

Testing MMALPHA

Patrice Quinton

Version of April 10, 2010

Abstract

This document gives information on testing MMALPHA. Some other information is also available in `doc/Install`. Information regarding compiling `domlib` is given in `doc/Developers/Compiling`.

1 Right after installing MMAAlpha

MMALPHA provides testing facilities which are explained in this document. After installation, run `test1[]`. This checks various things, in particular, that the `domlib` is properly installed. The report of this test is written in file `MMALPHA/tests/testReportFile.txt`. If this test does not return the value `True`, please send a mail together with this file to

`patrice.quinton@irisa.fr`

Warning: Notice that executing `test1[]` erases the previous version of the test report file.

Do it yourself: You may also fix by yourself some common errors: report to Section .

Tested configurations The current distribution runs properly on a Mac-Book Pro, with MATHEMATICA version 5 and 6. Version 7.0 of MATHEMATICA is under test.

2 More tests

Other tests are done using the commands `test2[]`, `test3[]`, and `test4[]`. All these tests append information in the file `$MMALPHA/tests/testReportFile.txt`.

If one of these tests fails (i.e., it does not return the value `True`), please send a mail together with the test report file to

`patrice.quinton@irisa.fr`

If all these tests are successful, your installation should be correct.

3 Exploring tests

Commands `test1`, `test2`, etc. make use of another test program, which allows each individual package to be tested. For example,

```
test[ "Domlib" ]
```

runs a test for the `domlib` and returns `True` if the test was successful. This command writes information in the test report file. If one test fails, the name of this test is written in the test report file. Say test `"domlib 23"` fails, then you can replay it by evaluating :

```
test[ "Domlib" , "domlib 23"]
```

The nature of this test will then be displayed. Notice however, that running an individual test may not always provide the same result as when it is run in the context of a full package test. The reason is that some of the package test were written before this facility was offered. I am trying¹ to fix this, but test programs are numerous and long...

Test files are in directory `$MMALPHA/Tests`. Commands `test1` to `test4` are part of `$MMALPHA/lib/Packages/Alpha.m`.

4 Common Problems at Installation and How to Fix Them

(See also `doc/Install`).

Check this in order.

¹As of December 2008.

1. Is environment variable `$MMALPHA` set? If this is not the case, `MATHEMATICA` does not start `MMALPHA`. To check this, start `MMALPHA` and evaluate

```
Environment[ "MMALPHA" ]
```

If this variable was not set, then the result is `$Failed`, otherwise, it should be the path of the directory where you installed `MMALPHA`. To fix this problem, report to the installation manual. Remember on `MacOSX` that you must start `MATHEMATICA` from a shell window, and not directly by double-clicking on the `MATHEMATICA` icon: in this case, `MATHEMATICA` does not import the environment variables.

2. Does the environment variable `$PATH` contain the directory where the binaries of `MMALPHA` are ? If this is not the case, `domlib` cannot be loaded. Check this by evaluating

```
$Path
```

and checking that there appears a path of the form `"... /mmalpha/bin.ostype"` where `ostype` depends on your operating system. For `MacOSX`, it should be `bin.darwin`, for `Linux`, it should be `bin.linux`, etc. To fix this, report to the installation manual and set the environment variable. You can get the value of your OS type by evaluating `Environment["OSTYPE"]` in `MATHEMATICA`.

3. Does the directory of the binaries exist? To check this, get the value of environment variable `$OSTYPE`, and check that there exist a directory `bin.ostype` in your `MMALPHA` repository. If such a directory does not exist, the problem has to be reported to the author of this note...
4. Even if the `bin.ostype` directory exists, it may not contain the proper binary files: `domlib`, etc.