

niminst User's manual 0.9.2

Andreas Rumpf

May 21, 2013

Contents

1	Introduction	2
2	Configuration file	2
2.1	Project section	2
2.2	files key	2
2.3	Config section	2
2.4	Documentation section	3
2.5	Other section	3
2.6	Lib section	3
2.7	Windows section	3
2.8	UnixBin section	3
2.9	Unix section	3
2.10	InnoSetup section	4
2.11	C_Compiler section	4
3	Real world example	4

1 Introduction

niminst is a tool to generate an installer for a Nimrod program. Currently it can create an installer for Windows via Inno Setup as well as installation/deinstallation scripts for UNIX. Later versions will support Linux' package management systems.

niminst works by reading a configuration file that contains all the information that it needs to generate an installer for the different operating systems.

2 Configuration file

niminst uses the Nimrod parsecfg module to parse the configuration file. Here's an example of how the syntax looks like:

```
# This is a comment.
; this too.

[Common]
cc=gcc      # '=' and ':' are the same
--verbose

[Windows]
isConsoleApplication=False ; another comment

[Posix]
isConsoleApplication=True

key1: "in this string backslash escapes are interpreted\n"
key2: r"in this string not"
key3: """triple quotes strings
are also supported. They may span
multiple lines."""

--"long option with spaces": r"c:\myfiles\test.txt"
```

The value of a key-value pair can reference user-defined variables via the `$variable` notation: They can be defined in the command line with the `-var:name=value` switch. This is useful to not hard-coding the program's version number into the configuration file, for instance.

It follows a description of each possible section and how it affects the generated installers.

2.1 Project section

The project section gathers general information about your project. It must contain the following key-value pairs:

2.2 files key

Many sections support the `files` key. Listed filenames can be separated by semicolon or the `files` key can be repeated. Wildcards in filenames are supported. If it is a directory name, all files in the directory are used:

```
[Config]
Files: "configDir"
Files: "otherconfig/*.conf;otherconfig/*.cfg"
```

2.3 Config section

The `config` section currently only supports the `files` key. Listed files will be installed into the OS's configuration directory.

Key	description
Name	the project's name; this needs to be a single word
DisplayName	the project's long name; this can contain spaces. If not specified, this is the same as Name.
Version	the project's version
OS	the OSes to generate C code for; for example: "windows;linux;macosx"
CPU	the CPUs to generate C code for; for example: "i386;amd64;powerpc"
Authors	the project's authors
Description	the project's description
App	the application's type: "Console" or "GUI". If "Console", niminst generates a special batch file for Windows to open up the command line shell.
License	the filename of the application's license

Key	description
BinPath	paths to add to the Windows %PATH% environment variable. Example: BinPath: r"bin;dist\mingw\bin"
InnoSetup	boolean flag whether an Inno Setup installer should be generated for Windows. Example: InnoSetup: "Yes"

2.4 Documentation section

The documentation section supports the `files` key. Listed files will be installed into the OS's native documentation directory (which might be `$appdir/doc`).

There is a `start` key which determines whether the Windows installer generates a link to e.g. the `index.html` of your documentation.

2.5 Other section

The other section currently only supports the `files` key. Listed files will be installed into the application installation directory (`$appdir`).

2.6 Lib section

The `lib` section currently only supports the `files` key. Listed files will be installed into the OS's native library directory (which might be `$appdir/lib`).

2.7 Windows section

The windows section supports the `files` key for Windows specific files. Listed files will be installed into the application installation directory (`$appdir`).

Other possible options are:

2.8 UnixBin section

The `UnixBin` section currently only supports the `files` key. Listed files will be installed into the OS's native bin directory (e.g. `/usr/local/bin`). The exact location depends on the installation path the user specifies when running the `install.sh` script.

2.9 Unix section

Possible options are:

Key	description
InstallScript	boolean flag whether an installation shell script should be generated. Example: InstallScript: "Yes"
UninstallScript	boolean flag whether a deinstallation shell script should be generated. Example: UninstallScript: "Yes"

Key	description
path	Path to Inno Setup. Example: path = r"c:\inno setup 5\iscc.exe"
flags	Flags to pass to Inno Setup. Example: flags = "/Q"

2.10 InnoSetup section

Possible options are:

2.11 C_Compiler section

Possible options are:

3 Real world example

The installers for the Nimrod compiler itself are generated by niminst. Have a look at its configuration file:

```
[Project]
Name: "Nimrod"
Version: "$version"
; Windows and i386 must be first!
OS: "windows;linux;macosx;solaris;freebsd;netbsd;openbsd"
CPU: "i386;amd64;powerpc64;arm" # ;sparc
Authors: "Andreas Rumpf"
Description: ""This is the Nimrod Compiler. Nimrod is a new statically typed, imperative programming language, that supports procedural, functional, object oriented and generic programming styles while remaining simple and efficient. A special feature that Nimrod inherited from Lisp is that Nimrod's abstract syntax tree (AST) is part of the specification - this allows a powerful macro system which can be used to create domain specific languages."
```

Nimrod is a compiled, garbage-collected systems programming language which has an excellent productivity/performance ratio. Nimrod's design focuses on the 3E: efficiency, expressiveness, elegance (in the order of priority).""

```
App: Console
License: "copying.txt"
```

```
[Config]
Files: "config/nimrod.cfg"
Files: "config/nimdoc.cfg"
Files: "config/nimdoc.tex.cfg"
```

```
[Documentation]
```

Key	description
path	Path to the C compiler.
flags	Flags to pass to the C Compiler. Example: flags = "-w"

```

Files: "doc/*.txt"
Files: "doc/*.html"
Files: "doc/*.cfg"
Files: "doc/*.pdf"
Files: "doc/*.ini"
Start: "doc/overview.html"

[Other]
Files: "readme.txt;install.txt;contributors.txt;copying.txt"
Files: "configure;makefile"
Files: "*.ini"
Files: "koch.nim"

Files: "icons/nimrod.ico"
Files: "icons/nimrod.rc"
Files: "icons/nimrod.res"
Files: "icons/nimrod_icon.o"
Files: "icons/koch.ico"
Files: "icons/koch.rc"
Files: "icons/koch.res"
Files: "icons/koch_icon.o"

Files: "compiler/readme.txt"
Files: "compiler/nimrod.ini"
Files: "compiler/nimrod.cfg"
Files: "compiler/*.nim"
Files: "build/empty.txt"
Files: "bin/empty.txt"

[Lib]
Files: "lib/nimbase.h"
Files: "lib/copying.txt"
Files: "lib/*.nim"
Files: "lib/*.cfg"

Files: "lib/system/*.nim"
Files: "lib/core/*.nim"
Files: "lib/pure/*.nim"
Files: "lib/pure/collections/*.nim"
Files: "lib/impure/*.nim"
Files: "lib/wrappers/*.nim"

Files: "lib/wrappers/cairo/*.nim"
Files: "lib/wrappers/gtk/*.nim"
Files: "lib/wrappers/lua/*.nim"
Files: "lib/wrappers/openssl/*.nim"
Files: "lib/wrappers/readline/*.nim"
Files: "lib/wrappers/sdl/*.nim"
Files: "lib/wrappers/x11/*.nim"
Files: "lib/wrappers/zip/*.nim"
Files: "lib/wrappers/zip/libzip_all.c"

Files: "lib/windows/*.nim"
Files: "lib/posix/*.nim"
Files: "lib/js/*.nim"
Files: "lib/packages/docutils/*.nim"

[Other]
Files: "examples/*.nim"
Files: "examples/gtk/*.nim"
Files: "examples/0mq/*.nim"
Files: "examples/c++iface/*.nim"
Files: "examples/objciface/*.nim"
Files: "examples/cross_calculator/"

Files: "examples/*.html"
Files: "examples/*.txt"

```

```

Files: "examples/*.cfg"
Files: "examples/*.tmpl"

[Windows]
Files: "bin/nimrod.exe"
Files: "bin/c2nim.exe"
Files: "bin/niminst.exe"
Files: "bin/nimgrep.exe"

Files: "dist/*.dll"
Files: "koch.exe"
Files: "dist/mingw"
Files: "start.bat"
BinPath: r"bin;dist\mingw\bin;dist"
InnoSetup: "Yes"

[UnixBin]
Files: "bin/nimrod"

[Unix]
InstallScript: "yes"
UninstallScript: "yes"

[InnoSetup]
path = r"c:\Program Files (x86)\Inno Setup 5\iscc.exe"
flags = "/Q"

[C_Compiler]
path = r""
flags = "-w"

[deb]
buildDepends: "gcc (>= 4:4.3.2)"
pkgDepends: "gcc (>= 4:4.3.2)"
shortDesc: "The Nimrod Compiler"
licenses: "bin/nimrod,MIT;lib/*,MIT;"

```