Producing the ooRexx Documents

At the 2020 Symposium, I presented a talk about building the ooRexx documentation. Due to Internet connectivity issues, the talk was somewhat disjointed and I have therefore decided to summarize the points I made in this short paper.

Background

Shortly after the 2019 Symposium at the end of October 2019, a problem surfaced on the ooRexx developers list. The principle developers were experiencing problems building the documentation using the Publican tool chain which had been in use for many years. This seemed to be related to moving from Windows 7 to Windows 10. It was suggested that Pandoc might be a good replacement for Publican and I volunteered to investigate that possibility.

After a short while, it became apparent that Pandoc would not be a simple replacement for Publican as it could not directly produce PDFs. I decided to investigate other tool possibilities and starting learning about DocBook, the language used to write our documents, and the various tools that were available to process DocBook source files. I decided to investigate XSLTPROC as a starting point for several reasons - it was one of the oldest and therefore most stable transform tools and had a supported port for Windows even though it was originally written for Linux. There also seemed to be some evidence that Publican was using it under the covers.

Thus began a long journey of learning, detective work and trial and error that got us to the point that I could create PDFs of our documents without using Publican. Along the way I needed to become familiar with not only DocBook and XSLTPROC but also what I could learn about Publican from the tools we were using and the meager information still available on the Internet. It also involved XML, stylesheets and the languages used by them - XSLT and X-Path, DTDs, XML catalogs, fonts, CSS, FOP, Powershell and advanced batch scripting.

Once I could create PDFs, I worked on also creating HTML versions of our documents as well and, together with Rony Flatcher and P.O. Jonsson, ports for Linux and Mac of the tool chain. All of these tools were committed to SourceForge in June of 2020.

Today's talk will be more of a demonstration of how anyone, not just the ooRexx developers, can use these tools to build any or all of the ooRexx documents.

Get the document source files

In order to build the documents you need their source files which are stored on SourceForge just like the code source for the ooRexx programs. We are currently using Subversion, a software versioning and revision control system and you will therefore need a Subversion client in order to retrieve the files. This is referred to as checking out a working copy. While you could simply download the files, I do not recommend it as keeping up with changes to the files then becomes extremely difficult. A working copy on the other hand can be kept in sync by simply using the svn update command.

When checking out the files I recommend getting the entire tree under the trunk node. Individual documents are stored in their own directories but they all need to "include" some common files that are stored in their own directory. If you only get the directory for the document in which you are interested, you will not be able to get the common files during the build process.

You may place the working copy anywhere on your system that you desire. On my system I placed it in c:\Rexx\ooRexxDocs.

Get the document build tools

In addition to the source for the documents, you obviously need the tools to build them. The tools consist of two parts - the generic tools, like XSLTPROC and FOP, and the custom tools that are unique to building the ooRexx documents. The custom tools are also on SourceForge but you need not check them out separately as they are part of the docs working copy that you have already gotten. In addition to directories for each document and the common files directory, there is a directory named tools. In it you will find several directories containing the custom tools. The ones for Windows are in bldoc_win while those for Linux and Mac are in their own zip files. There will soon be another directory named bldoc_orx which contains a version of the tools I will be demonstrating today that are written in ooRexx and are intended to be usable on all operating systems.

Copy the appropriate set of tools to a directory of your choice on your system. This should be a directory that you will switch to when you wish to build a document and should therefore be one in which you can create files and not a system subdirectory. Once you have the custom tools, consult the documentation files - either read1st.txt on Windows or WhatIsHere.txt on the other platforms - for the procedure to get the generic tools. On Windows this means running setup.rex which will download each of the zip files needed, unzip them and make them available for use.

One of the generic tools is optional - the Liberation Fonts. Not installing them will not prevent you from building the documents but they will not look quite the same as those historically produced by Publican. I recommend installing them.

Once you have both the custom and generic tools in your working directory you are then able able to build any of the ooRexx documents.

Set the path to the source files for the tools

In order to build a document, the tool chain needs to know 1) where the source files are located, 2) which document you wish to build and 3) whether you want to create a PDF or a set of HTML files. The tool chain is designed to 'remember' what you are working on so you do not have to repeatedly enter the same information. In the original sets of tools, the information was saved in environment variables which only persisted for the length of the session. The new ooRexx-based tools save the information in a Properties object which is written to a file so the information persists across sessions. So you only need to 'tell' the tool chain the location of the source files once - unless you decide to move them. You do that by issuing the command 'docpath'. For my system this looks like:
docpath \rexx\oorexxdocs

Build a document!

I next demonstrated building the HTML files for the RxMath book. This involved entering the command:
doc2html rxmath
This produced a series of messages detailing what the tool chain was doing and, after 4 seconds, reported that 47 HTML files had been produced and Zipped up together with the CSS and image files that were needed. This Zip file could then be copied to another location if needed. I also entered the command:
html_folders\rxmath\index.html
which caused the first page of that book to be displayed in my browser and from which I could then navigate to any other part of the book.

I also demonstarted building the PDF version of the RxMath book by entering:
doc2pdf
This command used the same document - rxmath - as I did not specify the document name and produced a PDF in 3 seconds. Entering:
pdf_files\rxmath.pdf
resulted in Adobe Acrobat Reader displaying the document.

Finally I demonstarted building a PDF of one of our largest documents, the Open Object Rexx Reference - rexxref. This involved running:
doc2pdf rexxref
which produced a 759 page PDF in less than two minutes.

Gil Barmwater
gbarmwater@alum.rpi.edu