

# Nimrod Standard Library 0.9.2

Andreas Rumpf

May 21, 2013

## Contents

<b>1</b>	<b>Pure libraries</b>	<b>2</b>
1.1	Core . . . . .	2
1.2	Collections and algorithms . . . . .	2
1.3	String handling . . . . .	2
1.4	Generic Operating System Services . . . . .	3
1.5	Math libraries . . . . .	3
1.6	Internet Protocols and Support . . . . .	3
1.7	Parsers . . . . .	4
1.8	XML Processing . . . . .	4
1.9	Cryptography and Hashing . . . . .	4
1.10	Multimedia support . . . . .	5
1.11	Miscellaneous . . . . .	5
1.12	Database support . . . . .	5
1.13	Modules for JS backend . . . . .	5
<b>2</b>	<b>Impure libraries</b>	<b>5</b>
2.1	Regular expressions . . . . .	5
2.2	Database support . . . . .	5
2.3	Other . . . . .	5
<b>3</b>	<b>Wrappers</b>	<b>6</b>
3.1	Windows specific . . . . .	6
3.2	UNIX specific . . . . .	6
3.3	Regular expressions . . . . .	7
3.4	Graphics libraries . . . . .	7
3.5	GUI libraries . . . . .	7
3.6	Database support . . . . .	8
3.7	XML Processing . . . . .	8
3.8	Network Programming and Internet Protocols . . . . .	8
3.9	Scripting languages . . . . .	8
3.10	Data Compression and Archiving . . . . .	8
3.11	Scientific computing . . . . .	8

"The good thing about reinventing the wheel is that you can get a round one."

Though the Nimrod Standard Library is still evolving, it is already quite usable. It is divided into *pure libraries*, *impure libraries* and *wrappers*.

Pure libraries do not depend on any external `*.dll` or `lib*.so` binary while impure libraries do. A wrapper is an impure library that is a very low-level interface to a C library.

Read this document for a quick overview of the API design. If you can't find here some functionality you are looking for you could try using the 3rd party package manager Babel and its list of packages.

## 1 Pure libraries

### 1.1 Core

- `system` Basic procs and operators that every program needs. It also provides IO facilities for reading and writing text and binary files. It is imported implicitly by the compiler. Do not import it directly. It relies on compiler magic to work.
- `unsigned` This module implements basic arithmetic operators for unsigned integers. To discourage users from using unsigned integers, it's not part of `system`, but an extra import.
- `threads` Nimrod thread support. **Note:** This is part of the `system` module. Do not import it explicitly.
- `channels` Nimrod message passing support for threads. **Note:** This is part of the `system` module. Do not import it explicitly.
- `locks` Locks and condition variables for Nimrod.
- `macros` Contains the AST API and documentation of Nimrod for writing macros.
- `typeinfo` Provides (unsafe) access to Nimrod's run time type information.
- `actors` Actor support for Nimrod; implemented as a layer on top of the threads and channels modules.

### 1.2 Collections and algorithms

- `tables` Nimrod hash table support. Contains tables, ordered tables and count tables.
- `sets` Nimrod hash and bit set support.
- `lists` Nimrod linked list support. Contains singly and doubly linked lists and circular lists ("rings").
- `queues` Implementation of a queue. The underlying implementation uses a `seq`.
- `intsets` Efficient implementation of a set of ints as a sparse bit set.
- `critbits` This module implements a *crit bit tree* which is an efficient container for a set or a mapping of strings.
- `sequtils` This module implements operations for the built-in `seq` type which were inspired by functional programming languages.

### 1.3 String handling

- `strutils` This module contains common string handling operations like changing case of a string, splitting a string into substrings, searching for substrings, replacing substrings.
- `parseutils` This module contains helpers for parsing tokens, numbers, identifiers, etc.

- `strtabs` The `strtabs` module implements an efficient hash table that is a mapping from strings to strings. Supports a case-sensitive, case-insensitive and style-insensitive mode. An efficient string substitution operator `%` for the string table is also provided.
- `unicode` This module provides support to handle the Unicode UTF-8 encoding.
- `encodings` Converts between different character encodings. On UNIX, this uses the `iconv` library, on Windows the Windows API.
- `pegs` This module contains procedures and operators for handling PEGs.
- `ropes` This module contains support for a *rope* data type. Ropes can represent very long strings efficiently; especially concatenation is done in  $O(1)$  instead of  $O(n)$ .
- `unidecode` This module provides Unicode to ASCII transliterations: It finds the sequence of ASCII characters that is the closest approximation to the Unicode string.
- `matchers` This module contains various string matchers for email addresses, etc.
- `subexes` This module implements advanced string substitution operations.

## 1.4 Generic Operating System Services

- `os` Basic operating system facilities like retrieving environment variables, reading command line arguments, working with directories, running shell commands, etc.
- `osproc` Module for process communication beyond `os.execShellCmd`.
- `times` The `times` module contains basic support for working with time.
- `dynlib` This module implements the ability to access symbols from shared libraries.
- `streams` This module provides a stream interface and two implementations thereof: the *PFileStream* and the *PStringStream* which implement the stream interface for Nimrod file objects (*TFile*) and strings. Other modules may provide other implementations for this standard stream interface.
- `marshal` Contains procs for serialization and deserialization of arbitrary Nimrod data structures.
- `terminal` This module contains a few procedures to control the *terminal* (also called *console*). The implementation simply uses ANSI escape sequences and does not depend on any other module.
- `memfiles` This module provides support for memory mapped files (Posix's `mmap`) on the different operating systems.
- `fsmonitor` This module implements the ability to monitor a directory/file for changes using Posix's `inotify` API.

## 1.5 Math libraries

- `math` Mathematical operations like cosine, square root.
- `complex` This module implements complex numbers and their mathematical operations.

## 1.6 Internet Protocols and Support

- `cgi` This module implements helpers for CGI applications.
- `scgi` This module implements helpers for SCGI applications.
- `sockets` This module implements a simple portable type-safe sockets layer.
- `asyncio` This module implements an asynchronous event loop for sockets.
- `browsers` This module implements procs for opening URLs with the user's default browser.

- `httpserver` This module implements a simple HTTP server.
- `httpclient` This module implements a simple HTTP client.
- `smtp` This module implement a simple SMTP client.
- `irc` This module implements an asynchronous IRC client.
- `ftpcient` This module implements an FTP client.
- `cookies` This module contains helper procs for parsing and generating cookies.
- `mimetypes` This module implements a mimetypes database.
- `uri` This module provides functions for working with URIs.

## 1.7 Parsers

- `parseopt` The `parseopt` module implements a command line option parser. This supports long and short command options with optional values and command line arguments.
- `parsecfg` The `parsecfg` module implements a high performance configuration file parser. The configuration file's syntax is similar to the Windows `.ini` format, but much more powerful, as it is not a line based parser. String literals, raw string literals and triple quote string literals are supported as in the Nimrod programming language.
- `parsexml` The `parsexml` module implements a simple high performance XML/HTML parser. The only encoding that is supported is UTF-8. The parser has been designed to be somewhat error correcting, so that even some "wild HTML" found on the Web can be parsed with it.
- `parsecsv` The `parsecsv` module implements a simple high performance CSV parser.
- `parsesql` The `parsesql` module implements a simple high performance SQL parser.
- `json` High performance JSON parser.
- `lexbase` This is a low level module that implements an extremely efficient buffering scheme for lexers and parsers. This is used by the diverse parsing modules.

## 1.8 XML Processing

- `xmldom` This module implements the XML DOM Level 2.
- `xmldomparser` This module parses an XML Document into a XML DOM Document representation.
- `xmltree` A simple XML tree. More efficient and simpler than the DOM. It also contains a macro for XML/HTML code generation.
- `xmlparser` This module parses an XML document and creates its XML tree representation.
- `htmlparser` This module parses an HTML document and creates its XML tree representation.
- `htmlgen` This module implements a simple XML and HTML code generator. Each commonly used HTML tag has a corresponding macro that generates a string with its HTML representation.

## 1.9 Cryptography and Hashing

- `hashes` This module implements efficient computations of hash values for diverse Nimrod types.
- `md5` This module implements the MD5 checksum algorithm.
- `base64` This module implements a base64 encoder and decoder.

## 1.10 Multimedia support

- **colors** This module implements color handling for Nimrod. It is used by the `graphics` module.

## 1.11 Miscellaneous

- **events** This module implements an event system that is not dependant on external graphical toolkits.
- **oids** An OID is a global ID that consists of a timestamp, a unique counter and a random value. This combination should suffice to produce a globally distributed unique ID. This implementation was extracted from the `Mongodb` interface and it thus binary compatible with a Mongo OID.
- **endians** This module contains helpers that deal with different byte orders.

## 1.12 Database support

- **redis** This module implements a redis client. It allows you to connect to a redis-server instance, send commands and receive replies.

## 1.13 Modules for JS backend

- **dom** Declaration of the Document Object Model for the JS backend.

# 2 Impure libraries

## 2.1 Regular expressions

- **re** This module contains procedures and operators for handling regular expressions. The current implementation uses PCRE.

## 2.2 Database support

- **db\_postgres** A higher level PostgreSQL database wrapper. The same interface is implemented for other databases too.
- **db\_mysql** A higher level MySQL database wrapper. The same interface is implemented for other databases too.
- **db\_sqlite** A higher level SQLite database wrapper. The same interface is implemented for other databases too.
- **db\_mongo** A higher level **mongodb** wrapper.
- **mongodb** Lower level wrapper for the **mongodb** client C library.

## 2.3 Other

- **graphics** This module implements graphical output for Nimrod; the current implementation uses SDL but the interface is meant to support multiple backends some day.
- **dialogs** This module implements portable dialogs for Nimrod; the implementation builds on the GTK interface. On Windows, native dialogs are shown if appropriate.
- **zipfiles** This module implements a zip archive creator/reader/modifier.
- **web** This module contains simple high-level procedures for dealing with the Web like loading the contents of a Web page from an URL.
- **ssl** This module provides an easy to use sockets-style Nimrod interface to the OpenSSL library.

- `rdstdin` This module contains code for reading from `stdin`. On UNIX the GNU readline library is wrapped and set up.
- `zmq` Nimrod 0mq wrapper. This file contains the low level C wrappers as well as some higher level constructs.

## 3 Wrappers

The generated HTML for some of these wrappers is so huge that it is not contained in the distribution. You can then find them on the website.

### 3.1 Windows specific

- `windows` Contains a wrapper for the Win32 API.
- `ole2` Contains GUIDs for OLE2 automation support.
- `shellapi` Contains a wrapper for the `shellapi.h` header.
- `shfolder` Contains a wrapper for the `shfolder.h` header.
- `mmsystem` Contains a wrapper for the `mmsystem.h` header.
- `nb30` This module contains the definitions for portable NetBIOS 3.0 support.

### 3.2 UNIX specific

- `posix` Contains a wrapper for the POSIX standard.
- `cursorfont` Part of the wrapper for X11.
- `keysym` Part of the wrapper for X11.
- `x` Part of the wrapper for X11.
- `xatom` Part of the wrapper for X11.
- `xcms` Part of the wrapper for X11.
- `xf86dga` Part of the wrapper for X11.
- `xf86vmode` Part of the wrapper for X11.
- `xi` Part of the wrapper for X11.
- `xinerama` Part of the wrapper for X11.
- `xkb` Part of the wrapper for X11.
- `xkblib` Part of the wrapper for X11.
- `xlib` Part of the wrapper for X11.
- `xrandr` Part of the wrapper for X11.
- `xrender` Part of the wrapper for X11.
- `xresource` Part of the wrapper for X11.
- `xshm` Part of the wrapper for X11.
- `xutil` Part of the wrapper for X11.
- `xv` Part of the wrapper for X11.

- xvlib Part of the wrapper for X11.
- readline Part of the wrapper for the GNU readline library.
- history Part of the wrapper for the GNU readline library.
- rltypedefs Part of the wrapper for the GNU readline library.

### 3.3 Regular expressions

- pcre Wrapper for the PCRE library.
- tre Wrapper for the TRE library.

### 3.4 Graphics libraries

- cairo Wrapper for the cairo library.
- cairoft Wrapper for the cairoft library.
- cairowin32 Wrapper for the cairowin32 library.
- cairoxlib Wrapper for the cairoxlib library.
- sdl Part of the wrapper for SDL.
- sdl\_gfx Part of the wrapper for SDL.
- sdl\_image Part of the wrapper for SDL.
- sdl\_mixer Part of the wrapper for SDL.
- sdl\_net Part of the wrapper for SDL.
- sdl\_ttf Part of the wrapper for SDL.
- smpeg Part of the wrapper for SDL.
- gl Part of the wrapper for OpenGL.
- glext Part of the wrapper for OpenGL.
- glu Part of the wrapper for OpenGL.
- glut Part of the wrapper for OpenGL.
- glx Part of the wrapper for OpenGL.
- wingl Part of the wrapper for OpenGL.
- opengl New wrapper for OpenGL supporting up to version 4.2.

### 3.5 GUI libraries

- atk Wrapper for the atk library.
- gdk2 Wrapper for the gdk2 library.
- gdk2pixmap Wrapper for the gdk2pixmap library.
- gdkglext Wrapper for the gdkglext library.
- glib2 Wrapper for the glib2 library.
- gtk2 Wrapper for the gtk2 library.

- gtkglext Wrapper for the gtkglext library.
- gtkhtml Wrapper for the gtkhtml library.
- libglade2 Wrapper for the libglade2 library.
- pango Wrapper for the pango library.
- pangoutils Wrapper for the pangoutils library.
- iup Wrapper of the IUP GUI library.

### **3.6 Database support**

- postgres Contains a wrapper for the PostgreSQL API.
- mysql Contains a wrapper for the mySQL API.
- sqlite3 Contains a wrapper for SQLite 3 API.
- odbcsql interface to the ODBC driver.
- sphinx Nimrod wrapper for shpinx.

### **3.7 XML Processing**

- expat Wrapper of the expat XML parser.

### **3.8 Network Programming and Internet Protocols**

- libuv Wrapper for the libuv library used for async I/O programming.
- joyent\_http\_parser Wrapper for the joyent's high-performance HTTP parser.
- libcurl Wrapper for the libcurl library.
- openssl Wrapper for OpenSSL.

### **3.9 Scripting languages**

- lua Part of the wrapper for Lua.
- lualib Part of the wrapper for Lua.
- lauxlib Part of the wrapper for Lua.
- tcl Wrapper for the TCL programming language.
- python Wrapper for the Python programming language.

### **3.10 Data Compression and Archiving**

- zlib Wrapper for the zlib library.
- libzip Interface to the lib zip library by Dieter Baron and Thomas Klausner.

### **3.11 Scientific computing**

- libsvm Low level wrapper for lib svm.