GetPot Example

April 5, 2016

GetPot

Intro to

GetPot

Web page

- http://getpot.sourceforge.net/

GetPot

GetPot

GetPot *header file only* library, to facilitate command line and config file parsing. Useful for changing algorithm parameters without recompiling etc. . .

GetPot provides a class to parse `argc` and `argv` in alternative to POSIX standard getopt (in C).

GetPot

Passing parameters on the command line directly

```
#include <iostream>
#include "GetPot"

int main ( int argc, char* argv[] )
{
    // Read, using GetPot, the data from input
    GetPot commandLine ( argc, argv );
    double a = commandLine ( "a", 0. );
    double b = commandLine ( "b", 1. );
    int nint = commandLine ( "nint", 10 );

    std::cout << a << " " << b << " " << nint << std::endl;
    return 0;
}
```

Parameters are read via the `()` operator for the `GetPot` class. It requires the name of the parameter to be read and the default value. The type is deduced from the class of the default value.

GetPot

Run

```
./main a=10. b=70. nint=100
./main nitn=100 b=70. a=10.
./main a=10. b=70.
./main
```

Sorting of command line arguments is unimportant

GetPot

Configuration files

```
#                    -*- GetPot -*-
#-------------------------------------------------
#           Data file for integration
#-------------------------------------------------
[integration]

    [./domain]
    a = 1.
    b = 4.
    [../]

    [./mesh]
    nint = 20
    [../]

[../]
```

GetPot

To parse the file in C++ code

```
#include <iostream>
#include "GetPot"

int main()
{
    // Read, using GetPot, the data from input
    GetPot fileData("data");

    double a = fileData ( "integration/domain/a", 0.);
    double b = fileData ( "integration/domain/b", 1.);
    int nint = fileData ( "integration/mesh/nint", 10);

    std::cout << a << " " << b << " " << nint << std::endl;
    return 0;
}
```

GetPot

To parse the file in C++ code

```
#include <iostream>
#include <string>
#include "GetPot"

int main()
{
    // Read, using GetPot, the data from input
    GetPot fileData ("data");

    const std::string globalsection = "integration/";
    const std::string section1 = globalsection + "domain/";
    const std::string section2 = globalsection + "mesh/";

    double a = fileData ( (section1 + "a").data(), 0. );
    double b = fileData ( (section1 + "b").data(), 1. );
    int nint = fileData ( (section2 + "nint").data(), 10 );

    std::cout << a << " " << b << " " << nint << std::endl;
    return 0;
}
```

GetPot

Esecuzione

```
./main
```

GetPot

Perché non entrambi?

The name of the config file may be passed on the command line

GetPot

Command line and config file

```
#include <iostream>
#include <string>
#include "GetPot"

int main (int argc, char* argv[])
{
    GetPot commandLine ( argc, argv );
    const std::string fileName =
            commandLine.follow ( "data", 2, "-f", "--file" );
    GetPot fileData ( fileName.c_str() );

    const std::string globalsection = "integration/";
    const std::string section1 = globalsection + "domain/";
    const std::string section2 = globalsection + "mesh/";

    double a = fileData ( (section1 + "a").data(), 0. );
    double b = fileData ( (section1 + "b").data(), 1. );
    int nint = fileData ( (section2 + "nint").data(), 10 );

    std::cout << a << " " << b << " " << nint << std::endl;
    return 0;
}
```

Run

```
./main -f data1
./main --file data1
./main
```