First steps

Louis Opter

January 12, 2012

Rathaxes is a Domain Specific Language to describes peripherals drivers. It compiles to kernel modules written in C for any defined Operating System.

This document explains how to setup the Rathaxes compiler on Windows and Unix like platforms and generate your first peripheral driver.

This document will also give you links to the complete documentation and explains how to build the latest version of Rathaxes.

Contents

1 Installation

An installer exists for Windows, on other operating systems only installations from the sources are supported at the moment. Of course, if you intend to develop on Rathaxes itself from Windows you can also choose to install Rathaxes from the sources (i.e: without the installer).

1.1 Pre-requisites

Rathaxes doesn't need to be installed on the same operating system as the operating system you target to generate drivers. In other words, you can install Rathaxes on Windows and generate the sources of a Linux driver (that is, from Windows).

We made it very easy to build Rathaxes source files from a CMake "Makefile", so you will need to install CMake (>= 2.6) along with Rathaxes to build our first example. CMake works on all the operating systems supported by Rathaxes and has a Windows installer[1].

CMake uses the native build tools for your operating system, that means that you will need to install "make" on an Unix-like (this often shipped in a "`build-essential`" package). And, that you will need to install the Windows SDK on Windows.

Finally, if you intend to build the source generated by Rathaxes you will also need the Linux sources on Linux, the Microsoft Windows Driver Development Kit on Windows and so on.

1.2 Using the Windows installer

An installer is available for Windows. It includes everything you need to write drivers using Rathaxes: the Rathaxes compiler and its documentation.

Download the installer, then simply execute it and follow the on-screen instructions.

1.3 From the sources on Unix

If you are not on Windows you will have to install Rathaxes from a "source release".

You will need to have CMake >= 2.6 installed and to download a Rathaxes source tarball.

Then extract the source tarball and "cd" into it, finally you can install Rathaxes using:

```
$ mkdir build
$ cd build
$ cmake -DCMAKE_INSTALL_PREFIX=/usr/local/ -DCMAKE_BUILD_TYPE=RELEASE ..
$ sudo make install
```

You will need to be root to issue "make install", this example uses "sudo" but you may need to adapt it to your local setup.

2 Generate your first driver

The process to generate a driver is roughly the same but some tools and paths are a bit different, so pick the instructions for your Operating System.

[1] When you install CMake on Windows, make sure to select the option to add CMake to the system `PATH` for all users.

2.1 From Windows

Open the "Rathaxes Shell". Then, "cd" somewhere you have read-write rights and generate the sources of our "LKM" example:

```
$ mkdir build
$ cd build
$ cmake "-GNMake Makefiles" %RATHAXES_PATH%/share/doc/samples/LKM/
$ nmake lkm_src
```

2.2 From any Unix like

Open your favorite terminal emulator and generate the sources of our "LKM" example:

```
$ mkdir build
$ cd build
$ cmake /usr/share/doc/rathaxes/samples/LKM/
$ make lkm_src
```

If you are on Linux and have a valid source tree you can also compile the kernel module you just generated:

```
$ make lkm
```

3 Diving in

Before we explain the compiler and the language in details, here is an overview of the generation of a driver:

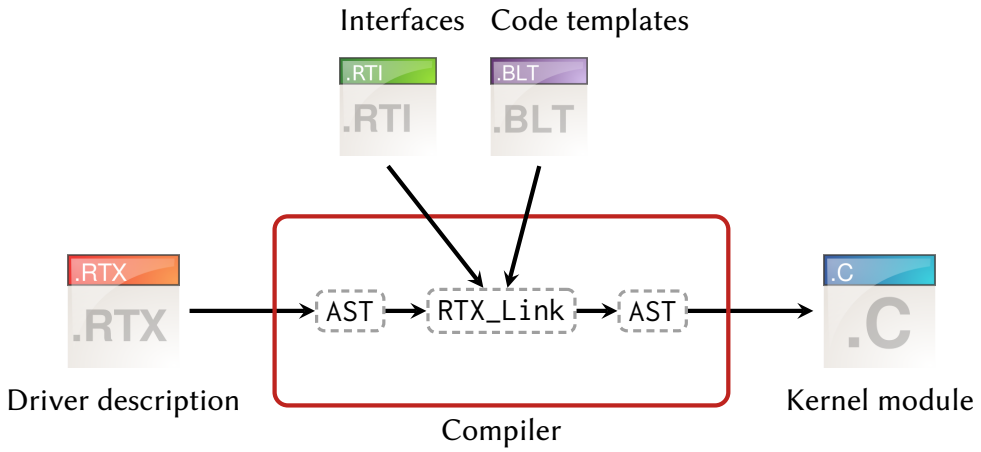

Figure 1: Overview of the generation of a driver

You can see that the compiler uses three types of files. These files all have different extensions but are actually written in the same language: Rathaxes. (Like in C, where `.c` and `.h` files have

different extensions but are written using the same language). These three "kind" of files have different purposes:

.rtx : describes your driver as well as its specific and platform-independent algorithms;

.rti : hold *interfaces* that describe what must be implemented in the `.rtx` file and what will be provided by the *templates* described in the `.blt` files;

.blt : hold *chunks* of C instrumented code that can be OS specific, the chunks are specified using informations written in the `.rtx` file and system introspection.

A Rathaxes project is compiled of one `.rtx` file and one or several `.rti` and `.blt` files. The complete documentation of the part of the language used to write `.rtx` files is available here: dsl_frontend_en.pdf. Another document covers the `.rti` and `.blt` files: dsl_backend_en.pdf.

4 Install the development version

You can install the latest version of Rathaxes if you need to have the latest bugs and features or if you want to contribute to the project. This involves checking out the current version of the project using Mercurial and to build it manually.

4.1 Pre-requisites

To checkout and build the project you need to install the following softwares:

- Mercurial >= 1.5 (you can check the version with "hg --version" and use TortoiseHg on Windows);
- Subversion (you need to install Slik SVN on Windows which ships the command line executables).

As well as the requirements to use Rathaxes (CMake and a compiler tool-chain). If you want to generate the documentation you will also need a LaTeX distribution (TeXLive >= 2009 or MikTeX on Windows) and Ghostscript.

4.2 Checkout the sources

Open a shell (from the Visual Studio menu on Windows), and checkout the project using:

```
$ hg clone https://rathaxes.googlecode.com/hg/ rathaxes
$ cd rathaxes
```

Keep the shell open, the next section explains how to build Rathaxes on Windows or Unix.

4.3 Build Rathaxes

If you are using an Unix like operating system use the following commands:

```
$ mkdir build
$ cd build
$ cmake ..
$ make
```

If you are on Windows use:

```
$ mkdir build
$ cd build
$ cmake "-GNMake Makefiles" ..
$ nmake
```

Documentation control

Group:	Rathaxes
Project name:	Rathaxes
Document type:	User Documentation
Version:	0.5
Document status:	Release

Table 1: Document informations

Version	Date	Author	Summary
0.1	04/07/2011	Louis Opter	Initial release
0.2	04/09/2011	Louis Opter	Lot of improvements
0.3	04/09/2011	David Pineau	Quick fixes
0.4	07/19/2011	Louis Opter	Add on overview of the generation of a driver
0.5	01/08/2012	Louis Opter	Update to cover the latest improvements

Table 2: Versions history

Contact

Code: http://code.google.com/p/rathaxes/

Mailing list: rathaxespublic@googlegroups.com