

AdaWorks::Config Manual

Marcelo Coraça de Freitas

February 20, 2007

Contents

1	Introduction	5
2	Building	7
2.1	Dependencies	7
2.2	Building everything	8
2.3	Building only the Core Library	8
2.4	Building the text file support	8
2.4.1	Building and running the test	8
2.5	Building the XML file support	8
3	Instalation	9

Chapter 1

Introduction

AdaWorks::Config (aka Aw_Config and AwConfig) is a library designed to use of configuration files in the Ada programming language.

It was designed to fit the way the host Operating System works with file based configurations and, for this purpose, it uses the *AdaWorks::Libs* component (see section ??, page 7).

This component is released by the GNAT Modified GPL license (please see LICENSE file in the root directory) and is distributed with no warranty. Use it at your risk.

Chapter 2

Building

As default, no binary package is provided by the AdaWorks team. As this is an open sourced library there is no good reason in doing so. This chapter will enlighten you through the building process of the shared libs and example programs that ships with it.

When building this library from source you'll get new files in:

./obj where the object files are built. It's for internal use of the compiler

./lib where the shared libraries will be linked.

./bin where the example programs will be linked.

When using one of the commands described in the next sections, keep in mind the build system should handle the compilation of all dependencies within the AdaWorks project (including AdaWorks::Lib).

2.1 Dependencies

To build this library you'll need:

AdaWorks::Lib The source package should be unpacked in the same folder AdaWorks::Config was (there should be *awlib* and *awconfig* folders in the root).

GNAT 4.1 or greater The only supported compiler is GNAT-GCC 4.1.1 but it should work fine with GNAT-GPL and GNAT-PRO.

XML/Ada the command `xmlada-config` got to be in the path. *Note:* it's only required when using XML based configuration files (see following sections for more information).

GNU Make or other compatible make tool.

2.2 Building everything

This is the easiest and recommended way for new users. Just type:

```
$ make
```

In the root of this source distribution. This will build the core library, all the parsers and test applications.

2.3 Building only the Core Library

This is only useful when creating your own parser (section ??, page ??). For doing this just type:

```
make base
```

It will build the *libawconfig.so* file that should be available when running the final package.

2.4 Building the text file support

This is to whom wants to use the Text Parser shipped with this distribution. To do it simply type:

```
make text-parser
```

It will build *libawconfig.so* and *libawconfig-text_parser.so*.

2.4.1 Building and running the test

Type

```
make text-test
```

It will create the executable *./bin/text_test*. To run it you have two options. The easier one is typing:

```
make run-text
```

This will ensure the code is compiled and will set the required environment variable before running it.

The other way - the one you should use to run your own programs if you want to customize the location of your configuration files - is issuing:

```
TEST_CONFIG_PATH=./data ./bin/text_test
```

TEST_CONFIG_PATH is the name of the environment variable that should contain a list with the Configuration Path where to look for config files. It should be set as your system's PATH (ie, "SOME_FOLDER:OTHER_FOLDER" in unix systems).

2.5 Building the XML file support

Just proceed the same as described in the last section replacing "text" by "xml" when appropriated.

Chapter 3

Instalation

This target isn't ready yet but the installation procedure is quite simple.

- Copy the contents of `src/**/*.ads` to the system's "include" folder.
- Copy the contents of `lib/*.{ali,so,-VERSION_NUMBER}` to the "lib" folder

With this you should be able to compile fine using the `-L` and `-I` options in GNAT.