niminst User's manual 0.14.3

Andreas Rumpf

June 29, 2016

Contents

1   Introduction

niminst is a tool to generate an installer for a Nim 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 Nim 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 Nim compiler itself are generated by niminst. Have a look at its configuration file:

```
; This config file holds configuration information about the Nim compiler
; and project.

[Project]
Name: "Nim"
Version: "$version"
Platforms: """
  windows: i386;amd64
  linux: i386;amd64;powerpc64;arm;sparc;mips;mipsel;powerpc;powerpc64el;arm64
  macosx: i386;amd64;powerpc64
  solaris: i386;amd64;sparc
  freebsd: i386;amd64
  netbsd: i386;amd64
  openbsd: i386;amd64
  haiku: i386;amd64
"""

Authors: "Andreas Rumpf"
Description: """This is the Nim Compiler. Nim 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 Nim inherited from Lisp is that Nim'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.

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

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

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

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

[Documentation]
; Files: "doc/*.html"
; Files: "doc/*.cfg"
; Files: "doc/*.pdf"
; Files: "doc/*.ini"
Files: "doc/overview.html"
Start: "doc/overview.html"


[Other]
Files: "readme.txt;install.txt;contributors.txt;copying.txt"
Files: "makefile"
Files: "koch.nim"
Files: "install_nimble.nims"

Files: "icons/nim.ico"
Files: "icons/nim.rc"
Files: "icons/nim.res"
Files: "icons/nim_icon.o"
Files: "icons/koch.ico"
Files: "icons/koch.rc"
Files: "icons/koch.res"
Files: "icons/koch_icon.o"

Files: "compiler"
Files: "doc"
Files: "tools"
Files: "web/website.ini"
Files: "web/ticker.html"
Files: "web/*.nim"
Files: "web/*.rst"
Files: "web/*.csv"
Files: "web/news/*.rst"
Files: "bin/nimblepkg/*.nim"
Files: "bin/nimblepkg/*.cfg"

[Lib]
Files: "lib"

[Other]
Files: "examples"
#Files: "dist/nimble"

Files: "tests"

[Windows]
Files: "bin/nim.exe"
Files: "bin/c2nim.exe"
Files: "bin/nimgrep.exe"
Files: "bin/nimsuggest.exe"
Files: "bin/nimble.exe"
Files: "bin/*.dll"

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

;             Section | dir | zipFile | size hint (in KB) | url | exe start menu entry
Download: r"Documentation|doc|docs.zip|13824|http://nim-lang.org/download/docs-${version}.zip|overview.html"
Download: r"C Compiler (MingW)|dist|mingw.zip|82944|http://nim-lang.org/download/${mingw}.zip"
```

```
Download: r"Support DLL's|bin|nim_dlls.zip|479|http://nim-lang.org/download/dlls.zip"
Download: r"Aporia IDE|dist|aporia.zip|97997|http://nim-lang.org/download/aporia-0.4.0.zip|aporia-0.4.0\bin\apori
; for now only NSIS supports optional downloads

[UnixBin]
Files: "bin/nim"


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


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

[NSIS]
path = r"c:\Program Files (x86)\NSIS\makensis.exe"
flags = "/V0"

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


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

[nimble]
pkgName: "compiler"
pkgFiles: "compiler/*;doc/basicopt.txt;doc/advopt.txt"
```