

ProGFE Formula Functions

ProGFE uses a different method of processing formulas than ProForm which allow among other things IF statements.

ProForm fields are entered in the standard ProForm {{SalePric}} format. To add 2 numbers use the Syntax {{SalePric}} + {{LoanAmt}}, rather than {{SalePrice + LoanAmt}}. There can be only one field inside any curly brackets {{FieldName}}

If lookup tables are to be used by both employees and ProGFE. Setup them up by using both ProForm formulas, and ProGFE formulas in different fields. Then Use the Power of ProGFE Custom Formulas to move amount fields from ProForm Text fields, to Hud Lines.

ProGFE AMS Plans do not include formula creation. We will assist with direction, but not creation.

Expression Operators

(,) Parenthesis used to indicate priority IE: $((1+1)*3) + 1 = 7$

< Less Than

<= Less Than or Equal To

> Greater Than

>= Greater Than or Equal To

<> Not Equal To

- Minus

+ Plus

* Multiply

/ Divide

% - percent (Not operational)

& - Concat Strings. Do not use + to Concatenate Strings

" " = String Container

Decimal numbers need to have a preceding 0, Ie 0.05 will work whereas .05 will fail.

Priority Order Low to High, parenthesis overrides this

None

Equality

Concat

PlusMinus

MulDiv

Percent

Numeric Functions

Rand()

Returns a 2 digit Random Number between 0 and 1

Rand() Returned :0.568684875965118

Rand() Returned :0.0783611536026001

Rand() Returned :0.790249049663544

Modu(Number1, Number2)

Returns the Modulus of Number1 and Number 2

Modu(7, 4) Returned :3

Abs("Value")

Returns the Absolute Value of "Value"

Abs(-234) Returned :234

Int("Value")

Returns the Integer portion of "Value"

Int(Now - Date(Year(Now),1,1))+ " days since new year"

Returns x days since New Year.

String Functions

Trim("string")

Returns a string with preamble and postamble spaces trimmed or removed

Trim(" String ") Returned :String

RightTrim("string")

Returns "string" with right side spaces removed

RightTrim(" String ") Returned : String

LeftTrim("string")

returns "string with Left side spaces removed

LeftTrim(" String") Returned :String

PadLeft("String", Wanted Length, "CharacterToAdd")

Pads the left side of "String" up to the "Wanted Length" using "CharacterToAdd" which is typically " "

PadLeft("String" , 20 , " ") Returned : String

PadRight("String", Wanted Length, "CharacterToAdd")

Pads the Right side of "String" up to the "Wanted Length" using "CharacterToAdd" which is typically " "

PadRight("String" , 20 , ".") Returned :String.....

Lower("String")

Returns Lower Case "String"

Lower("String") Returned :string

Upper("String")

Returns Upper case "String"

Upper("String") Returned :STRING

WCase("String")

Returns "String" with first character of first word Uppercase

Wcase("this is a TEST only a TEST") Returned :This is a test only a test

Replace("BaseString", "SearchString", "ReplaceString")

Returns the "SearchString" if found in "BaseString", with "ReplaceString"

Replace("asdfjkl", "asd", "xxxx") Returned :xxxxfjkl

SubStr("BaseString", FromNumber, Length)

Returns the characters of "BaseString", starting with character FromNumber, for Length Number of characters

SubStr("This is a String", 6, 24) Returned :is a String, Note: the length can be longer than string length SubStr("This is a String", 6, 2) would return "is"

Len("String")

Returns the Length of "String"

Len("String") Returned :6

Index("String", "CheckString")

Returns a zero index number where CheckString exists in String.

Index("Dan", "a") Returned :1

IndexFrom("String", "CheckString", StartPosition)

IndexFrom("Bill Backerson", "B", 3) would return a 5

Date Functions

Now()

Returns system date and time, which can be parsed with Substr or other String Functions.

Now Returned :2009/12/11 6:23:35 PM

Today()

Returns only today's date

Today Returned :2009/12/11

Date("Year", "Month", "Day")

Returns formatted Date from input strings

Date("2009", "12", "01") Returned :2009/12/01

Year("DateValue")

Returns only the year part of the date, as a number
Year(Now) Returned :2009

Month("DateValue")
Returns only the Month part of the date, as a number
Month(Now) Returned :12

Day("DateValue")
Returns only the Day part of the date, as a number
Day(Now) Returned :11

WeekDay("DateValue")
Returns the Day of the week as a Number from a DateValue
WeekDay(Now) Would return 1 to 7 depending on the day of the week
WeekDay(Now) Returned :6

SoftPro Emulated Functions

NotEmpty(Field)
Return True if "Field" is not Empty
NotEmpty({{Byr1Nam1}}) Returned :False
NotEmpty({{FirmFile}}) Returned :True

IsEmpty(Field)
Return True if "Field" is Empty
IsEmpty({{FirmFile}}) Returned :False
IsEmpty({{Byr1Nam1}}) Returned :True

Rnd2(number)
Rounds a number to 2 decimal places
Rnd2(100.2149) Returned :100.21
Rnd2(100.2150) Returned :100.22

Int() - Covered above

TrimU(x) - Trims extra spaces from start and end, and converts string to uppercase
Trimu(" This is a string to uppercase and trim ") Returned :THIS IS A
STRING TO UPPERCASE AND TRIM

SubStr - Covered Above

Index - Covered Above

Other Functions

ifelse(Condition, TrueValue, FalseValue)
Ifelse is a nestable IF Statement, Values can be Strings, or Numeric.
ifelse({{SalePric}} > 100000 , 500 , 400) - If the Sales price is over
100,000 put in 500, otherwise put in 400.

`ifelse({{SalePric}} > 1000000 , 2000 , ifelse({{SalePric}} > 500000 , 1500 , ifelse({{SalePric}} > 100000 , 1001 , 500)))` If the Sales price is over 1 Million put 2000, if it's between a half, to one million put 1500, if it's between 100,000 and 499,999.99 put 1001, and finally if it's below 100000, put 500.

`ChCR()`

Returns Carrage Return Character

`CHLF()`

Returns LineFeed Character

`ChCrLf()`

Returns Carrage Return / LineFeed Characters together.

Undocumented, unsupported functions

These are here for your enjoyment, use them at your own risk. If you use them let dan@softprosupport.com know, and I'll consider adding them to the documented list. If there is a function you need that's not listed, ask for it, our best ideas come from you.

Format(value, Style) formats a value in the .NET style

Lower(String) - Returns the lower case value of String

Upper(String) - returns the uppercase value of STring

Min(v1,v2,v3,v4,v5) returns the minimum

Max(v1,v2,v3,v4,v5) returns the maximum

InList(Search, List, delim) Search string for match to delimited list List = "1:2:3

delim = ":"

FileExists(filename)

FileInfo(fileName)

Long_date(date) returns 11th December 2009. (Note note 11th day of December 2009)

Money(value) returns a string formatted as Money from a number

Format_Date(now, "w x d e f m n o p z" returns Friday, fri, 11th, 11, 11, December, 12 , 12, Dec, 09) where f,o and z are padded to 2 characters

Sqrt(Value) returns square root of value

Power(Value, Exponent) raises Value to Exponent Power(10,3) = 1000

and - AND - Used in ifelse statements IE: ifelse({{SalesPrice}} > 100000 and {{LoanAmount}} > 90000 , "SpecialTrue" , "SpecialFalse")

or - OR

not - NOT

or

and

not