

The Wegina Wexx Intuh-pweter

Patrick TJ McPhee (ptjm@interlog.com)

Version 1.0.0, 30 March 2003

Contents

1	Introduction	1
1.1	Installation	1
1.2	Uninstalling	1
1.3	Reporting Bugs	2
1.4	Licencing	2
2	Starting the interpreter	2
3	Loading Libraries	3
4	Say and Pull	3
5	Tracing	4

1 Introduction

This paper describes a simple Windows front-end to the Regina Rexx interpreter which allows rexx scripts to run without a console window. This can be useful for running scripts in the background, or for running programs which use a windowing library such as Rexx/TK.

In its current form, wegina simply invokes the rexx interpreter on the script given on the command-line and executes it. If the script turns on tracing or attempts to communicate with the user using the 'say' or 'pull' instructions, dialog windows are displayed. Otherwise, the interpreter is quiet. None of regina's command-line switches are supported, and there is no way to start tokenized programs.

I have no definite plans to do further work on this program, however I will entertain suggestions and respond to bug reports. I'm releasing it today because I noticed there was an outstanding request for such a program in the Regina bug queue, and it occurred to me that it was stupid to have one and not let anyone know. I hope that it's useful.

My apologies to people who suffer from rhoticism.

1.1 Installation

The wegina package includes source code and a pre-compiled program. The distribution does not include an installation program. To install the pre-compiled program, extract wegina.exe from the installation zip file and copy it to either a directory in your program search path.

From this package, only wegina.exe is required to use the library, and applications which use it can be distributed with only this file. The DLL version of the Regina interpreter is also required, and it must be called regina.dll unless you rebuild the wegina executable.

You may want to name scripts which use wegina with the extension '.wex', and associate wegina with this extension. Possible commands are:

```
assoc .wex=WexxScwipt
ftype WexxScwipt=c:\ptjm\bin\wegin.exe %1 "%*"
```

If you also add '.wex' to the list of extensions in the PATHEXT environment variable, then scripts with that extension will run almost exactly as if they were executable programs.

1.2 Uninstalling

The distribution does not include an uninstall program. Some people may find this irritating, but what I find irritating is applications that plunk down 2 Mb of programs and 10 Mb of uninstall data. In any case, uninstall is pretty simple and it's left to the user. There are two steps:

1. Delete the files from the distribution;
2. Delete the Registry key HKEY_CURRENT_USER\Software\ptjm\Wegina.

1.3 Reporting Bugs

Theorem A: Every program can be reduced by at least one line.

Theorem B: Every program contains at least one bug.

Corollary: Every program can be reduced to one line which doesn't work correctly.

Wegina is very simple and defers most of its operations to the Regina interpreter, so there shouldn't be any significant bugs. If you do find a bug, an error in the documentation, or you simply have a suggestion for improving the distribution, please send me details at ptjm@interlog.com. It's useful to know the operating system you're using, the version of Regina, the version of regutil, and to have a set of steps for reproducing the bug.

1.4 Licencing

Wegina is distributed free of charge in the hopes that it will be useful, but without any warranty. This version is distributed under the terms of the Mozilla Public License. The precise details of the licence are found in the file MPL-1.0.txt in the distribution.

If you use the library purely as distributed by me, then you can cheerfully ignore licencing issues. If you modify the source code or adopt portions of it in your own programs or libraries, you should be aware of and fulfill your obligations under the licence. I believe that the restrictions placed by the Mozilla licence are less onerous than the ones in the GNU Library licence, and they are more in the spirit in which I would like my work to be distributed.

Although there are no obligations or restrictions related to use of the program, I would prefer that you do not use wegina in applications which cause injury or hardship to others. Also, if you derive a significant monetary benefit from the use of wegina, please share a portion with someone less fortunate. For instance, if you save \$10,000 by implementing an application with Regina and wegina, rather than buying a commercial Rexx interpreter, you could give \$1,000 to Unicef.

2 Starting the interpreter

To execute a program, x.rex, give the command

```
wegina x.rex arg1 arg2 ...
```

If the file's extension is '.rex' or '.wex', the extension can be left off.

For the command above to work, both wegina.exe and regina.dll must be in a directory which is listed in the PATH environment variable. You can also type the full path to wegina.exe provided regina.dll is either in a directory in the PATH or in the same directory as wegina.exe.

If you put spaces in the script name, or if you have spaces in the path containing the script or wegina.exe, the program is likely to fail.

3 Loading Libraries

I have not observed and do not anticipate any conflict between wegina and external function libraries. Since it's a simple shell around the true interpreter, there shouldn't be any difference in the execution of scripts versus executing the same script with regina.exe.

4 Say and Pull

Wegina handles 'say' and 'pull' instructions through the dialog shown in figure 1.

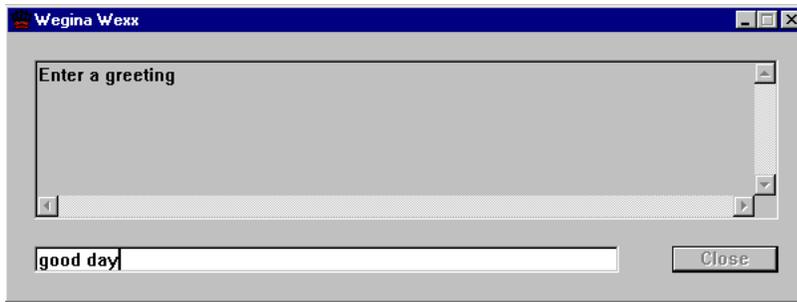


Figure 1: The say dialog

The dialog cannot be closed until the rexx program finishes running. After the script is finished running, the dialog will display until it is explicitly closed.

The dialog has two edit controls and a button. The large edit control with the scroll bars is used to display the output of 'say' instructions. The smaller edit control at the bottom is used to retrieve user input. You can cut, copy and paste with these controls either using the standard keys Ctrl-X, Ctrl-C, and Ctrl-V, respectively, or by right-clicking and using the appropriate context menu selections. The button is enabled only when the script has finished running and is used to dismiss the dialog.

When the 'pull' edit control is enabled, the rexx script is stopped waiting for input from the user. You ought to type something, then press 'Enter' to pass it to the script. The value entered is not echoed back to the screen, and it's cleared out of the edit box, so there's no way to perform a visual check to see what it was. This shouldn't be a problem, since this window is more of a debugging convenience than a user interface component, but I'm contemplating saving all or at least many previous pulled values in a pull-down box.

When you close the say dialog, its position on the screen is saved to the registry, in HKEY_CURRENT_USER\Software\ptjm\wegina\Position. The dialog should open in the same spot the next time wegina runs.

5 Tracing

Wegina handles tracing through the dialog shown in figure 2.

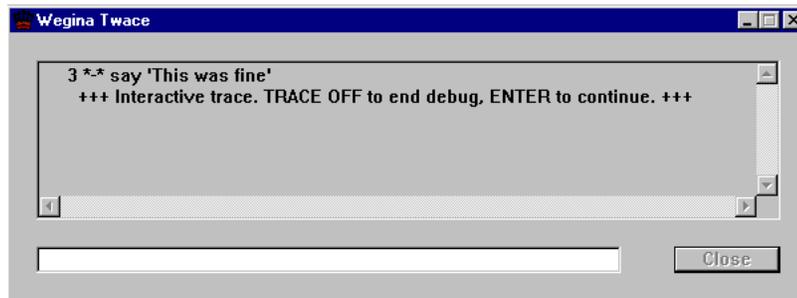


Figure 2: The trace dialog

The trace dialog operates in the same way as the say dialog: trace information is written to the large edit control, and interactive input is sought through the smaller one. Turn on tracing with the trace instruction. For instance, to trace statements interactively

```
trace '?r'
```

I've found that separating the trace window from the normal output window is quite convenient. Note, however, that if you print the value of an expression in an interactive trace, it will appear in the normal output window, which can be confusing the first few times, at least if you're used to it appearing on the next line of the screen.

In case you start tracing interactively without reading the interpreter's documentation, you can step to the next trace point by pressing enter, or stop interactive tracing by typing

```
trace 'o'
```

When you close the trace dialog, its position on the screen is saved to the registry, in `HKEY_CURRENT_USER\Software\ptjm\wegina\TwacePosition`. The dialog should open in the same spot the next time wegina runs.