with support for comments and "pcomb sliders"
3. Multiplatform `rvu`, with OpenGL support
4. Python library for lighting and daylighting


Radiance
Knowledge
Base
Search
You are here: start
Home
Radiance Knowledge Base
1. Installation and Set-up
I. Standard (UNIX) distribution
II. Radzilla
III. Windows alternatives
IV. Other packages
2. Tutorials
I. Basic
II. Advanced
III. Electric Lighting
IV. Daylighting
3. Input
I. CAD links and conversion tools
II. Photometry & Sky models
III. Materials
IV. Generator programs
V. Scene processing (xform, replmarks, oconv, etc.)
4. Computation
I. Rendering (batch & interactive)
II. Illumination calculations
III. Executive programs (rad, rayfront, etc.)
IV. Limitations of the Radiance calculation methods
5. Output
I. Image processing
II. Image viewing
III. Image format conversion
IV. Graphing & plotting
6. Radiance based software
7. Ideas for further development
8. Other on-line Radiance resources
3ds 3ds2mgf autocad blender cad converters cygwin daylighting deb debian desktop_radiance exporters gensky installation linux macosx mingw radout rpm source torad unix windows
start.txt · Last modified: 2008/08/17 10:18 by pisuke
Recent changes · Show pagesource · Login

Thank you!