Mass Conversions in DLL file

1 Introduction:

The conversion DLL file of unit Mass is written in C/C++ programming language style, and do not require any extra code in using this DLL file. This DLL file includes 2 functions to handle the Mass conversions as follow:
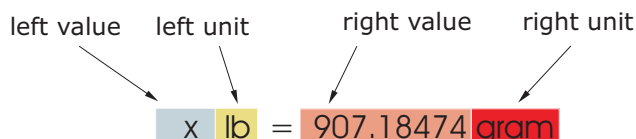
```
double LP_UnitConversionsMass_GetLeftValue (char* LeftUnit, char* RightUnit, double RightValue)  ;
double LP_UnitConversionsMass_GetRightValue(char* LeftUnit, char* RightUnit, double LeftValue )  ;
```

2 Problems in Mass conversion

The two functions in DLL file are used to handle all Mass conversions in two problems.

Problem 1   The unknown value is on the **left hand side** of equation

This problem in conversions is described in the figure:

The value x is obtained by either one of two methods:

- Method A : The code is :

```
double x = LP_UnitConversionsMass_GetLeftValue("Pound", "Gram", 907.18474)    ;
```

- Method B : The code is :
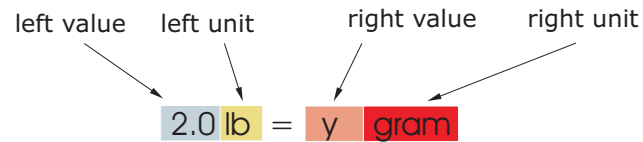
```
char LeftUnit[]   = "Pound"                 ;
char RightUnit[]  = "Gram"             ;

double RightValue = 907.18474             ;

double x = LP_UnitConversionsMass_GetLeftValue(LeftUnit, RightUnit, RightValue)    ;
```

Problem 2   The unknown value is on the **right hand side** of equation

This problem in conversions is described in the figure:

The value y is obtained by either one of two methods:

- Method A : The code is :

```
double y = LP_UnitConversionsMass_GetRightValue("Pound", "Gram", 2.0)   ;
```

- Method B : The code is :

```
char RightUnit[]  = "Pound"              ;
char LeftUnit[]   = "Gram"        ;

double LeftValue = 2.0              ;

double y = LP_UnitConversionsMass_GetRightValue(LeftUnit, RightUnit, LeftValue)   ;
```

3 Unit names in Mass conversions

You can choose the unit name (case sensitive) in the following table for parameters, LeftUnit and/or RightUnit

Milligram	Stone
Gram	OunceAvoir
Kilogram	MetricTon
Carat	ShortTon
Grain	LongTon
OunceTroy	Pennyweight
Pound	ShortHundredweight
Slug	LongHundredweight

When your unit name is not in this table, the returns of functions are −1 and the error message will issue as shown in the following figures:
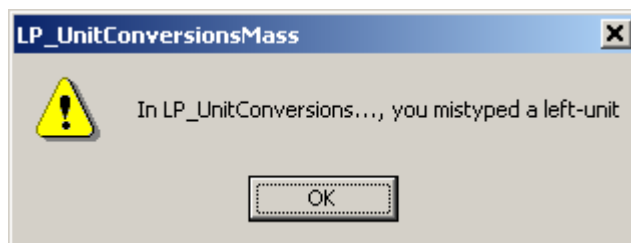
or

4 Tip

1. The parameters in two functions have the same order of unit name (LeftUnit is first and RightUnit is second) and the last parameter is the known value.

2. Use function LP_UnitConversionsMass_GetLeftValue(..)　if your **unknown** value is on the **left hand side** of the equation.

3. Use function LP_UnitConversionsMass_GetRightValue(..)　if your **unknown** value is on the **right hand side** of the equation.