



Technical Manual of the Universal Webserver

Organisation: Copyright (C) 2019-2020 Olivier Boudeville

Contact: about (dash) universal-webserver (at) esperide (dot) com

Creation date: Saturday, May 2, 2020

Lastly updated: Friday, May 15, 2020

Status: Work in progress

Version: 0.0.1

Dedication: Users and maintainers of the Universal Webserver.

Abstract: The [Universal Webserver](#), part of the [Universal Server](#) umbrella project, provides a multi-domain, multi-virtualhost webserver integrating various web-related services.

We present here a short overview of these services, to introduce them to newcomers.

The next level of information is to read the corresponding [source files](#), which are intensely commented and generally straightforward.

Table of Contents

Technical Manual of the Universal Webserver	1
Overview	3
Layer Stack	3
Licence	4
Current Stable Version & Download	4
Using Cutting-Edge GIT	4
Using OTP-Related Build/Runtime Conventions	5
Support	5
Please React!	5
Ending Word	5

Overview

We present here a short overview of the services offered by the *Universal Webserver*, to introduce them to newcomers.

The next level of information is to read the corresponding [source files](#), which are intensely commented and generally straightforward.

Layer Stack

From the highest level to the lowest, as summarised [here](#), a software stack involving the Universal Webserver usually is like:

- the *Universal Webserver* services themselves (i.e. this [us-web](#) layer)
- [Cowboy](#) (for a small, fast and modern web framework)
- [optional] [Awstats](#) (for the analysis of access log files)
- [US-Common](#) (for US base facilities)
- [Ceylan-Traces](#) (for advanced runtime traces)
- [Ceylan-WOOPER](#) (for OOP)
- [Ceylan-Myriad](#) (as an Erlang toolbox)
- [Erlang](#) (for the compiler and runtime)
- [GNU/Linux](#)

The shorthand for Universal Webserver is `uw`.

Licence

The `Universal Webserver` is licensed by its author (Olivier Boudeville) under the [GNU Affero General Public License](#) as published by the Free Software Foundation, either version 3 of this license, or (at your option) any later version.

This allows the use of the `Universal Webserver` code in a wide a variety of software projects, while still maintaining copyleft on this code, ensuring improvements are shared.

We hope indeed that enhancements will be back-contributed (ex: thanks to merge requests), so that everyone will be able to benefit from them.

Current Stable Version & Download

As mentioned, the single, direct prerequisites of the [Universal Webserver](#) are:

- [Cowboy](#)
- [Awstats](#) (as an optional, runtime-only dependency)
- [US-Common](#)

The latter relies on [Ceylan-Traces](#), which implies in turn [Ceylan-WOOPER](#), then [Ceylan-Myriad](#) and [Erlang](#).

We prefer using GNU/Linux, sticking to the latest stable release of Erlang, and building it from sources, thanks to GNU `make`.

We recommend indeed obtaining Erlang thanks to a manual installation (refer to the corresponding [Myriad prerequisite section](#) for more precise guidelines), Awstats thanks to your distribution of choice (ex: `pacman -S awstats`) and, for all Erlang-related software, to rely on `rebar3`.

One wanting to be able to operate on the source code of these dependencies may define appropriate symbolic links in a `_checkouts` directory created at the root one's `us-web` clone, these links pointing to relevant GIT repositories.

Using Cutting-Edge GIT

This is the installation method that we use and recommend; the `Universal Webserver` `master` branch is meant to stick to the latest stable version: we try to ensure that this main line always stays functional (sorry for the pun). Evolutions are to take place in feature branches and to be merged only when ready.

Once Erlang, Cowboy and possibly Awstats are available, it should be just a matter of executing:

```
$ git clone https://github.com/Olivier-Boudeville/Ceylan-Myriad myriad
$ cd myriad && make all && cd ..
```

```
$ git clone https://github.com/Olivier-Boudeville/Ceylan-WOOPER wooper
$ cd wooper && make all && cd ..
```

```
$ git clone https://github.com/Olivier-Boudeville/Ceylan-Traces traces
$ cd traces && make all && cd ..
```

```
$ git clone https://github.com/Olivier-Boudeville/us-common
$ cd us-common && make all

$ git clone https://github.com/Olivier-Boudeville/us-web
$ cd us-web && make all
```

Running a corresponding test just then boils down to:

```
$ make debug
```

Using OTP-Related Build/Runtime Conventions

As discussed in these sections of [Myriad](#), [WOOPER](#), [Traces](#) and [US-Common](#), we added the (optional) possibility of generating a Universal Webserver *OTP application* out of the build tree, ready to result directly in an (*OTP*) *release*. For that we rely on [rebar3](#), [relx](#) and [hex](#).

Then we benefit from a standalone, complete Universal Webserver able to host as many virtual hosts on any number of domains as needed.

As for Myriad, WOOPER, Traces and US-Common, most versions of the Universal Webserver are also published as [Hex packages](#).

For more details, one may have a look at [rebar.config.template](#), the general rebar configuration file used when generating the Universal Webserver OTP application and release (implying the automatic management of all its dependencies).

Support

Bugs, questions, remarks, patches, requests for enhancements, etc. are to be reported to the [project interface](#) (typically [issues](#)) or directly at the email address mentioned at the beginning of this document.

Please React!

If you have information more detailed or more recent than those presented in this document, if you noticed errors, neglects or points insufficiently discussed, drop us a line! (for that, follow the [Support](#) guidelines).

Ending Word

Have fun with the Universal Webserver!

Universal Webserver