PROCESS | UTILITES

[Documentation]

www.ProcessUtilities.com

ProcessUtilities is the ultimate add-in for chemical engineers. Its thirty-plus custom functions and macros are designed specifically to streamline engineering in Excel. This document describes its function, but the user is strongly encouraged to visit www.ProcessUtilities.com to download the latest examples and access tutorials.

ProcessUtilities Functions

PRIMARY CONVERSION FUNCTIONS	
Conv	Converts a value from one unit of measure to another.
TConv	Converts a value from one temperature scale to another.
PConv	Converts a value from one pressure scale to another.
ConvFactor	Evaluates a conversion factor, i.e. ConvFactor("gal/ft3") returns the number of gallons in one cubic foot.

DIMENSIONAL ANALYSIS FUNCTIONS	
BaseUnits	Returns the metric base units of a unit string.
UnitMath	Returns the result of multiplying or dividing unit strings.
Dimensionless	Uses dimensional analysis to construct and calculate the value of a dimensionless number based on the arguments specified.
Dimensional	Uses dimensional analysis to construct and calculate a result with the specified units.
MassMolesVolume	Converts between moles/standard volume, mass, volume, density, molecular weight, and molar volume.
IdealGas	Uses up to five specified parameters in the ideal gas equation to calculate the parameter with the result units specified, i.e. temperature, pressure, moles, volume, mass, molecular weight, molar volume, molar density, and mass density.

MOLECULAR FORMULA FUNCTIONS	
MW	Returns the molecular weight corresponding to a molecular formula.
CountAtom	Returns the number of atoms of a specified element in a molecular formula, or an array of chemical formulas.
MassToMoleComposition	Convert a composition from mass fraction to mole fraction based on chemical formulas.
MolesToMassComposition	Convert a composition from mole fraction to mass fraction based on chemical formulas.

ENGINEERING REFERENCE FUNCTIONS	
Constant	Returns the value of the specified physical constant in the specified units.
STP	Returns the value of temperature, pressure, ideal molar volume, or ideal molar density at standard conditions, based on the units specified.
SaturatedSteam	Returns temperature, pressure, specific volume, internal energy, enthalpy, or entropy for saturated steam, based another of these properties.
SuperheatedSteam	Returns specific volume, internal energy, enthalpy, or entropy for super-heated steam, based on temperature and pressure.
Z	Returns the generalized compressibility factor for a gas based on reduced temperature and pressure.
PipeSize	Returns inner diameter, outer diameter or internal area of a pipe of the specified size and schedule, in the specified units.
Roughness	Returns the absolute surface roughness of the specified material, for use in pressure drop calculations.
EquivalentLength	Returns the length of pipe that would produce equivalent pressure drop to various standard pipe fittings.

PIPE FLOW & PRESSURE DROP FUNCTIONS	
PressureDrop	Solves the Darcy-Weisbach equation to calculate head loss or pressure drop due to friction.
FlowRegime	Returns the flow regime of liquid flowing in a pipe based on Reynolds number.
PipeFlow	Converts between flow rate and velocity in a pipe.
HeadPressure	Converts between pressure drop and head pressure.

CvLiquid	Liquid flow coefficient calculator. Calculates whichever value is not specified: Cv, volumetric flow rate, or pressure drop.
CvGas_Subcritical	Gas flow coefficient calculator for subcritical flow (pressure ratio > 0.5). Calculates whichever value is not specified: Cv, standard flow rate/mole flow rate, or outlet pressure.
CvGas_Critical	Gas flow coefficient calculator for critical flow (pressure ratio < 0.5). Calculates whichever value is not specified: Cv, standard flow rate, or inlet pressure.
PackedBedPressureDrop	Uses the Ergun Equation to estimate pressure drop through a packed bed.

OTHER ENGINEERING FUNCTIONS	
PumpPower	Calculates pump fluid power based on any appropriate combination of volume flow rate, mass flow rate, density, head, and pressure drop.
CompressorPower	Calculates compressor power based on inlet pressure, outlet pressure, and either volume flow rate, or inlet temperature and molar flow rate. Optionally, heat capacity ratio can also be specified.
LMTD	Returns the log mean temperature difference.
QuadraticFormula	Uses the quadratic formula to find roots of quadratic equations of the form 0=a*x^2+b*x+c.

Conv function

Description

The **Conv** function converts a value from one unit of measure to another. It works with more than 150 different units, as well as combinations of these units.

Syntax

```
Conv(Value, Units, NewUnits)
```

The **Conv** function syntax has the following arguments:

- **Value** Value to convert
- **Units** Units for 'Value'
- **NewUnits** Units for the result

Conv accepts the following text values (in quotation marks or as cell references) for Units and NewUnits. Combinations of these units are also acceptable, for example, ‘ft3/s’ or ‘L-atm/(mol-K)’. Units marked with an ‘*’ can be used with metric prefixes. See the 'Remarks' section below for more information and examples.

Mass	
atomic mass unit	u
grain	gr
gram *	g
ounce	oz
pound	lb
slug	slug
short ton	ton
metric ton	t
Time	
second *	s
minute	min
hour	h
day	d
week	wk
month	mon
year	yr
Quantity	
gram mole *	mol, gmol
elementary entities	ee
kilogram mole	kgmol
pound mole	lbmol
standard cubic foot	scf, SCF
standard cubic meter	scm, SCM
hundred standard cubic feet	ccf, CCF
thousand standard cubic feet	mcf, Mcf, MCF
million standard cubic feet	mmcf, MMcf, MMCF
billion standard cubic feet	Bcf, BCF
standard liter	sL
standard cubic centimeter	scc, SCC
Quantity flow	
standard liter per minute	sLpm, SLPM
standard liter per hour	sLph
standard liter per day	sLpd

standard cubic centimeter per minute	sccm, SCCM
standard cubic feet per minute	scfm, SCFM
standard cubic feet per hour	scfh, SCFH
standard cubic feet per day	scfd, SCFD
thousand standard cubic feet per day	mcfd, Mcfd, MCFD
million standard cubic feet per day	mmcfd, MMcfd, MMCFD
billion standard cubic feet per day	bcfd, Bcfd, BCFD
Temperature	
kelvin	K
degree Celsius	degC
degree Rankine	degR
degree Fahrenheit	degF
Length	
meter *	m
angstrom	ang
mil	mil
inch	in
foot	ft
yard	yd
nautical mile	nmi
mile	mi
fathom	ftm
Speed	
mile per hour	mph
kilometer per hour	kmph
Area	
acre	acre
hectare *	ha
Volume	
liter *	L
ounce	floz
quart	qt
gallon	gal
barrel	bbl
thousand barrels	Mbbl
million barrels	MMbbl
Volume rate	
actual cubic feet per minute	acfm

gallons per minute	gpm
gallons per hour	gph
gallons per day	gpd
Liters per minute	Lpm
Liters per hour	Lph
Liters per day	Lpd
Force	
newton *	N
dyne	dyn
pound-force	lbf
kilogram-force	kgf
Energy	
joule *	J
erg	erg
electron volt	eV
calorie	cal
British thermal unit	btu, Btu, BTU
kilocalorie	kcal
therm	thm
ton of refrigeration	TR
thousand British thermal units	MBtu, MBTU, Mbtu
million British thermal unit	MMBtu, MMBTU, MMbtu
Power	
watt *	W
horsepower	hp, Hp, HP
Pressure	
pascal *	Pa
bar	bar
millimeter Hg	mmHg
cmHg	cmHg
torr	torr
foot water	ftH2O
inch water	inH2O
inch Hg	inHg
pound per square inch	psi
meter water	mH2O
kilogram per square centimeter	kgcm2
atmosphere	atm

Kinematic viscosity	
stokes *	St
Dynamic viscosity	
poise *	P
Angle	
radian	rad
degree	deg
turn	rev
Electric potential	
volt *	V
Electrical charge	
coulomb *	C
Current	
ampere *	A
Resistance	
Ohm *	Ohm
Electrical capacitance	
Farad	F
Electrical conductance	
siemens	S
Magnetic flux	
weber	Wb
Magnetic field strength	
Tesla	T
Inductance	
Henry	H
Luminous intensity	
candle	cd
Luminous flux	
lumen	lm
Illuminance	
lux	lx
Radioactivity	
Becquerel	Bq
Absorbed dose	
gray	Gy
Equivalent dose	
Sievert	Sv

Catalytic activity	
katal	kat
Frequency	
hertz	Hz
Gas permeability	
barrer	barrer

The following abbreviated unit prefixes can be used with the metric unit marked above marked with an '*'.

Prefix	Multiplier	Abbreviation
tera	1E+12	T
giga	1E+09	G
mega	1E+06	M
kilo	1E+03	k
hecto	1E+02	h
deci	1E-01	d
centi	1E-02	c
milli	1E-03	m
micro	1E-06	u
nano	1E-09	n
pico	1E-12	p
femto	1E-15	f
atto	1E-18	a

Remarks

- Exponents can be entered as digits after a unit, i.e. ‘m3’ or ‘m^3’ will be interpreted as cubic meters and can be converted to any other combination of units that has the dimensions of volume.
- Exponents can be integers, decimals or fractions, but they must be positive. Units with negative exponents should be put in the denominator.
- If the input data types are incorrect, the units do not exist, or the dimensions of the units do not match, CONV returns the #VALUE! error value.
- Order of operations is observed in compound units, for example, "J/mol-K" is equivalent to "J-K/mol", not "J/(mol-K)".
- Unit names and prefixes are case-sensitive.
- Differential temperature conversions can be performed by the CONV function. Temperature scale conversions should use the TCONV function, for example converting a measured temperature.
- Pressure unit conversions can be performed by the CONV function, for example converting units for pressure drop. Gauge pressure conversions should use the PCONV function.
- Dashes or asterisks should be used to signify multiplication in compound units, i.e. "N-m" or "N*m"
- A forward slash should be used to signify division in compound units, i.e. "m/s"

Examples

1	1	lb	*Converts 1 pound to kilograms*
	=Conv(C4,D4,D5)	kg	
2	5	m3	*Converts 5 cubic meters to gallons*
	1,321	gal	
3	8.314	J/(mol-K)	*Converts the ideal gas constant from L/(mol-K) to ft3-psi/(scf-degF)*
	0.0299	ft3-psi/(scf-degF)	
4	3.2	kg-m/s	*Unit dimensions don't match, so an error is returned*
	#VALUE!	N	
5	0	degC	*Converts temperature differential of 0 degC to degF*
	0.00	degF	

* All formulas are constructed similarly to first example.

TConv function

Description

The **TConv** function converts a value from one temperature **scale** to another. This is distinguished from the **Conv** function, which is used to convert temperature differentials, but not temperature scales.

Syntax

```
TConvValue, Units, NewUnits)
```

The **TConv** function syntax has the following arguments:

- **Value** Value to convert
- **Units** Units for 'Value'
- **NewUnits** Units for the result

TConv accepts the following text values (in quotation marks or as cell references) for Units and NewUnits.

Temperature scale	
Celsius	degC
Kelvin	K
Fahrenheit	degF
Rankine	degR

Remarks

- If the input data types are incorrect or the units do not exist, **TConv** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	200	degC	*Converts 200 degC to degR*
	=TConv(C4,D4,D5)	degR	
2	0	degC	*Converts 0 degC to degF*
	32.0	degF	
3	298	K	*Converts 298 K to degF*
	76.7	degF	

* All formulas are constructed similarly to first example.

PConv function

Description

The **PConv** function converts a value from one pressure **scale** to another. It works with pressure units that have gauge and ambient notation. This is different from the **Conv** function, which is used to convert pressure differentials or pressures on the same basis, but not between gauge and absolute pressures.

Syntax

```
PConv(Value, Units, NewUnits)
```

The **PConv** function syntax has the following arguments:

- **Value** Value to convert
- **Units** Units for Value
- **NewUnits** Units for the result

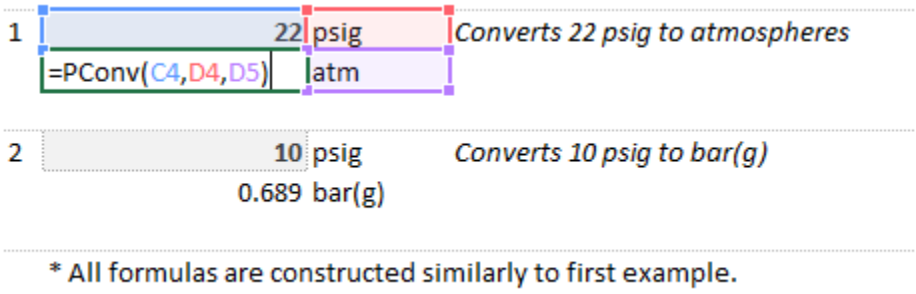
PConv accepts the following text values (in quotation marks or as cell references) for Units and NewUnits. Note that a 'g' or an 'a' can be added to any of these units to signify gauge or absolute pressure, respectively.

Pressure scale	
pascal *	Pa
bar *	bar
atmosphere	atm
inch water	inH2O
foot water	ftH2O
meter water	mH2O
millimeter mercury	mmHg
centimeter mercury	cmHg
inch mercury	inHg
torr	torr
kilogram per square centimeter	kg_cm2
pound per square inch	psi

Remarks

- Gauge pressures are based on the setpoint ambient pressure, which can be set through the ProcessUtilities user interface under the heading 'Std / Amb Conditions'. Default is 1 atm.
- If the input data types are incorrect or the units do not exist, **PConv** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example



ConvFactor function

Description

The **ConvFactor** function returns the value of a conversion factor, i.e. ‘gal/ft3’.

Syntax

```
ConvFactor (UnitRatio)
```

The **ConvFactor** function syntax has the following argument:

- **Value** A ratio of two units that have the same dimensions

Example

1	=ConvFactor(D4)	gal/ft3	*Calculates the equivalent of 1 ft3 in gallons*
2	44.70	(cm/s)/mph	*Calculates the equivalent of 1 mph in 'cm/s'*

* All formulas are constructed similarly to first example.

BaseUnits function

Description

The **BaseUnits** function returns the metric base units of a unit string.

Syntax

```
BaseUnits (Units)
```

The **BaseUnits** function syntax has the following arguments:

- **Units** The unit string to be simplified.

BaseUnits returns the following unit symbols for each dimension.

Dimension	Unit	Symbol
Time	second	s
Length	meter	m
Mass	kilogram	kg
Electric current	ampere	A
Temperature	kelvin	K
Luminous intensity	candela	cd
Amount of substance	mole	mol

Remarks

- See entry for the **Conv** function regarding acceptable units arguments.
- If the input data types are incorrect or the units do not exist, **BaseUnits** returns the #VALUE! error value.

Example

1	Units	gpm
	BaseUnits	=BaseUnits(D4)
2	Units	Btu
	BaseUnits	m2-kg/s2
3	Units	ft3-psi/(lbmol-degF)
	BaseUnits	m2-kg/(mol-K-s2)

* All formulas are constructed similarly to first example.

UnitMath function

Description

The **UnitMath** function returns the product of one to six unit strings, raised to the specified exponents and simplified.

Syntax

```
UnitMath(Units1, Exponent1, Units2, Exponent2, Units3, Exponent3, ...)
```

The **UnitMath** function syntax has the following arguments:

- **Units1** First unit string
- **Exponent1** Exponent for Units1
- **Units2** First unit string
- **Exponent2** Exponent for Units2
- **Units3** First unit string
- **Exponent3** Exponent for Units3
- **…**

Remarks

- See entry for the **Conv** function regarding acceptable units arguments.
- If the input data types are incorrect or the units do not exist, **UnitMath** returns the #VALUE! error value.
- Division of unit strings can be performed by specifying negative exponents.
- **UnitMath** can accept up to six string/exponent pairs as arguments.

Example

1	Units/Exponent 1	m2	1
	Units/Exponent 2	m/s	1
	Result	=UnitMath(D4,E4,D5,E5)	
2	Units/Exponent 1	gal	1
	Units/Exponent 2	gal/s	-1
	Result	s	
3	Units/Exponent 1	mol	1
	Units/Exponent 2	L-atm/(mol-K)	1
	Units/Exponent 3	K	1
	Units/Exponent 4	atm	-1
	Result	L	

* All formulas are constructed similarly to first example.

Dimensionless function

Description

The **Dimensionless** Function uses dimensional analysis to construct and calculate the value of a dimensionless number based on the arguments specified. Value/unit pairs can be entered in any order, given that all numerator value/unit pairs are entered before all denominator pairs.

Syntax

```
Dimensionless(Value1, Units1, Value2, Units2, Value3, Units3, ...)
```

The **Dimensionless** function syntax has the following arguments:

- **Value1** First parameter value
- **Units1** First parameter units
- **Value2** Second parameter value
- **Units2** Second parameter units
- **Value3** Third parameter value
- **Units3** Third parameter units
- **…**

Remarks

- See entry for the **Conv** function regarding acceptable units arguments.
- If the input data types are incorrect, the units do not exist, or a dimensionless number cannot be constructed from the arguments, **Dimensionless** returns the #VALUE! error value.
- Dimensionless numbers that can be calculated include Re, Pr, Nu, Sh, Ma, Sc, Fo, La, Kn, and others.
- Dimensionless can accept up to six value/units pairs as arguments.

Example

1	Density	1.45	g/cm3
	Velocity	55.0	ft/min
	Inner diameter	0.80	in
	Viscosity	1.10	cP
	Reyonlds number	=Dimensionless(D4,E4,D5,E5,D6,E6,D7,E7)	
2	Velocity	55.0	ft/min
	Inner diameter	0.80	in
	Viscosity	0.76	cSt
	Reyonlds number	7,480	
3	Heat trans. coefficient	4.50	Btu/(ft2-h-degF)
	Characteristic length	1.25	m
	Thermal conductivity	10.0	W/m/K
	Nusselt number	3.19	

* All formulas are constructed similarly to first example.

Dimensional function

Description

Uses dimensional analysis to construct and calculate a result with the specified units. Value/unit pairs can be entered in any order, given that all numerator value/unit pairs are entered before all denominator pairs.

Syntax

```
Dimensional(Result units, Value1, Units1, Value2, Units2, Value3, Units3, ...)
```

The **Dimensional** function syntax has the following arguments:

- **Result units** Units for the result
- **Value1** First parameter value
- **Units1** First parameter units
- **Value2** Second parameter value
- **Units2** Second parameter units
- **Value3** Third parameter value
- **Units3** Third parameter units
- **…**

Remarks

- See entry for the **Conv** function regarding acceptable units arguments.
- If the input data types are incorrect, the units do not exist, or a number with the specified dimensions cannot be constructed from the arguments, **Dimensionless** returns the #VALUE! error value.
- Dimensionless can accept up to six value/units pairs as arguments.
- Only multiplication and division operations are supported.
- Values with the units 'degC' and 'degF' are considered as temperature differentials. Units of 'K' or 'degR' should be used where absolute temperature is needed.

Example

1	Heat trans. coefficient	4.50	Btu/(ft2-h-degF)
	LMTD	25.0	degC
	Heat transfer area	12.0	m2
	Heat transfer rate	=Dimensional(E7,D4,E4,D5,E5,D6,E6)	W
2	Length	55.0	ft
	Width	0.80	m
	Height	2.00	in
	Volume	180	gal
3	Flow rate	22.0	scfm
	Ideal gas constant	8.31	J/mol/K
	Characteristic length	80.0	degF
	Pressure	150	psi
	Volume rate	505.0	ft3/d

* All formulas are constructed similarly to first example.

MassMolesVolume function

Description

Converts between moles/standard volume, mass, volume, density, molecular weight, mole volume, and mole density.

$$V_m = \frac{V}{n} = \frac{m}{\rho \cdot n} = \frac{MW \cdot V}{m} = \frac{MW}{\rho}$$

where	V_m	molar volume	MW	molecular weight.
	V	volume or volume rate	n	moles or mole rate
	m	mass or mass rate	ρ	mass density

Syntax

```
MassMolesVolume(Formula, Value, Units, NewUnits)
```

The **MassMolesVolume** function syntax has the following arguments:

- **Formula** Chemical formula
- **Value** Value to convert
- **Units** Units for Value
- **NewUnits** Units for the result

MassMolesVolume accepts any chemical formula that meets the requirements described under the section for the **MW** function. It will likewise accept any combination of units as described under the section for the **CONV** function, as long as the dimensions of Units/NewUnits can be simplified to Mass/Quantity or Quantity/Mass.

Remarks

- If the input data types are incorrect or the units do not exist, **MassMolesVolume** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Standard volume rate	18.4	scfh	*Calculates the actual volume rate*
	Molecular weight	32.0	lb/lbmol	*equivalent to 18.4 scfh*
	Mass density	2.1	kg/m3	
	Actual volume rate	=MassMolesVolume(E7,D4,E4,D5,E5,D6,E6)	m3/d	
2	Mass	5.00	lb	*Calculates the number of moles*
	Molecular weight	134	g/mol	*in 5 lb of a substance with a*
	Moles	16.9	mol	*molecular weight of 134 g/mol*
3	Volume	25.4	gal	*Calculates the number of moles in*
	Mass density	0.92	g/cm3	*25.4 gal of a substance with a*
	Molecular weight	48	g/mol	*molecular weight of 48 and a density of 0.92*
	Moles	1,843	mol	

* All formulas are constructed similarly to first example.

IdealGas function

Description

The **IdealGas** function uses up to five specified parameters in the ideal gas equation to calculate the parameter with the result units specified:

$$V_m = \frac{RT}{p} = \frac{V}{n} = \frac{m}{\rho \cdot n} = \frac{MW \cdot V}{m} = \frac{MW}{\rho}$$

V_m	molar volume	R	ideal gas constant
T	temperature	p	pressure
V	volume or volume rate	n	moles or mole rate
m	mass or mass rate	ρ	mass density
MW	molecular weight.		

Syntax

```
IdealGas(ResultUnits, Value1, Units1, Value2, Units2, …, Value5, Units5)
```

The **IdealGas** function works with any dimensionally consistent combination of the following parameters:

- **Moles / Mole rate** Amount of gas in units of moles (rate) or standard volume (rate)
- **Volume / Volume rate** Volume of gas
- **Mass / Mass rate** Mass of gas
- **Temperature** Temperature of gas
- **Pressure** Pressure
- **Molecular weight** Molecular weight
- **Molar volume** Molar volume in units of volume per mole (or std volume)
- **Mass density** Mass density in units of mass per volume
- **Molar density** Molar density in units of moles per volume

Parameters can be entered in any order, as long as ResultUnits is specified first and units are specified after their corresponding values.

Remarks

- If the input data types are incorrect or the units do not exist, **IdealGas** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Standard volume rate	10	scfm	*Calculates the actual volume rate equivalent to 10 scfm of an ideal gas at 100degC and 15 psig*
	Temperature	100.0	degC	
	Pressure	15	psig	
	Actual volume rate	=IdealGas(E7,D4,E4,D5,E5,D6,E6)	m3/d	
2	Temperature	325	K	*Calculates the mass density of an ideal gas with an MW of 14 g/mol at 325K and 5.2 bar*
	Pressure	5.2	bar	
	Molecular weight	14	g/mol	
	Mass density	2.69	kg/m3	
3	Standard volume rate	18.4	scfh	*Calculates the actual volume rate equivalent to 18.4 scfh*
	Molecular weight	32.0	lb/lbmol	
	Density	2.1	kg/m3	
	Actual volume rate	8.50	m3/d	
4	Mass	5.00	lb	*Calculates the number of moles in 5 lb of a substance with a molecular weight of 134 g/mol*
	Molecular weight	134	g/mol	
	Moles	16.9	mol	

* All formulas are constructed similarly to first example.

MW function

Description

Returns the molecular weight of the compound with the specified molecular formula. Units other than "g/mol" can be specified optionally.

Syntax

```
MW(Formula, [Units])
```

The **MW** function syntax has the following arguments:

- **Formula** Required. The molecular formula of the compound.
- **Units** Optional. The units for the molecular weight. Default is 'g/mol'.

Remarks

- If the input data types are incorrect, do not exist, or units are of the wrong dimensions, **MW** will return #VALUE! error.
- Formulas and prefixes are case-sensitive.
- Hydrate and similar formulas are acceptable, see example 3 below.
- See 'Remarks' section for **Conv** function regarding acceptable units arguments.
- Source: IUPAC

Example

1	Formula	Na2SO4		*Returns the molecular weight of sodium sulfate*
	Molecular weight	=MW(D4)	g/mol	*in default units of g/mol*
2	Formula	(CH3)3N		*Returns the molecular weight of trimethylamine*
	Molecular weight	130.32	lb/kmol	*in lb/kmol*
3	Formula	Na2SO4-10H2O		*Returns the molecular weight of sodium sulfate*
	Molecular weight	322.19	g/mol	*decahydrate*
4	Formula	EtOH		*Returns the molecular weight of ethanol*
	Molecular weight	46.07	g/mol	

* All formulas are constructed similarly to first example.

MW accepts the following text values (in quotation marks or as cell references) for formula.

Atomic no.	Name	Symbol
1	Hydrogen	H
2	Helium	He
3	Lithium	Li
4	Beryllium	Be
5	Boron	B
6	Carbon	C
7	Nitrogen	N
8	Oxygen	O
9	Fluorine	F
10	Neon	Ne
11	Sodium	Na
12	Magnesium	Mg
13	Aluminum	Al
14	Silicon	Si
15	Phosphorus	P
16	Sulfur	S
17	Chlorine	Cl
18	Argon	Ar
19	Potassium	K
20	Calcium	Ca
21	Scandium	Sc
22	Titanium	Ti
23	Vanadium	V
24	Chromium	Cr
25	Manganese	Mn
26	Iron	Fe
27	Cobalt	Co
28	Nickel	Ni
29	Copper	Cu
30	Zinc	Zn
31	Gallium	Ga
32	Germanium	Ge
33	Arsenic	As
34	Selenium	Se
35	Bromine	Br
36	Krypton	Kr
37	Rubidium	Rb

Atomic no.	Name	Symbol
38	Strontium	Sr
39	Yttrium	Y
40	Zirconium	Zr
41	Niobium	Nb
42	Molybdenum	Mo
43	Technetium	Tc
44	Ruthenium	Ru
45	Rhodium	Rh
46	Palladium	Pd
47	Silver	Ag
48	Cadmium	Cd
49	Indium	In
50	Tin	Sn
51	Antimony	Sb
52	Tellurium	Te
53	Iodine	I
54	Xenon	Xe
55	Cesium	Cs
56	Barium	Ba
57	Lanthanum	La
58	Cerium	Ce
59	Praseodymium	Pr
60	Neodymium	Nd
61	Promethium	Pm
62	Samarium	Sm
63	Europium	Eu
64	Gadolinium	Gd
65	Terbium	Tb
66	Dysprosium	Dy
67	Holmium	Ho
68	Erbium	Er
69	Thulium	Tm
70	Ytterbium	Yb
71	Lutetium	Lu
72	Hafnium	Hf
73	Tantalum	Ta
74	Tungsten	W

Atomic no.	Name	Symbol
75	Rhenium	Re
76	Osmium	Os
77	Iridium	Ir
78	Platinum	Pt
79	Gold	Au
80	Mercury	Hg
81	Thallium	Tl
82	Lead	Pb
83	Bismuth	Bi
84	Polonium	Po
85	Astatine	At
86	Radon	Rn
87	Francium	Fr
88	Radium	Ra
89	Actinium	Ac
90	Thorium	Th
91	Protactinium	Pa
92	Uranium	U
93	Neptunium	Np
94	Plutonium	Pu
95	Americium	Am
96	Curium	Cm
97	Berkelium	Bk
98	Californium	Cf
99	Einsteinium	Es
100	Fermium	Fm
101	Mendelevium	Md
102	Nobelium	No
103	Lawrencium	Lr
104	Rutherfordium	Rf
105	Dubnium	Db

Atomic no.	Name	Symbol
106	Seaborgium	Sg
107	Bohrium	Bh
108	Hassium	Hs
109	Meitnerium	Mt
110	Darmstadtium	Ds
111	Roentgenium	Rg
112	Copernicium	Cn
113	Ununtrium	Uut
114	Flerovium	Fl
115	Ununpentium	Uup
116	Livermorium	Lv
117	Ununseptium	Uus
118	Ununoctium	Uuo

Organic group symbols:

Name	Symbol	Formula
Acetyl	AC	C2H3O
Benzoyl	Bz	C6H5CO
Benzyl	Bn	C6H5C2
Butyl	Bu	C4H9
Cyclopentadienyl	Cp	C5H5
Cyclohexyl	Cy	C6H11
Ethyl	Et	C2H5
Methyl	Me	CH3
Mesyl	Ms	CH3SO2
Phenyl	Ph	C6H5
Propyl	Pr	C3H7
Triflyl	Tf	F3CSO2
Trityl	Tr	Ph3C
Tosyl	Ts	CH3C6H4SO2

CountAtom function

Description

Returns the number of atoms of the specified element in a compound of the specified molecular formula or set of formulas.

Syntax

```
CountAtom(Symbol, Formula(s), [Coefficients])
```

The **CountAtom** function syntax has the following arguments:

- **Symbol(s)** Required. Symbol for the element to be counted.
- **Formula(s)** Required. Molecular formula of the compound, or a range containing multiple formulas.
- **Coefficient(s)** Optional. Coefficients corresponding to Formula(s)

CountAtom accepts them same symbols as arguments as the MW function.

Remarks

- If the input data types are incorrect or do not exist, **CountAtom** will return #VALUE! error.
- Formulas and prefixes are case-sensitive.
- Hydrate and similar formulas are allowed, i.e. “Na2SO4-10H2O”.

Example

1	Element	Na	*Returns the number of sodium atoms in one*
	Formula	Na2SO4	*molecule of sodum sulfate*
	Count	=CountAtom(D4,D5)	
2	Element	O	*Returns the number of oxygen atoms in*
	Formula	Na2SO4-10H2O	*sodium sulfate decahydrate*
	Count	14	

* All formulas are constructed similarly to first example.

Stream	1	2	3	4
Description	Liquid in	Gas in	Liquid out	Gas out
Mole rates, mol/s				
Total	2.63E-02	1.69E-02	2.63E-02	1.67E-02
H_2O	2.63E-02		2.60E-02	
H_2O_2			3.54E-04	
H_2SO_4	2.43E-05		2.43E-05	
O_2		3.54E-03		3.37E-03
N_2		1.33E-02		1.33E-02
Atom rates, mol/s				
H	5.27E-02	0.00E+00	-5.27E-02	0.00E+00
O	2.64E-02	7.08E-03	-2.68E-02	-6.73E-03
N	0.00E+00	2.67E-02	0.00E+00	-2.67E-02
S	2.43E-05	0.00E+00	-2.43E-05	=-CountAtom($B47,$B$29:$B$33,G$29:G$33)

Balance checks		
Mass	0.00E+00	Check
H	0.00E+00	Check
O	0.00E+00	Check
N	0.00E+00	Check
S	0.00E+00	Check

MassToMoleComposition function

Description

MassToMoleComposition is an array function that converts a composition from mass fraction to mole fraction. Array functions produce an array of multiple cells as a result and

Syntax

```
MassToMoleComposition(Formulas, Mass fractions)
```

The **MassToMoleComposition** function syntax has the following arguments:

- **Formulas** A range of molecular formulas.
- **Mass fractions** A range of mass fractions corresponding to Formulas.

Remarks

- If the input data types are incorrect or do not exist, **MassToMoleComposition** will return #VALUE! error.
- Array formulas need to be entered differently than normal formulas. The steps are:
    - Select the range of cells that the result values will occupy. (See Example 1 below.)
    - Type in the formula.
    - Hold Control and Shift, and then press Enter.
    - Check out Microsoft's online resources for more information.

Example

1

Species	Mass fraction	Mole fraction
CO2	25.0%	=MassToMoleComposition(C4:C8,D4:D8)
CH4	54.0%	
H2S	1.5%	
H2O	16.0%	
N2	3.5%	

2

Species	Mass fraction	Mole fraction
N2	7.81E-01	8.04E-01
O2	2.09E-01	1.89E-01
Ar	9.34E-03	6.75E-03
CO2	4.00E-04	2.62E-04
Ne	1.82E-05	2.60E-05
He	5.24E-06	3.78E-05
CH4	1.79E-06	3.22E-06

MoleToMassComposition function

Description

MoleToMassComposition is an array function that converts a composition from mole fraction to mass fraction. Array functions produce an array of multiple cells as a result and

Syntax

```
MoleToMassComposition(Formulas, Mole fractions)
```

The **MoleToMassComposition** function syntax has the following arguments:

- **Formulas** A range of molecular formulas.
- **Mole fractions** A range of mole fractions corresponding to Formulas.

Remarks

- If the input data types are incorrect or do not exist, **MoleToMassComposition** will return #VALUE! error.
- Array formulas need to be entered differently than normal formulas. The steps are:
    - Select the range of cells that the result values will occupy. (See Example 1 below.)
    - Type in the formula.
    - Hold Control and Shift, and then press Enter.
    - Check out Microsoft's online resources for more information.

Example

1	Species	Mole fraction	Mass fraction
	CO2	11.4%	=MoleToMassComposition(C4:C8,D4:D8)
	CH4	67.4%	
	H2S	0.9%	
	H2O	17.8%	
	N2	2.5%	

2	Species	Mass fraction	Mass fraction
	N2	8.04E-01	7.81E-01
	O2	1.89E-01	2.09E-01
	Ar	6.75E-03	9.34E-03
	CO2	2.62E-04	4.00E-04
	Ne	2.60E-05	1.82E-05
	He	3.78E-05	5.24E-06
	CH4	3.22E-06	1.79E-06

Constant function

Description

The **Constant** function returns the value of a specified physical constant in the units specified.

Syntax

```
Constant(Symbol, Units)
```

The **Constant** function syntax has the following arguments:

- **Symbol** Symbol or name of the constant
- **Units** Units for the constant

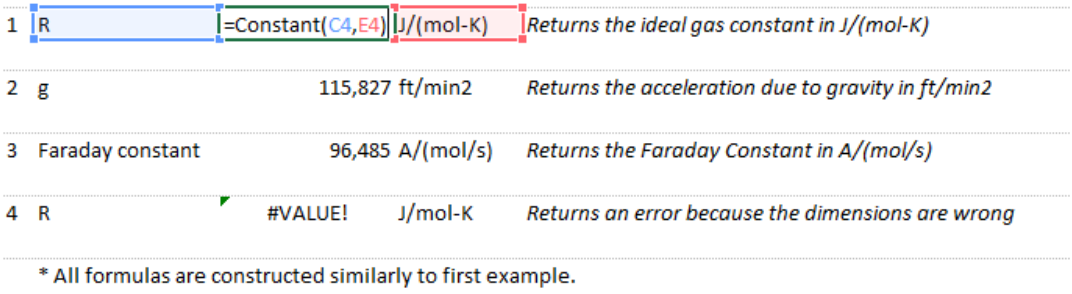
Constant accepts the following text values (in quotation marks or as a cell reference) for symbol, and requires text values for units to be of the following dimensions.

NAME	SYMBOL	VALUE	UNIT DIMENSIONS
Molar gas constant	R	8.3144321	J/(mol-K)
Faraday constant	F	96,485	C/mol
Avogadro's number	Na	6.022 E23	1/mol
Acceleration due to gravity	g	9.80665	m/s2
Newtonian constant of gravitation	G	6.67384 E-11	m3/(kg-s2)
Speed of light	c	2.99776 E8	m/s
Boltzmann constant	k	1.38065 E-23	J/K
Plank constant	h	6.62607 E-34	J-s
Elementary charge	e	1.602176565 E-19	C
Electron mass	me	9.10938291 E-31	kg
Proton mass	mp	1.672621777 E-27	kg

Remarks

- If the input data types are incorrect, do not exist, or units are of the wrong dimensions, **Constant** will return #VALUE! error.
- Unit names and prefixes are case-sensitive.
- See 'Remarks' section of **Conv** function regarding acceptable units arguments.

Example



1	R	=Constant(C4,E4)	J/(mol-K)	*Returns the ideal gas constant in J/(mol-K)*
2	g	115,827	ft/min2	*Returns the acceleration due to gravity in ft/min2*
3	Faraday constant	96,485	A/(mol/s)	*Returns the Faraday Constant in A/(mol/s)*
4	R	#VALUE!	J/mol-K	*Returns an error because the dimensions are wrong*

* All formulas are constructed similarly to first example.

STP function

Description

The **STP** function returns the value of temperature, pressure, ideal molar volume, or ideal molar density at standard conditions, based on the units specified. Standard temperature and pressure can be set through the ProcessUtilities user interface. Default is 0 degC and 1 atm.

Syntax

```
STP(Units)
```

The **STP** function syntax has the following arguments:

- **Units** Required. The desired units for the result.

Remarks

- If the input data types are incorrect or the units do not exist, **STP** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Std. temperature	=STP(E4)	degF	*Returns the standard temperature setpoint in degF*
2	Std. pressure	0.00	psig	*Returns the standard pressure setpoint in psig*
3	Molar volume	359	ft3/lbmol	*Returns the ideal gas molar volume at setpoint STP*

* All formulas are constructed similarly to first example.

SaturatedSteam function

Description

The **SaturatedSteam** function returns temperature, pressure, specific volume, internal energy, enthalpy, or entropy for saturated steam, based any other of these properties.

Syntax

```
SaturatedSteam(Input property name, Input value, Input Units, Result property name, Result units)
```

The **SaturatedSteam** function syntax has the following arguments:

- **Input property name** Name of the input property (see below for options)
- **Input value** Value of the input property
- **Input units** Units for Input value
- **Result property name** Name of result property (see below for options)
- **Result units** Units for the result

SaturatedSteam accepts the following case-insensitive text values (in quotation marks or as cell references) for Input property name and Result property name.

Property	Symbol; Dimensions
Temperature	"T", temperature
Pressure	"P", pressure
Specific volume	"vf", "vg"; volume/mass
Internal energy	"uf", "ug*"; energy/mass
Enthalpy	"hf", "hfg", "hg*"; energy/mass
Entropy	"sf", "sfg", "sg"; energy/(mass*temperature)

Remarks

- Gas internal energy "ug" and gas enthalpy "hg" are acceptable as result property names but not input property names.
- If the input data types or units are incorrect, **SaturatedSteam** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.
- In the property names, a suffix of 'f' indicates the liquid phase, 'g' indicates the gas phase, and 'fg' indicates the difference between the liquid and gas phases.
- Source: NIST

Example

1	T	150	degC
	P	=SaturatedSteam(C4,D4,E4,C5,E5)	psia
2	P	6.50	bar
	hf	294.3	Btu/lb
3	vg	1,215	cm3/g
	T	229.7	degF

* All formulas are constructed similarly to first example.

SuperheatedSteam function

Description

Returns specific volume, internal energy, enthalpy, or entropy for superheated steam, based on temperature and pressure.

Syntax

```
SuperheatedSteam(Temperature, Temperature units, Pressure, Pressure units,
Result property name, Result units)
```

The **SuperheatedSteam** function syntax has the following arguments:

- **Temperature** Temperature value
- **Temperature units** Units for Temperature
- **Pressure** Pressure value
- **Pressure units** Units for Pressure
- **Result property name** Name of result property (see below for options)
- **Result units** Units for the result

SuperheatedSteam accepts the following case-insensitive text values (in quotation marks) for Result Property Name.

Property	Symbol; Dimensions
Specific volume	"v"; volume/mass
Internal energy	"u"; energy/mass
Enthalpy	"h"; energy/mass
Entropy	"s"; energy/(mass*temperature)

Remarks

- If the input data types or units are incorrect, **SuperheatedSteam** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.
- If the temperature and pressure specified does not correspond to steam in a superheated state, or is out of the range of the database, **SuperheatedSteam** will return a #VALUE! error.
- Source: NIST

Example

1	Temperature	150.0	degC
	Pressure	1	atm
	u	=SuperheatedSteam(D4,E4,D5,E5,C6,E6)	kJ/kg
2	Temperature	350	degF
	Pressure	10	psig
	v	144.5	gal/lb
3	Temperature	1050	K
	Pressure	112	psig
	h	4,102	kJ/kg

* All formulas are constructed similarly to first example.

Z function

Description

The **Z** function returns the compressibility factor for a gas based on reduced temperature and pressure.

Syntax

```
Z(Reduced temperature, Reduced pressure)
```

The **Z** function syntax has the following arguments:

- **Reduced temperature** Reduced temperature (actual divided by critical temperature)
- **Reduced pressure** Reduced pressure (actual divided by critical pressure)

Remarks

- Based on linear interpolation of the Nelson-Obert Generalized Compressibility Charts. Accuracy of charts is reported to be 1-2% for **Z** values greater than 0.6, and 4-6% for Z values 0.3-0.6. Strongly polar gases may have more error. Unit names and prefixes are case-sensitive.
- For most gases, reduced pressure is defined as actual temperature divided by critical temperature, and reduced pressure is defined as actual pressure divided by critical pressure.
- For the gases hydrogen, helium, and neon, reduced temperature is defined as $Tr = T/(T_c+8)$, and reduced pressure is defined as $Pr = P/(P_c+8)$.
- If the reduced temperature or reduced pressure specified are outside the range of the charts, Z will return a #VALUE! error.

Example

Standard volume-to-actual volume calculator

Temperature	-15.0	degC	Critical temperature	-146.9	degC
Pressure	10,140	psig	Critical pressure	33.5	atm
Standard volume rate	150.0	scfh			
Actual volume rate	0.343	ft3/h			

Calculations

Temperature	258.2	K	Gen. compressibility factor	=Z(C17,C21)	-
Critical temperature	126.3	K	Ideal volume	0.205	ft3/h
Reduced temperature	2.045	-	Real gas volume	0.343	ft3/h
Pressure	70,014	kPa			
Critical pressure	3,394	kPa			
Reduced pressure	20.626	-			

PipeSize function

Description

The **PipeSize** function returns the inner diameter, outer diameter or transverse inner area for an NPS pipe of the specified size and schedule.

Syntax

```
PipeSize(Pipe size, Dimension, Units)
```

The **PipeSize** function syntax has the following arguments:

- **Pipe size** Pipe size and schedule, in quotes or as a cell reference (see examples)
- **Dimension** Name of result dimension (ID, OD, A)
- **Units** Units for the result

Remarks

- Pipe size should be entered as a string or cell reference including both NPS size and schedule, i.e. "1.5" Sch80", "0.5inch SCH40.
- If the input data types are incorrect or the units do not exist, **PipeSize** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Pipe size	1" Sch 40	
	Outer diameter	=PipeSize($D$4,C5,E5)	in
	Inner diameter	1.049	in
	Area	5.576	cm2
	Wall thickness	3.378	mm
2	Pipe size	3.5in Sch80	
	OD	0.333	ft
	ID	3.364	in
	A	6.17E-02	ft2
	t	8.08E-03	m

* All formulas are constructed similarly to first example.

Roughness function

Description

The **Roughness** function returns the absolute surface roughness of the specified material, for use in pressure drop calculations.

Syntax

```
Roughness(Material, Units)
```

The **Roughness** function syntax has the following arguments:

- **Material** Material name (see options below)
- **Units** Units for the result in dimensions of length

Roughness accepts the following case case-insensitive text values for fitting type.

Materials	
Drawn tubing (brass, lead, glass, steel, etc.)	Galvanized iron
Plastic pipe	Cast iron
Commercial steel	Wood stove
Wrought iron	Concrete
Asphalted cast iron	Riveted steel

Remarks

- If the input data types or units are incorrect, **Roughness** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.
- Data source: Perry's 6-10, GPSA Handbook

Example

1	Material	Drawn tubing	
	Roughness	=Roughness(D4,E5)	in
2	Material	Plastic pipe	
	Roughness	1.50E-03	mm
3	Material	Commercial steel	
	Roughness	1.50E-04	ft

* All formulas are constructed similarly to first example.

EquivalentLength function

Description

The **EquivalentLength** function returns the length of pipe that would produce an equivalent pressure drop to various standard pipe fittings.

Syntax

```
EquivalentLength(Fitting type, Diameter, Diameter units, Length/result units)
```

The **EquivalentLength** function syntax has the following arguments:

- **Fitting type** Fitting type. See below for options
- **Diameter** Inner diameter of fitting pipe size
- **Diameter units** Units for Diameter
- **Length units** Units for the result in dimensions of length

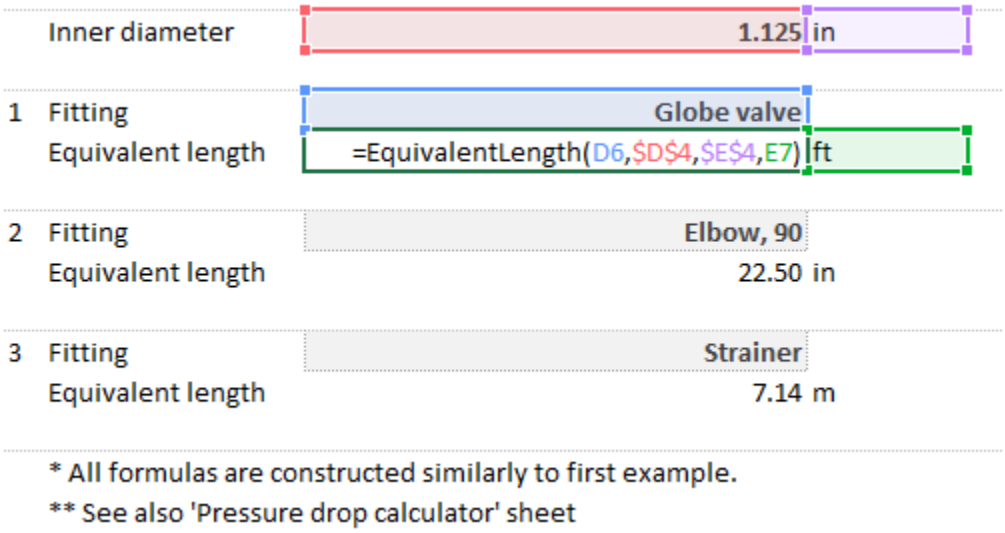
EquivalentLength accepts the following case-insensitive text values for fitting type.

Fitting type	Notes
Ball valve, small	Reduced bore, 1.5" and smaller
Ball valve, large	reduced bore, 2" and larger
Gate valve	standard bore
Gate valve, reduced bore	
Globe valve	Straight pattern
Globe valve, Y	Y pattern
Globe valve, angle	Angle pattern
Check valve, swing	
Check valve, ball	1.5" and smaller
Check valve, piston	1.5" and smaller
Plug valve	
Butterfly valve	6" and larger
Tee, straight	
Tee, branch	
Elbow, 90	R = 1.5D
Elbow, 45	R = 1.5D
Bend, 90, 4D	R = 4D
Bend, 45, 5D	R = 5D
Bend, 90, 4D	R = 4D
Bend, 45, 5D	R = 5D
Strainer	Pump suction Y type and bucket type
Nozzle	Suction nozzle vessel/tank

Remarks

- If the input data types or units are incorrect, **EquivalentLength** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.
- Data source: GPSA handbook

Example


Inner diameter
1.125 in
1 Fitting
Globe valve
Equivalent length
=EquivalentLength(D6,$D$4,$E$4,E7) ft
2 Fitting
Elbow, 90
Equivalent length
22.50 in
3 Fitting
Strainer
Equivalent length
7.14 m
* All formulas are constructed similarly to first example.
** See also 'Pressure drop calculator' sheet

PressureDrop function

Description

Uses the Darcy-Weisbach equation to calculate head loss or pressure drop due to friction.

$$h_f = f \cdot \frac{L}{D} \cdot \frac{u^2}{2g}$$

$$f = \frac{64}{Re}, for\ laminar\ flow$$

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{\varepsilon}{3.7D} + \frac{2.51}{Re\sqrt{f}}\right), for\ turbulent\ flow$$

where	h_f	frictional head loss	f	friction factor
	L	length	D	inner diameter
	u	velocity	g	acceleration of gravity
	Re	Reynolds number	ε	absolute roughness

Syntax

```
PressureDrop( Units,  Diameter,  Diameter units,  Length,  Length Units,
              Roughness,  Roughness units,  Flow rate,  Flow rate units,
              Viscosity,  Viscosity units,  Density,  Density units )
```

The **PressureDrop** function syntax has the following arguments:

- **Unit** Pressure drop or head units for result
- **Diameter** Inner diameter of pipe
- **Diameter units** Units for diameter
- **Length** Length of pipe
- **Length units** Units for length
- **Roughness** Absolute roughness of the pipe material (see ROUGHNESS function)
- **Roughness units** Units for roughness
- **Flow rate** Volume or mass flow rate, or velocity
- **Flow rate units** Units for flow rate
- **Viscosity** Kinematic or dynamic viscosity of the fluid
- **Viscosity units** Units for viscosity
- **Density** Density of the fluid
- **Density units** Units for density

Remarks

- If the input data types or units are incorrect, **PressureDrop** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Inner diameter	1.125	in
	Length	20.0	ft
	Roughness	1.50E-04	ft
	Flow rate	15.0	gpm
	Viscosity	1.01	cP
	Density	985	kg/m3
	Pressure drop	=PressureDrop(E9,D3,E3,D4,E4,D5,E5,D6,E6,D7,E7,D8,E8)	inH2O

2	Inner diameter	1.125	in
	Length	1.2	mi
	Roughness	1.50E-04	ft
	Velocity	55.0	ft/min
	Viscosity	1.01	cP
	Density	985	kg/m3
	Pressure drop	13.24	psi

3	Inner diameter	1.125	in
	Length	20.0	ft
	Roughness	1.50E-04	ft
	Mass flow rate	950.0	kg/s
	Viscosity	1.50	cSt
	Density	985	kg/m3
	Pressure drop	11.58	inH2O

FlowRegime function

Description

Returns the flow regime of liquid flowing in a pipe based on the Reynolds number, either laminar, transition, or turbulent. Flow is considered to be in the laminar regime below a Reynolds number of 2320, transition regime between 2320 and 4000, and turbulent regime above 4000.

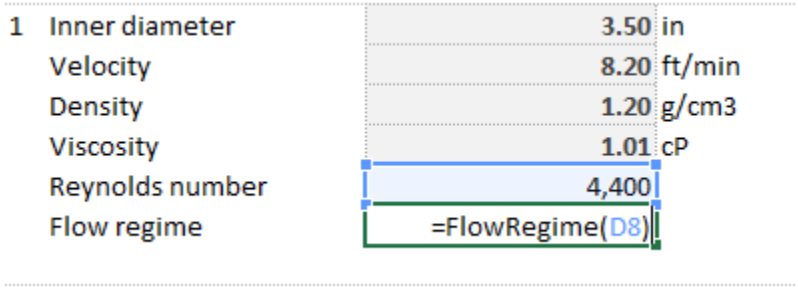
Syntax

```
FlowRegime(Reynolds number)
```

Remarks

- If the input data types are incorrect, or the equation specified has no real roots, **FlowRegime** returns the #VALUE! error value.

Example



PipeFlow function

Description

Converts between flow rate and velocity in a pipe.

$$\dot{V} = \frac{\pi d^2}{4} \cdot v$$

where	V	volume flow rate	v	velocity
	d	diameter		

Syntax

To convert volume rate to velocity:

PipeFlow(Velocity units, Diameter, Diameter units, Volume flow rate, Volume rate units)

To convert velocity to volume rate :

PipeFlow(Volume rate units, Diameter, Diameter units, Velocity, Velocity units)

Remarks

- If the input data types or units are incorrect, **PipeFlow** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Inner diameter	0.125	in
	Flow rate	250	mL/min
	Velocity	=PipeFlow(E6,D4,E4,D5,E5)	cm/s
2	Inner diameter	3.50	in
	Velocity	100	gpm
	Velocity	200.1	ft/min

* All formulas are constructed similarly to first example.

HeadPressure function

Description

Converts between pressure and head, according to the equation:

$$p = \rho \cdot g \cdot h_f$$

where p is pressure, ρ is density, g is gravitational acceleration, and h_f is head.

Syntax

To calculate head:

```
HeadPressure(Units, Density, DensityUnits, Pressure, PressureUnits)
```

To calculate pressure:

```
HeadPressure(Units, Density, DensityUnits, Head, HeadUnits)
```

The **HeadPressure** function syntax has the following arguments:

- **Units** Units for result
- **Density** Density of fluid
- **DensityUnits** Units for density
- **Pressure** Fluid pressure
- **PressureUnits** Units for Pressure
- **Head** Head
- **HeadUnits** Units for Head.

Remarks

- If the input data types or units are incorrect, **HeadPressure** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.
- Gauge pressures are not acceptable arguments.
- Pressure and density value/unit pairs can be entered in either order.

Example

1	Density	1010	kg/m3	*Calculates feet of head for a 1010 kg/m3 fluid at 50 psi*
	Pressure	50.0	psi	
	Head	=HeadPressure(E6,D4,E4,D5,E5)	ft	
2	Density	6.0	lb/gal	*Calculates pressure in bar for a 6 lb/gal fluid with 12 m head*
	Head	12	m	
	Pressure	0.85	bar	

* All formulas are constructed similarly to first example.

CvLiquid function

Description

Liquid flow coefficient calculator. Calculates whichever value is not specified of Cv, flow rate, pressure drop, or density, according to the equation:

$$\dot{V} = Constant \cdot C_v \sqrt{\frac{\Delta p}{SG}}$$

Syntax

To calculate Cv:

```
CvLiquid(Flow rate, Flow rate units, Pressure drop, Pressure drop units, Density, Density units)
```

To calculate pressure drop:

```
CvLiquid(Pressure drop units, Cv, Flow rate, Flow rate units, Density, Density units)
```

To calculate flow rate:

```
CvLiquid(Flow rate units, Cv, Pressure drop, Pressure drop units, Density, Density units)
```

Remarks

- Value/unit argument pairs can be entered in any order, as long as the result units are specified first. If Cv is the result, then units do not need to be specified.
- If the input data types are incorrect, or the equation specified has no real roots, **CvLiquid** returns the #VALUE! error value.
- See documentation for the **Conv** function for acceptable unit arguments.

Example

1	Volume rate	1.125	gpm
	Pressure drop	2.5	bar
	Density	0.85	g/cm3
	Cv	=CvLiquid(D4,E4,D5,E5,D6,E6)	
2	Cv	22.5	
	Pressure drop	15.0	psi
	Density	1.11	g/cm3
	Volume rate	450.9	m3/d
3	Mass rate	1,200	lb/h
	Cv	0.54	
	Density	1.01	g/cm3
	Pressure drop	45.0	ftH2O
4	Volume rate	1.125	gpm
	Cv	0.770	
	Pressure drop	12.50	kPa
	Density	0.85	g/cm3

* All formulas are constructed similarly to first example.

CvGas_Subcritical function

Description

Gas flow coefficient calculator for subcritical flow (pressure ratio > 0.5). Calculates whichever value is not specified: Cv, standard flow rate/mole flow rate, or outlet pressure. See documentation for examples.

$$\dot{n} = C_v P_{in} Y \sqrt{\frac{x}{MW \cdot T_{in} \cdot Z}} = C_v P_{in} \left(1 - \frac{2x}{3}\right) \sqrt{\frac{x}{MW \cdot T_{in}}}$$

$$Y = 1 - \frac{x}{3F_K x_T} = 1 - \frac{2x}{3}$$

where	C_v	flow coefficient	P_{in}	inlet pressure
	Y	expansion factor	n	mole flow rate
	MW	molecular weight	T_{in}	inlet temperature
	x	pressure drop ratio (Pout / Pin)		
	Z	compressibility, set at 1		
	F_k	Ratio of specific heat ratios (k/kair), set at 1		
	x_T	critical pressure drop, set at 0.5		

Syntax

To calculate Cv:

```
CvGas_Subcritical( Flow rate,  Flow rate units,  Inlet temperature,  Inlet temperature units,
                   Inlet pressure,  Inlet pressure units,  Outlet pressure,  Outlet pressure units,
                   MW,  MW units )
```

To calculate standard volume or mole flow rate:

```
CvGas_Subcritical( Flow rate units,  Cv, Inlet temperature,  Inlet temperature units,
                   Inlet pressure,  Inlet pressure units,  Outlet pressure,  Outlet pressure units,
                   MW, MW units )
```

To calculate outlet pressure:

```
CvGas_Subcritical( Outlet pressure units,  Cv,  Flow rate,  Flow rate units,
                   Inlet pressure,  Inlet pressure units,  Inlet temperature,  Inlet temperature units,
                   MW, MW units )
```

The **CvGas_Subcritical** function syntax has the following arguments:

- **Cv** Flow coefficient
- **Flow rate** Mole flow rate or standard volume flow rate
- **Flow rate units** Units for flow rate
- **Inlet pressure** Inlet pressure
- **Inlet pressure units** Units for inlet pressure
- **Outlet pressure** Outlet pressure
- **Outlet pressure units** Units for outlet pressure
- **Inlet temperature** Inlet temperature
- **Inlet temperature units** Units for inlet temperature
- **MW** Molecular weight of gas
- **MW units** Units for MW

Remarks

- Value/unit argument pairs can be entered in any order, as long as the result units are specified first. If Cv is the result, then units do not need to be specified.
- Solving for outlet pressure requires iteration and may be slower than some other functions.
- If the input data types are incorrect, or the equation specified has no real roots, **CvGas_Subcritical** returns the #VALUE! error value.
- See documentation for the **Conv** function for acceptable unit arguments.

Example

1	Rate	120.0	scfm
	Inlet pressure	110.0	psi
	Outlet pressure	80.00	psi
	Inlet temperature	120.00	degF
	Molecular weight	28.97	g/mol
	Cv	=CvGas_Subcritical(D4,E4,D5,E5,D6,E6,D7,E7,D8,E8)	

2	Cv	21.50	
	Inlet pressure	25.0	psi
	Outlet pressure	22.00	psi
	Inlet temperature	290.00	K
	Molecular weight	16.04	g/mol
	Rate	38.17	lbmol/h

3	Cv	2.71	
	Inlet pressure	110.0	psi
	Rate	120.00	scfm
	Inlet temperature	120.00	degF
	Molecular weight	28.97	g/mol
	Outletpressure	79.95	psi

* All formulas are constructed similarly to first example.

CvGas_Critical function

Description

Gas flow coefficient calculator for critical flow (pressure ratio < 0.5). Calculates whichever value is not specified: Cv, standard flow rate, or inlet pressure. See documentation for examples.

$$\dot{n} = C_v P_{in} Y_T \sqrt{\frac{x_T}{MW \cdot T_{in} \cdot Z}} = \frac{2}{3} C_v P_{in} \sqrt{\frac{1}{2 \cdot MW \cdot T_{in}}}$$

$$Y_T = 1 - \frac{1}{3F_K} = \frac{2}{3}$$

where	C_v	flow coefficient	P_{in}	inlet pressure
	Y_T	critical expansion factor	n	mole flow rate
	MW	molecular weight	T_{in}	inlet temperature
	Z	compressibility, set at 1		
	F_k	Ratio of specific heat ratios (k/k_{air}), set at 1		
	x_T	critical pressure drop, set at 0.5		

Syntax

To calculate Cv:

```
CvGas_Subcritical( Flow rate,  Flow rate units,  Inlet temperature,  Inlet temperature units,
                   Inlet pressure,  Inlet pressure units,  MW,  MW units )
```

To calculate standard volume or mole flow rate:

```
CvGas_Subcritical( Flow rate units,  Cv, Inlet temperature,  Inlet temperature units,
                   Inlet pressure,  Inlet pressure units,  MW, MW units )
```

To calculate inlet pressure:

```
CvGas_Subcritical( Inlet pressure units,  Cv,  Flow rate,  Flow rate units,
                   Inlet temperature,  Inlet temperature units,  MW, MW units )
```

The **CvGas_Cricitical** function syntax has the following arguments:

- **Cv** Flow coefficient
- **Flow rate** Mole flow rate or standard volume flow rate
- **Flow rate units** Units for flow rate
- **Inlet pressure** Inlet pressure
- **Inlet pressure units** Units for inlet pressure
- **Inlet temperature** Inlet temperature
- **Inlet temperature units** Units for inlet temperature
- **MW** Molecular weight of gas
- **MW units** Units for MW

Remarks

- Value/unit argument pairs can be entered in any order, as long as the result units are specified first. If Cv is the result, then units do not need to be specified.
- If the input data types are incorrect, or the equation specified has no real roots, **CvGas_Cricitical** returns the #VALUE! error value.
- See documentation for the **Conv** function for acceptable unit arguments.

Example

1	Rate	120.0	scfm
	Inlet pressure	110.0	psi
	Inlet temperature	120.0	degF
	Molecular weight	28.97	g/mol
	Cv	=CvGas_Critical(D4,E4,D5,E5,D6,E6,D7,E7)	
2	Cv	21.50	
	Inlet pressure	25.0	psi
	Inlet temperature	290.00	K
	Molecular weight	16.04	g/mol
	Rate	56.47	lbmol/h
3	Cv	2.46	
	Rate	120.00	scfm
	Inlet temperature	120.00	degF
	Molecular weight	28.97	g/mol
	Inlet pressure	109.89	psi

* All formulas are constructed similarly to first example.

PackedBedPressureDrop function

Description

Uses the Ergun Equation to estimate pressure drop through a packed bed.

$$f_p = \frac{150}{Gr_p} + 1.75$$

$$\text{with } f_p = \frac{\Delta p}{L}\frac{D_p}{\rho v_s^2}\left(\frac{\epsilon^3}{1-\epsilon}\right) \text{ and } Gr_p = \frac{\rho v_s D_p}{(1-\epsilon)\mu}$$

where	f_p	friction factor	v_s	superficial velocity
	Gr_p	modified Reynolds number	ρ	fluid density
	Δp	pressure drop	ϵ	void fraction
	L	bed length/height	μ	dynamic viscosity
	D_p	Particle spherical equivalent diameter		

Syntax

```
PackedBedPressureDrop( Pressure drop (result) units, Superficial velocity, Superficial velocity units,
                       Mass density, Mass density units, Viscosity, Viscosity units,
                       Length, Length units, Particle diameter, Particle diameter units,
                       Void fraction)
```

The **PackedBedPressureDrop** function syntax has the following arguments:

- **Pressure drop units** Units for pressure drop (result)
- **Superficial velocity** Superficial velocity of fluid in bed
- **Superficial velocity units** Units for superficial velocity
- **Mass density** Mass density of fluid
- **Mass density units** Units for mass density
- **Viscosity** Dynamic or kinematic viscosity of fluid
- **Viscosity units** Units for viscosity
- **Length** Length/height of packed bed
- **Length units** Units for length
- **Particle diameter** Equivalent spherical diameter of bed particles
- **Particle diameter units** Units for particle diameter
- **Void fraction** Void fraction of bed

Remarks

- If the input data types are incorrect, or the equation specified has no real roots, **PackedBedPressureDrop** returns the #VALUE! error value.

Example

1	Superficial velocity	65	ft/min
	Density	1000	g/L
	Viscosity	1	cP
	Length	6	ft
	Particle diameter	0.25	in
	Void fraction	0.54	
	Pressure drop	=PackedBedPressureDrop(E10,D4,E4,D5,E5,D6,E6,D7,E7,D8,E8,D9)	psi

2	Superficial velocity	40	cm/s
	Density	0.85	g/cm3
	Viscosity	0.95	cSt
	Length	25	cm
	Particle diameter	2	mm
	Void fraction	0.54	
	Pressure drop	0.910	bar

* All formulas are constructed similarly to first example.

PumpPower function

Description

Calculates pump fluid power based on any dimensionally consistent combination of volume flow rate, mass flow rate, density, differential head, and differential pressure.

$$P = \dot{V}\rho g h = \dot{V}\Delta p = \dot{m} g h$$

where	P	power	V	volume flow rate
	h	head	Δp	pressure drop
	ρ	mass density	g	acceleration of gravity

Syntax

```
PumpPower( Power units,  Volume rate, Volume rate units,  Head,  Head units,
                Density, Density units )
```

```
PumpPower( Power units,  Volume rate,  Volume rate units,  Pressure,  Pressure units )
```

```
PumpPower( Power units,  Mass rate,  Mass rate units,  Head,  Head units )
```

Remarks

- If the input data types or units are incorrect, **PumpPower** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Mass rate	25,000	lb/h
	Head differential	80	ft
	Fluid power	=PumpPower(E6,D4,E4,D5,E5)	hp
	Efficiency	80%	
	Shaft power	1.26	hp
2	Volume rate	125	gpm
	Pressure differential	45	psi
	Fluid power	3.28	hp
	Efficiency	80%	
	Shaft power	4.10	hp
3	Mass rate	150	t/d
	Pressure differential	80	atm
	Density	1.25	g/cm3
	Fluid power	11.26	kW
	Efficiency	80%	
	Shaft power	14.07	kW
4	Volume rate	150	L/min
	Head differential	450	m
	Density	0.95	g/cm3
	Fluid power	35,786	Btu/h
	Efficiency	80%	
	Shaft power	44,733	Btu/h

* All formulas are constructed similarly to first example.

CompressorPower function

Description

Calculates fluid power of compression or expansion based on suction pressure, discharge pressure, and either volume flow rate, or molar flow rate and suction temperature. Optionally, heat capacity ratio can also be specified.

$$P = p_1 \dot{V}_1 \left(\frac{k}{k-1}\right)\left[\left(\frac{p_2}{p_1}\right)^{\left(\frac{k-1}{k}\right)} - 1\right] = \dot{n} R T_1 \left(\frac{k}{k-1}\right)\left[\left(\frac{p_2}{p_1}\right)^{\left(\frac{k-1}{k}\right)} - 1\right]$$

Where	V	inlet volume flow rate	n	mole flow rate
	R	ideal gas constant	T	inlet temperature
	p1	inlet pressure	p2	outlet pressure
	k	ratio of specific heats, C_p/C_v		

Syntax

```
CompressorPower(Power units,  Volume flow rate,  Volume flow rate units,
                Inlet pressure,  Inlet pressure units,  Outlet pressure,  Outlet pressure units,
                [Heat capacity ratio] )
```

```
CompressorPower(Power units,  Mole or standard flow rate,  Mole or standard flow rate units,
                Inlet pressure,  Inlet pressure units,  Outlet pressure,  Outlet pressure units,
                Inlet temperature,  Inlet temperature units,  [Heat capacity ratio] )
```

Remarks

- Specifying the heat capacity ratio is optional. The default value is 1.4.
- If the input data types or units are incorrect, **CompressorPower** returns the #VALUE! error value.
- Unit names and prefixes are case-sensitive.

Example

1	Volume rate	1,200	m3/h
	Suction pressure	10	psig
	Discharge pressure	55	psig
	Fluid power	=CompressorPower(E7,D4,E4,D5,E5,D6,E6)	kW
	Efficiency	80%	
	Shaft power	85.7	kW
2	Std volume or mole rate	120	scfm
	Suction pressure	15	ftH2O
	Discharge pressure	35	ftH2O
	Suction temperature	150	degF
	Fluid power	9.1	hp
	Efficiency	80%	
	Shaft power	11.4	hp
3	Volume rate	1,200	m3/h
	Suction pressure	0	psig
	Discharge pressure	55	psig
	Heat capacity ratio	1.32	
	Fluid power	85.6	hp
	Efficiency	80%	
	Shaft power	107.1	hp
4	Std volume or mole rate	1,200	scfm
	Suction pressure	250	psig
	Discharge pressure	120	psig
	Suction temperature	15	degC
	Heat capacity ratio	1.12	
	Expansion fluid power	-39.5	kW
	Efficiency	80%	
	Expansion shaft power	-49.3	kW

* All formulas are constructed similarly to first example.

LMTD function

Description

Returns the log mean temperature difference used in heat transfer calculations.

$$LMTD = \frac{\Delta T_A - \Delta T_B}{\ln \frac{\Delta T_A}{\Delta T_B}}$$

$$\Delta T_A = T_{A,2} - T_{A,1}$$

$$\Delta T_B = T_{B,2} - T_{B,1}$$

Syntax

```
LMTD( Temperature A1,  Temperature A2,  Temperature B1,  Temperature B2 )
```

The LMTD function syntax has the following arguments:

- **Temperature A1** Temperature of Stream 1, on Side A.
- **Temperature A2** Temperature of Stream 2, on Side A.
- **Temperature B1** Temperature of Stream 1, on Side B.
- **Temperature B2** Temperature of Stream 2, on Side B.

Remarks

- Temperature units are assumed to be consistent.
- If the input data types incorrect, **LMTD** returns the #VALUE! error value.

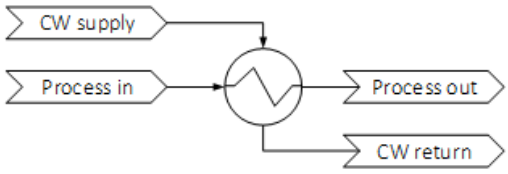
Example

Heat transfer rate calculator for a counter current heat exchanger

Heat transfer coefficient	10.5	Btu/(h-ft2-degF)	LMTD	193.2	degF
Heat transfer area	220	ft2	Heat transfer rate	401.6	MBtu/h
Fouling factor	0.90		*Alternative formula:*	*401.6*	*MBtu/h*

Stream temperatures

Process in	150.0	degC
Process out	120.0	degC
Cooling water supply	68.0	degF
Cooling water return	95.0	degF



Calculations

Heat transfer coefficient	59.6	W/m2/K	LMTD	=LMTD(C25,C24,C26,C23)	K
Heat transfer area	20.4	m2	Heat transfer rate	117,628	W
Fouling factor	0.90				

Stream temperatures

Process in	423.2	K
Process out	393.2	K
Cooling water supply	293.2	K
Cooling water return	308.2	K

QuadraticFormula function

Description

Uses the quadratic formula to find roots of quadratic equations of the form:

$$0 = ax^2 + bx + c$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Syntax

```
QuadraticFormula( a,  b,  c,  [Root index] )
```

The QUADRATIC function syntax has the following arguments:

- **a** Coefficient of the x^2 term.
- **b** Coefficient of the x^1 term.
- **c** Coefficient of the x^0 term.
- **Root index** Optional. Indicates which of up to two roots to return.

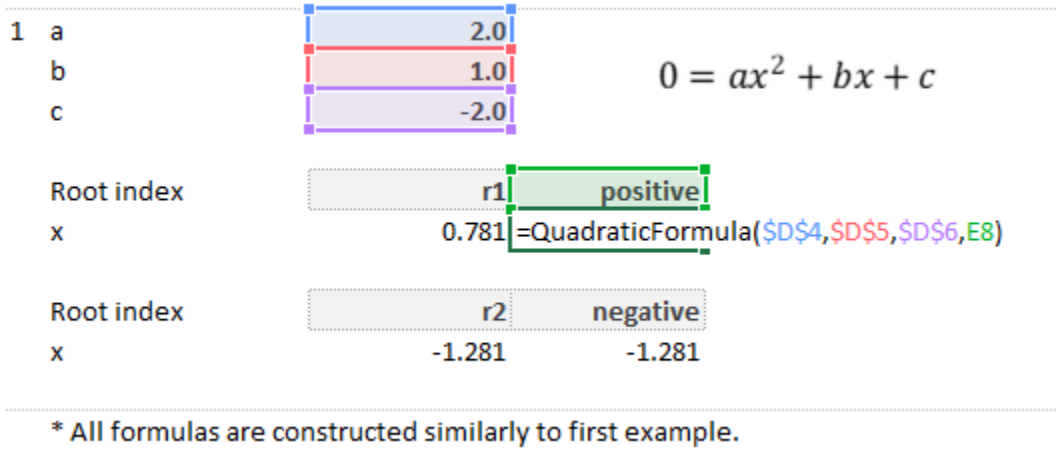
QuadraticFormula accepts the following case-insensitive text values (in quotation marks or as cell references) for Root index.

Root index	Description
r1	Returns the root that corresponds to (-b + sqrt(b2-4ac))/2a.
r2	Returns the root that corresponds to (-b - sqrt(b2-4ac))/2a.
pos, positive	If one positive and one negative root exist, returns the positive root, otherwise returns the #VALUE! error.
neg, negative	If one positive and one negative root exist, returns the negative root, otherwise returns the #VALUE! error.

Remarks

- If the input data types are incorrect, or the equation specified has no real roots, QuadraticFormula returns the #VALUE! error value.

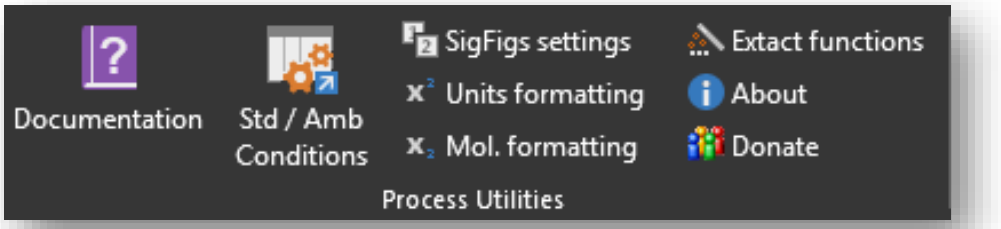
Example



Ribbon interface & shortcuts

Description

The Ribbon menu gives the user access to ProcessUtilities' documentation file and several other operations.

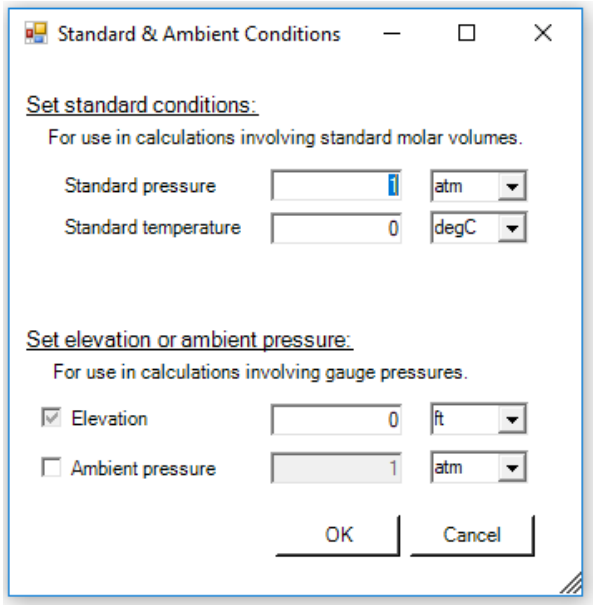


Submenus and commands	
Documentation	Opens this help file as a PDF file.
Std / Amb conditions	Opens window for setting custom values for standard pressure, standard temperature, and ambient pressure or elevation.
SigFigs settings	Allows the user to specify and apply rules for the number of significant figures used to display values in the spreadsheet.
Units formatting	Applies unit-style superscript formatting to selection.
Mol. formatting	Applies molecular formula-type formatting to selection.
Extract functions	Opens menu for extracting Process Utilities functions from the spreadsheet.
About	Displays licensing and other information about Process Utilities.
Donate	Links to Process Utilities donation page. You should try using this link!!
AutoConv shortcut	**Automatically inserts a Conv function and cycles between Conv, TConv, & PConv.**

Conditions Menu

Description

The Conditions menu allows the user to specify custom settings for standard pressure, standard temperature, elevation, and ambient Pressure. These are values that are used by Process Utilities functions to convert between standard volume and moles, and between ambient and gauge pressure.



Either elevation or ambient pressure can be specified, but not both. The value that is not specified is calculated using the equation below, where Po is pressure at sea level, L is temperature lapse rate, h is elevation, To is temperature at sea level, g is Earth-surface gravitational acceleration, M is the molar mass of dry air, and R is the Universal Gas Constant.

$$p = p_0 \cdot \left(1 - \frac{L \cdot h}{T_0}\right)^{\frac{g \cdot M}{R \cdot L}}$$

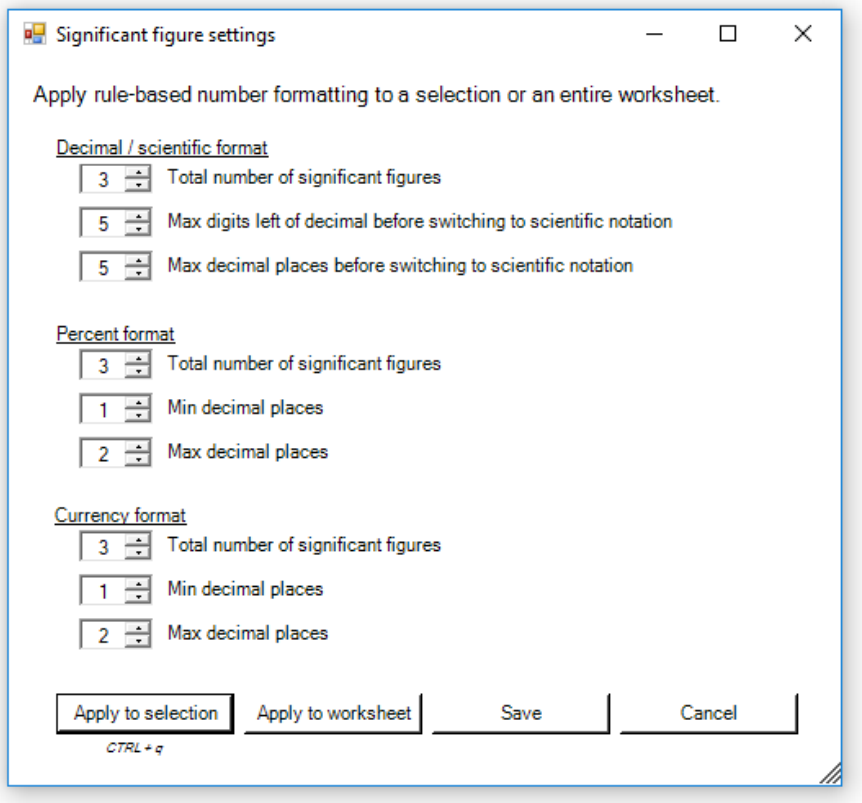
Setpoints specified in the Conditions menu are stored by Excel as named ranges and will be saved along with the worksheet.

SigFigs settings menu

Description

The SigFigs menu allows the user to specify rules for how numbers are displayed in a spreadsheet. Note that these settings only change how values are displayed, and not the values themselves. Setpoints specified in the SigFigs menu are stored by Excel as named ranges and will be saved along with the worksheet.

You can choose to apply settings to either a selection or to the entire workbook. **The settings can be applied to a selection anytime by pressing 'Ctrl + q'.**



Decimal / scientific format

These settings apply to values that are formatted in decimal or scientific notation. These are some examples of how values would be formatted according to the settings shown in the screenshot above.

54.235	→	54.2
0.0000001235	→	1.23E-07
43,234.342	→	43,234
12,235,2123.3	→	1.22E+07

Percent format

These settings apply to values that are formatted as percents. These are some examples of how values would be formatted according to the settings shown in the screenshot above.

1.023%	→	1.02%
0.00123%	→	0.00%
95%	→	95.0%

Currency format

These settings apply to values that are formatted as currency. These are some examples of how values would be formatted according to the settings shown in the screenshot above.

$1.023	→	$1.02
$0.00123	→	$0.00
$1,200	→	$1,200.0

Units formatting

Description

This command button applies units-style formatting to a selection – all numbers that follow letters are made superscript.

Examples

Unformatted	Formatted
W/m2/K	$W/m^2/K$
ft3/s	ft^3/s
kg-m2/s2	$kg\text{-}m^2/s^2$

Mol. formatting

Description

This command button applies molecular formula-style formatting to a selection – all numbers that follow letters are made subscript.

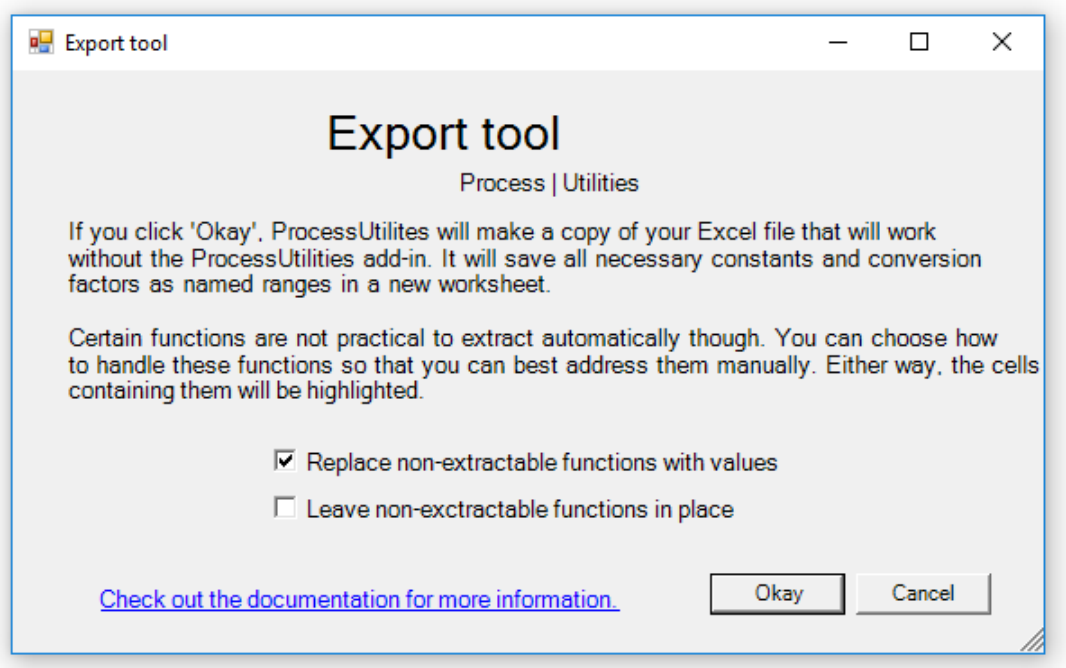
Examples

Unformatted	Formatted
H2O	H_2O
C6H6	C_6H_6
Na2SO4-10H2O	Na_2SO_4-$10H_2O$

Extract functions

Description

The Extract functions button opens the Process Utilities Export Tool. The Export Tool allows you to make a copy of your Excel file that will work without the ProcessUtilities add-in. It will save all necessary constants and conversion factors as named ranges in a new worksheet. This is useful when you want to share workbooks with people who might not have Process Utilities.



Extractable functions	
Conv	HeadPressure
TConv	PipeFlow
PConv	EquivalentLength
ConvFactor	Roughness
Dimensional	FlowRegime
Dimensionless	PipeSize
Constant	LMTD
STP	QuadraticFormula
MW	

AutoConv keyboard shortcut (Ctrl + j)

Description

This is a very useful shortcut which automatically inserts a **Conv** function. When 'Ctrl + j' is pressed, ProcessUtilities searches adjacent cells for a value/units pair. If it finds one, it will insert a **Conv** function below the cell containing the value. If no 'NewUnits' are specified to the right of this cell, then ProcessUtilities will insert the BaseUnits corresponding to the original units.

If the **AutoConv** shortcut is used on a cell that already contains a **Conv** function, then it will cycle this function from **Conv**, to **TConv**, to **PConv**, and back to **Conv**.

Examples