Pessum Documentation

Arden Rasmussen

24/02/2017

Contents

1   Functions

1.1 General

1.1.1 InitializePessum

```cpp
void pessum::InitializePessum(bool devmode, bool logtimes);
```

devmode Sets the log to only output for Fatal, Error, and Warnings.

logtimes Sets the log to include the current time in hh:mm:ss to every log entry.

Initializes Pessum and any sub-components that need initialization. This function will open the log file. **Note:** This funciton must called before any other Pessum functions.

```cpp
//Initialize pessum in development mode, and with logging of times
pessum::InitializePessum(true, true);
```

1.1.2 TerminatePessum

```
1    void pessum :: TerminatePessum ( ) ;
```

Terminates Pessum and any sub-components that need termination. This closes the log file and saves it. **Note:** This funciton should be the last of pessum functions to be called.

```
1    //Terminate pessum and all sub namespaces
2    pessum :: TerminatePessum ( ) ;
```

1.2   Logging

1.2.1   InitializeLogging

```
void pessum::logging::InitializeLogging(std::string outputfile, bool
    recordtime, bool dev);
```

outputfile   File name to save output log to.

recordtime   Determines if the time in hh:mm:ss is recored with each log entry.

dev   Determines if the log is to run in development mode.

This funciton creates a logfile of the name *outputfile*, which will be used for all logging output. If *recordtime* is *true* the time is included in the log entry. If *recordtime* is *false* then the time is not included. If *dev* is *true* then all log entries are recored to the file. If *dev* is *false* then only FATAL, ERROR, and WARNING log entries are recored to the file. **Note:** This function must be called before any other logging funcitons to work.

```
//Create a log file with the name "log_output.log", that will include times
    of each entry, and runs in development mode
pessum::logging::InitializeLogging("log_output.log", true, true);
```

1.2.2 TerminateLogging

```
1    void pessum :: logging :: TerminateLogging ( ) ;
```

This funcition closes and saves the log file. **Note:** After this funciton is called no other logging funciton will work until InitializeLogging is called again.

```
1    // Terminate logging , and save the log file
2    pessum :: logging :: TerminateLogging ( ) ;
```

1.2.3 AddLogLocation

```
1    int pessum::logging::AddLogLocation(std::string locationstring);
```

locationstring Name of location that is to be saved for logging.

This function adds a file location into memory for later use in logging, permiting just an abbreviation or location index of the file path to be used in the log call. The abbreviation is made of the first three letters of each folder/file seperated by a '/'.

Returns: A integer value that point to the stored log location.

```
1    //Save a file of "folder/File.cpp" to the logging location
2    int locationindex = pessum::logging::AddLogLocation("folder/File.cpp");
3    //locationindex can be used in log calls to reference this locaiton, or the
         abbreviation "fol/Fil" can be used
```

1.2.4   Log

```
1    void pessum::logging::Log(std::string str);
2    void pessum::logging::Log(std::string typestr, std::string logstr, std::::
         string locationstring);
3    void pessum::logging::Log(std::string typestr, std::string logstr, std::::
         string locationstr, std::string funcitonstr);
```

1.

1.2.5 LogTimeStamp

```
1   void pessum::logging::LogTimeStamp(bool date);
```

1.

1.2.6   GetLocation

```
std::string pessum::logging::GetLocation(int index);
std::stirng pessum::logging::GetLocation(std::string str);
```

1.

1.2.7 RemoveCaps

```
1    std :: string  pessum :: logging :: RemoveCaps ( std :: string  str ) ;
```

1.

1.2.8 GetType

```
1    std :: string  pessum :: logging :: GetType ( std :: string  str ) ;
```

1.

1.3   LuxReader

1.3.1 LoadLuxFile

1.

1.3.2   FindType

1.

1.3.3 MakeCaps

1.