# GUI Development with R-wxPython and BoaConstructor

James Wettenhall[1]
The Walter and Eliza Hall Institute of Medical Research
Melbourne, Australia

**Abstract for the useR! 2004 Conference in Vienna, Austria**

Despite the enormous success of the *tcltk* package for *R* in building Graphical User Interfaces (GUIs) for statistical software, many software developers have expressed interest in accessing other GUI toolkits from within *R*, particularly open-source platform-independent toolkits. The *wxWidgets* GUI toolkit (formerly *wxWindows*) is an object-oriented open-source cross-platform GUI toolkit implemented in C++. It offers many widgets and extension packages not easily available in *Tcl/Tk*, and has a more native look and feel, especially on *Windows*.

While it is possible to port a C++ GUI library to an interpreted language (and it has been done before with *GCC-XML and CableSwig* for the *ITK* library), this would take a lot of work, because C++ compilers and their associated Integrated Development Environments (IDEs) vary between different operating systems. Generally it is preferable to build R and its packages from the command-line, whereas *wxWidgets* currently requires the use of an IDE (e.g. Visual C++, Borland C++, or Dev-C++).

The approach taken here is to use the *Python* bindings for *wxWidgets*, as implemented in *wxPython*, and to access these bindings from *R* by going through the *RSPython* package from the *Omegahat* project. There is a wealth of *wxPython* experience amongst the open-source community which could be leveraged, and *Python* offers the sophisticated *BoaConstructor* IDE, which can be used as a *wxPython* GUI builder, and allows editing of dialogs by dragging and dropping widgets. A *Python* dialog class can then be imported into *R*, using the new *R-wxPython* package, and the dialog events (e.g. button clicks) can be mapped to R functions.

The *R-wxPython* package is still in the prototyping stage, but initial investigations suggest that it could soon be a viable alternative to the *R-Tcl/Tk* interface, for building GUIs in *R*. An effort is being made to develop easy-to-use wrapper functions for access to common widgets (e.g. buttons and text labels), and common dialogs (e.g. message boxes), but a more general approach is also being implemented, to provide developers with access to as many widgets and commands as possible, in a similar fashion to the *tkcmd* and *tkwidget* functions in the *tcltk* package.

---

[1] Correspondence should be addressed to wettenhall@wehi.edu.au. The work described here was supported by an NHMRC Transitional Institute Grant.