

# Pressure Conversions in DLL file

## 1 Introduction

The conversion DLL file of unit Pressure 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 Pressure conversions as follow:

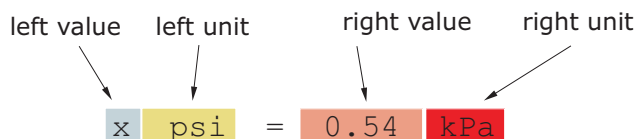
```
double LP_UnitConversionsPressure_GetLeftValue (char* LeftUnit, char* RightUnit, double RightValue) ;
double LP_UnitConversionsPressure_GetRightValue(char* LeftUnit, char* RightUnit, double LeftValue ) ;
```

## 2 Problems in Pressure conversion

The two functions in DLL file are used to handle all Pressure 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_UnitConversionsPressure_GetLeftValue("Psi", "Kilopascal", 0.54) ;
```

- Method B : The code is :

```
char LeftUnit[] = "Psi" ;
```

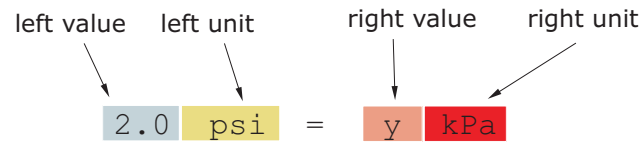
```
char RightUnit[] = "Kilopascal" ;
```

```
double RightValue = 0.54 ;
```

```
double x = LP_UnitConversionsPressure_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_UnitConversionsPressure_GetRightValue("Psi", "Kilopascal", 2.0) ;
```

- Method B : The code is :

```
char LeftUnit[] = "Psi" ;
```

```
char RightUnit[] = "Kilopascal" ;
```

```
double LeftValue = 2.0 ;
```

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

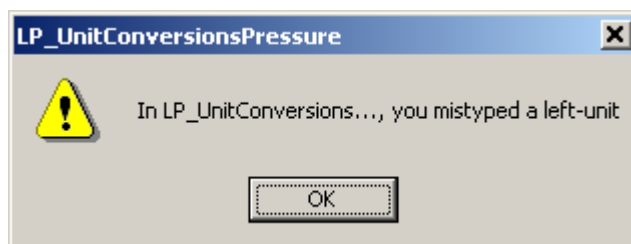
(...continue in next page)

### 3 Unit names in Pressure conversions

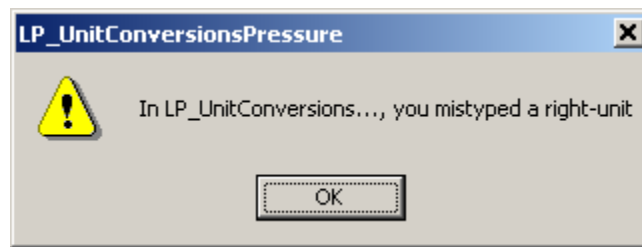
You can choose the unit name (case sensitive) in the following table for parameters, LeftUnit and/or RightUnit

Atmosphere	NewtonPerSquareMillimeter
Bar	NewtonPerSquareCentimeter
Millibar	NewtonPerSquareMeter
DynePerSquareCentimeter	Pascal
GramForcePerSquareMillimeter	Kilopascal
GramForcePerSquareCentimeter	Megapascal
GramForcePerSquareMeter	Psi
KilogramForcePerSquareMillimeter	PoundForcePerSquareInch
KilogramForcePerSquareCentimeter	PoundForcePerSquareFoot
KilogramForcePerSquareMeter	PoundalPerSquareInch
MillimeterHg	PoundalPerSquareFoot
CentimeterHg	Torr
MeterHg	MillieterWater
InchHg	CentimeterWater
FootHg	MeterWater
KipPerSquareInch	InchWater
KipPerSquareFoot	FootWater

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_UnitConversionsPressure_GetLeftValue(..)` if your **unknown** value is on the **left hand side** of the equation.
3. Use function `LP_UnitConversionsPressure_GetRightValue(..)` if your **unknown** value is on the **right hand side** of the equation.