ARexx Support for PasTeX

–Preliminary version–

Georg Heßmann
hessmann@unipas.fmi.uni-passau.de

Jörg Höhle
hoehle@cs.uni-sb.de
Joerg_Hoehle@saarag.zer

July 27, 1995

Abstract

ARexx is a powerful means of communication between processes on the Amiga Computer. ShowDVI features a very good ARexx interface, and a few ARexx macros have been written that support the communication between ShowDVI, the TeX compiler, `METAFONT`, the Shell and optionally the editors CygnusEd[1] and Micro(GNU)Emacs (MG)[2]. This paper tries to describe the installation and the use of these macros.

[1] CygnusEd is a product of ASDG.

[2] The latest release of MG, MG3β4 can be found on AmigaLibDisk352 of the Fred Fish collection

Contents

1 Copyright

All the macros have been put into the public domain by Georg Heßmann and Jörg Höhle.

We hope that users of PasTeX and ARexx will write macros for other editors and add new features to the existing ones. Bugs and suggestions are welcome!

2 Installation

Follows a short overview of the files that are part of the distribution:

- `doc/TeXRexx.tex` This document.
- `rexx/namestruc` This ARexx macro contains a function that returns the 3 main components of a filename, which are the device or volume name, any subdirectory, and the name withtout any extension. **This file** *must* **be placed in your `Rexx:` directory!**
- `rexx/TeX-Server.rexx` This is the main program that creates an ARexx port named `Start_TeX`, waits for an order to compile a file, then calls virtex to compile it.
- `rexx/TeXedit.rexx` This macro may be called by virtex or initex directly, or by the TeX-server in case of an unsuccessful compilation.
- `rexx/Start_TeX.#?` These are macros used to call the TeX-server. Currently available macros permit calls from within the CLI, ShowDVI, CED and MG.
- `rexx/Quit_TeX#?` These are macros used to terminate the TeX-server. The generic macro `rexx/Quit_TeX` may be called from any program. Specific macros are `rexx/Quit_TeX.ced` for the CED editor, and `rexx/Quit_TeX.sd` for ShowDVI.
- `rexx/#?ToFront` These are macros that may be used to bring some screen or window to the frontmost position, be it the CED, TeX-server or ShowDVI window or screen.
- `rexx/StartDVIprint.#?` These are simple macros that ask for which pages are to be printed and then calls DVIprint. The correct printer must already be defined in `ENV:DVIPRINT` (see the documentation on DVIprint), or the default preferences printer will be used.
- `rexx/NextTeXError.#?` These are very simple macros that may be helpfull when jumping from one error to the next one.

There are two ways to install all these macros correctly. An easy one, and one which does not require you to add lots of files to your single `rexx:` directory.

2.1 Easy installation

This method is not recommended. Move all the files in the `rexx/` directory into your REXX: directory.

2.2 Better installation

This other installation does not require you to put a dozen of files into your already overfull `rexx:` directory. *Only* the file `namestruc` need to be there. All the other files can reside in any directory, `TeX:rexx/` preferred. For this setup to work, you have to use full path names when calling the macros.

- TeX-server: Set the environment variable `ENV:TEXREXXEDIT` to the full path and name of the server macros, e.g. `"TeX:rexx/TeXedit.rexx %s %d"`.
- ShowDVI: Edit your `ShowDVI.config` file *before* you run ShowDVI, and change the line

  `arexx-start-script rx TeX-server.rexx` into

  `arexx-start-script rx TeX:rexx/TeX-Server.rexx`

  Change the entries for the function keys too, use full path names:

  ```
  f1 TeX:rexx/Start_TeX.sd
  f2 TeX:rexx/Start_TeX.sd ?
  f3 TeX:rexx/Quit_TeX
  f10 TeX:rexx/WBtoFront
  F2 TeX:rexx/Start_TeX.sd
  ```

- CLI: Either use `rx TeX:rexx/Start_TeX`, or add `TeX:rexx/` to your search path if you use csh or OS 2.0 and use `Start_TeX`
- CED: Install the commands with full pathes, like 'TeX:rexx/Start_TeX.ced' and 'TeX:rexx/Start_TeX.ced ?'
- MG: Install the commands with full pathes, like in the following example:

  ```
  ; Add some ESC-x functions
  ; this one will use the format in ENV:TEXFORMAT or ask for one
  fset tex-compile "\exrexx\nTeX:rexx/Start_TeX.mg\n"
  ; this one will ask for a format to be specified
  fset tex-compile-queryf "\exrexx\nTeX:rexx/Start_TeX.mg ?\n"
  ;fset tex-compile-latex "\exrexx\nTeX:rexx/Start_TeX.mg latexgde\n"

  ; demo other possibilities
  fset tex-view-to-front "\exrexx\nTeX:rexx/SDVItoFront.mg\n"
  fset tex-quit "\exrexx\nTeX:rexx/Quit_TeX\n"
  ```

That's all there is to do!

3 How to Use

Once you have correctly installed all the files, you sure would like to try them out. I'll explain it.

3.1 The TeX-server

First, you have to start the TeX-server. With OS2.0 you may use ShowDVI's `Shell commands -> ARexx TeX Shell` menu option. Otherwise open a new CLI window that will be reserved for interaction with the server. Under 1.3 don't use ShowDVI's menu for this, because ShowDVI is always waiting for this CLI to finish before responding to any other commands. then start the TeX-server ("`rx TeX:rexx/TeX-server`") in this CLI.

You will be prompted if you want to call the editor as soon as virtex encounters the first error. If you answer yes, virtex will call an editor through the `TeXedit.rexx` script as soon as the first error is found. Otherwise, you will be able to enter commands in virtex in the usual manner (you may still use the 'e' command to call the editor).

Then you will be prompted if you want to be asked for a format name whenever an application wants to compile a file using the default format. You are now going to ask "But what is this default format?". Here is the explanation: virtex and/or the TeX-server use the environment variable ENV:TEXFORMAT to store the name of the default format. Thus, if you call virtex from CLI without specifying a format, e.g. "virtex TeXRexx", then virtex will not use the plain format, as it always did before, but will use the format of name specified in ENV:TEXFORMAT.

When you call the TeX-server via some Start_TeX macro, you can specify a format name, or you can give no format name, or you can specify a format name of '?', in which case you will be prompted for the format to be used. If you called some Start_TeX macro without specifying a format name, and you started the TeX-server and answered "yes" to this second question, you will be prompted for a format name. If you answered "no", the default in `ENV:TEXFORMAT` will be used. If `ENV:TEXFORMAT` is not yet defined, you will be prompted too. Sounds complicated? Try it out!

3.2 ShowDVI

You have to edit `TeX:config/ShowDVI.config` (or the first `ShowDVI.config` in your `ENV:TEXCONFIG` path) manually, *before* you run ShowDVI. This is because once ShowDVI is run, it never re-reads its configuration file. This is an example of the end of your ShowDVI.config file:

```
f1 TeX:rexx/Start_TeX.sd
f2 TeX:rexx/Start_TeX.sd ?
```

```
f8 TeX:rexx/Quit_TeX
f9 TeX:Rexx/CEDtoFront
f10 TeX:rexx/WBtoFront
```

The function key f1 will compile the file currently shown using the format stored in ENV:TEXFORMAT (usually the last used). The function key f2 will prompt you for a format to be used, then compile. f8 will terminate the TeX-server. f9 may be used to pop the CygnusEd screen to the front. Same applies to Workbench with f10. Note that full pathnames are used: You are not required to put all these files in the `Rexx:` directory.

3.3 from the Shell

Calling the server differs from calling virtex directly in that you do not have all the options available that you have otherwise:

1. You cannot pass options like `-c, -d,` ... to virtex.
2. You *must* specify an existing filename. You cannot say "`Start_TeX testpage`" even though you now that testpage.tex is somewhere in your (LaTeX) include path. Start_TeX is looking for testpage in your current directory and will report an error if it cannot find it there.
3. You can, however, say `Start_TeX TeX:macros/latex/testpage`, as before.
4. You are not obliged to put a & in front of the format name. This is mostly usefull for many shells!
5. The server is called asynchronously, your CLI is again free for use, even if TeX has not finished compiling your file yet!

The general syntax is:
`[rx] Start_TeX [`$< formatname >$`] `$< filename >$
The same options relative to the format name are available than for ShowDVI: supply no format name, a format name or '?' as a format name. It works exactly the same.

The Start_TeX macro is not fully asynchronous, if you queue *several* request while the server is busy, you'll have to wait till your request is accepted. Don't queue several requests!

3.4 within MG

The small example file in the installation section shows you how to define some new commands to run the macros, and how you can bind some keys to these

newly created commands. You can load this file on demand (just use "ESC-x load $< return > < name - of - file >$"), or you can append it to your `s:mg-startup` file, which MG automatically reads upon startup.

MG compiles your current buffer. The file in this buffer must end in "tex".

The same options for starting the server are available than for ShowDVI: supply no format name, a format name or '?' as a format name. It works exactly the same way.

Note that the server is called fully asynchronously, thus you are still able to stay in your favourite editor while the server is compiling. If you queue several requests (don't do this!), i.e. you 'tex-compile' while the server is still running and busy with another file, you will get a short message in the status line when the server finally accepts your request.

3.5 `StartDVIprint` Macros

This is currently available only within ShowDVI. You can bind any function key to the `StartDVIprint.sd` macros in a manner described above. Then you'll be prompted for the first and the last page number to be printed, and if you want to use draft mode or not. Then DVIprint is called.

4 Enhancements

4.1 smallTeX or BigTeX?

You are not required to have virtex in the command search path. You can set the Rexx variable `virtex` (note the lowercase) to the full path of the virtex executable. If this variable is not set, it defaluts to `virtex`. This allows you to switch on the fly between then small and the big versions of TeX. For example, use "`rxset virtex TeX:bin/big/virtex`" to have the server use bigTeX.

4.2 asynchronous requests

All the Start_TeX macros call the server, freeing the caller as soon as the request is accepted. The caller *usually* does not have to wait until compilation finishes. This is only partially true. The server can only handle one request at a time. If you queue several requests while the server is busy, the `Start_TeX` macro and thus the caller will have to wait till compilation of the *previous* request finishes. Only MG is an exception to this rule, thanks to it's amazingly well-thought ARexx interface! Nice job, Mike!

5 Known bugs and features

- Simple NextError macros are supplied for editors. But they do no work correctly when your documents consist of multiple files, because the logfile is not analyzed deeply enough to determine the correct file.
- DME support: I'd like to add macros for Matthew Dillon's DME editor, however as far as I know, this editor is lacking one feature that the present macros require: DME cannot be called through ARexx from other sources. DME can call ARexx, but the reverse is not true. Thus I don't see how to write an appropriate `TeXedit.rexx` macro for DME. `Start_TeX.dme` would be no problem. You'll have to load the logfile yourself.

Please report all bugs, ideas, suggestions to:

Georg Heßmann
(hessmann@unipas.fmi.uni-passau.de)
or
Jörg Höhle
(hoehle@cs.uni-sb.de or Joerg_Hoehle@SAARAG.ZER)