

RRDIAGRAM User Guide

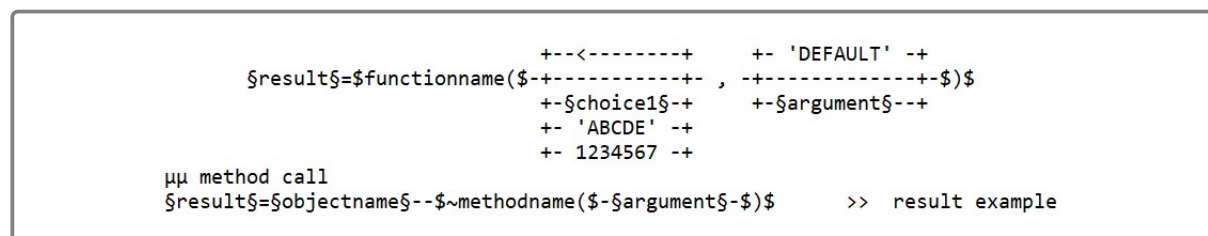
Jochem Peelen

The Rexx Language Association asked me for the program which created the syntax diagrams, presented by me at the Rexx Language Symposium of 2025. This is the guide to use it. It is written to run on Windows with ooRexx 5.0 or later installed. The freely downloadable software *GPL Ghostscript* is proposed for converting the created Postscript program (.eps) to PDF. This can then be imbedded as a syntax diagram in documents. 2025-06-16

Introduction to Workflow

Step 1: Create Diagram as ASCII File

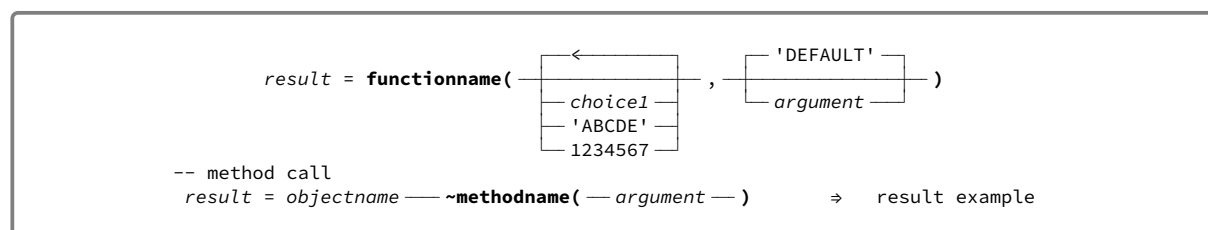
Open an editor, make sure it uses a **monospaced** font¹ and create a diagram like the following example:



Note the use of a blank to delimit text and lines; italic and bold text use § and \$ instead. For the – character, if it is not a part of a line, use µ. Double >> will appear as an arrow in the output. Save the file using extension **.rrd** in ANSI mode (that means no UTF characters allowed). The sample shown above accompanies this program and is named **rrdsample.rrd**.

Step 2: Convert to Postscript

Enter **rrdiagram rrdsample** as a console command. This will create **rrdsample.eps** in the same subdirectory. It contains Encapsulated Postscript source code for the following image:



The program appends to the ends of each line a horizontal bar of half width. This is not obvious, but improves appearance.

Step 3: Convert to PDF

Enter **rrdiagpdf rrdsample** as a console command. This will call *GPL Ghostscript* to convert the EPS file to a PDF file.

Use a PDF viewer to check the result. Note the width of the image for possible scaling when imbedding.

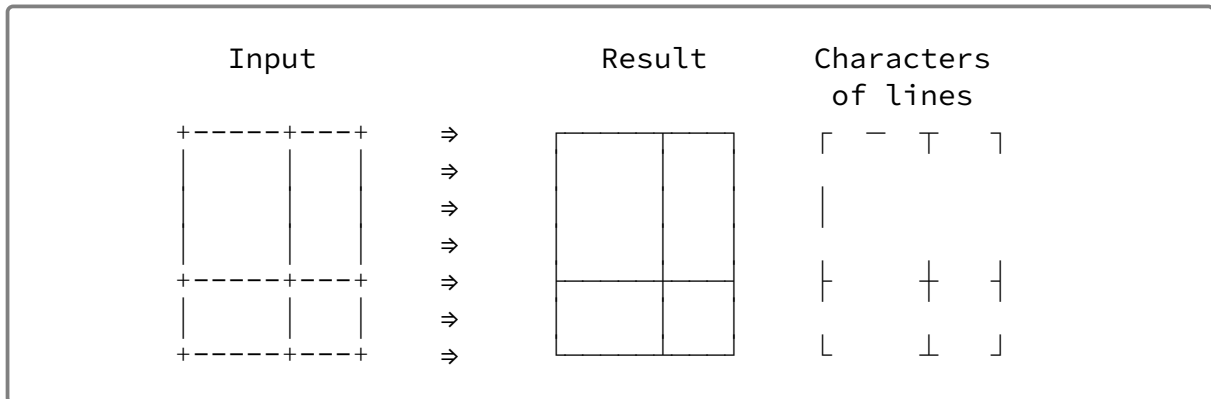
¹ For example *Consolas* in Notepad

Step 4: Embed PDF in your documentation

Embed the PDF file as an image in your documentation. To achieve the same font size in all syntax diagrams, scaling by a constant factor is recommended, 0.6 for example.

The above four steps are the way I created the around 200 diagrams in my *ooRexx Short Reference*.²

Algorithm



Basically, *rrdiagram* reads the ASCII characters on the left and converts them to lines as shown in the middle. The lines are not drawn, but made up from line drawing characters. These exist in the Adobe *SourceCodePro* font; actually its *light* cut is used.

The main work is going through all + characters and check its four neighbours. Depending on this, one of the line characters shown on the right is selected and put at the same position (line and column) in the output. This works because of the monospaced font and the absence of UTF control characters.

In addition, the contents is separated into four images, for the *light*, *bold*, *italic* and *regular* cuts of the font. By overprinting, these are merged into a single output image.

List of Included Files

`rrdiagram.pdf` this user guide

`rrdiagram.rex` translates diagram source (.rrd) to EPS

`rrdiagpdf.rex` calls *GPL Ghostscript* with necessary arguments to create PDF from EPS

`rrdsample.rrd` sample diagram source

`rrdsample.eps` sample diagram EPS

`rrdsample.pdf` sample diagram PDF

Installation of *GPL Ghostscript*

GPL Ghostscript is not a requirement. You can use any other graphics software that converts EPS to an image file, useable for inclusion in your documentation. In any case, you must make the fonts listed below available to your software.

From URL **ghostscript.com** download the „Ghostscript AGPL Release“ installer (**gs10051w64.exe** at the time of writing) and run it. The installation will use about 88 megabytes disk space and take only a few seconds.

After installation, edit file **rrdiagpdf.rex** and search for variable **gscall**. Compare it with the just created installation path and EXE file name on your computer. If necessary, modify variable **gscall** accordingly.

² PEELEN, Jochem: *Open Object Rexx 5.1 Classic Short Reference*. Wien: 36th International Rexx Language Symposium 2025.
German version: *Open Object Rexx 5.1 Kurzreferenz für Klassiker*.

Making the fonts accessible to GPL Ghostscript

Note that Ghostscript knows nothing about the way Windows handles its fonts. Create a directory `C:\Users\Public\rrdiag` for the font files.

- Go to github.com/adobe-font/source-code-pro
- Click on **OTF** to get a list of OTF files
- Download the following files to the **rrdiag** subdirectory:

SourceCodePro-Light.otf	SourceCodePro-It.otf
SourceCodePro-Bold.otf	SourceCodePro-Regular.otf

If you use a different font directory name, you must edit **rrdiagpdf.rex** and change variable **fontfiles** accordingly.

You are now ready to run **rrdiagpdf rrdsample**.

Do not be alarmed by two „Can't find“ messages *per font* – as long as the next line is a „Loading...“ message with the font name. That is the way Ghostscript works.

Program Code

I totally restructured my original program code and made use of many comments to make it possible for ooRexx programmers to adapt it to own needs. Particularly different code pages in non-English speaking European countries might require to change codepoints for some characters. The area marked DECLARATION BLOCK in the program code is for this purpose.

Windows command **chcp** shows you which codepage your command line window (console) uses. In German Windows 10 at the time of writing it is still 850. Most other software, for example Notepad (in ANSI mode) and the Hessling Editor since 4.0, use codepage 1252. On start, **rrdiagram.rex** queries the codepage. If 0850 is active, the console is switched to 1252, to bring it in line with *The Hessling Editor* I use on my German Windows.

Wikipedia is a good place to find codepage definitions. They show which characters are assigned to hexadecimal position 00 to FF. Before making changes to **rrdiagram.rex**, it is strongly recommended to have a close look at the codepage your computer uses.