

New FINCAD features make it easier to diagnose input errors

Introduction

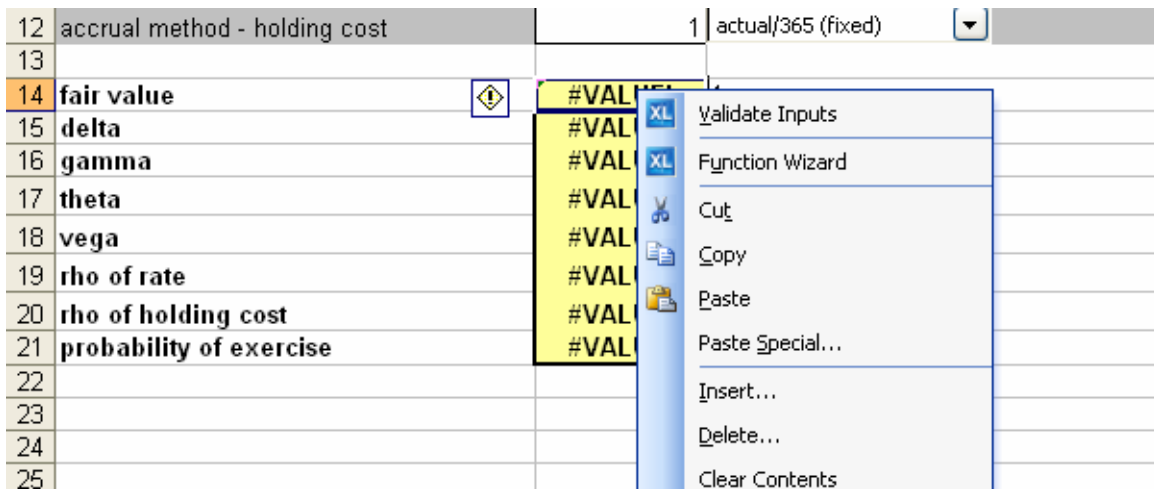
Recent versions of FINCAD products have a number of new features which make it easier than ever before to diagnose and correct input problems. We will look at the features in the FINCAD XL product first and then later describe similar features which have been added to the FINCAD Developer product.

Given a correct set of input arguments, a function will calculate and return results. However, if some input data is incorrect or the calculation could not be performed due to a particular combination of input values, an error condition will arise (i.e., #VALUE!).

Validate Inputs feature

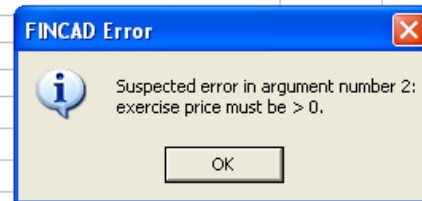
There is a new input validation feature available in FINCAD XL that can help in this situation. To use this feature, go to the cell with the function, i.e., the highlighted yellow cell on a pasted function and choose the 'Validate Inputs' menu item to validate the formula. You can access this menu item in two ways. The first way is through the FINCAD XL menu (FINCAD XL->Function Error Handling->Validate Inputs) and the second way is by right-clicking over the cell with the formula and selecting the option from the context menu that pops up.

The image below shows the second approach.



For a pasted example, the "Validate Inputs" feature will display a message box with a description of the input problem and will also highlight the cell which contains the invalid input in pink. An example is included below for the aaBSG function.

	A	B	C	D	E
1	aaBSG				
2	underlying price	14.5			
3	exercise price	-1			
4	expiry date	16-Mar-2000			
5	value (settlement) date	24-Sep-1999			
6	volatility	20.000%			
7	rate - annual compounding	8.000%			
8	holding cost - annual compounding	6.000%			
9	option type	1 call			
10	statistic	At bottom of pasted example			
11	accrual method - riskless rate	1 actual/365 (fixed)			
12	accrual method - holding cost	1 actual/365 (fixed)			
13					
14	fair value	#VALUE!	1		
15	delta	#VALUE!	2		
16	gamma	#VALUE!	3		
17	theta	#VALUE!	4		
18	vega	#VALUE!	5		
19	rho of rate	#VALUE!	6		
20	rho of holding cost	#VALUE!	7		
21	probability of exercise	#VALUE!	8		
22					
23					
24					



In the above case, the exercise price should be a positive number. When a correct set of inputs are used, the "Validate inputs" feature will return "Validation OK" and the results of the function call will be contained in the formula cells.

Note:

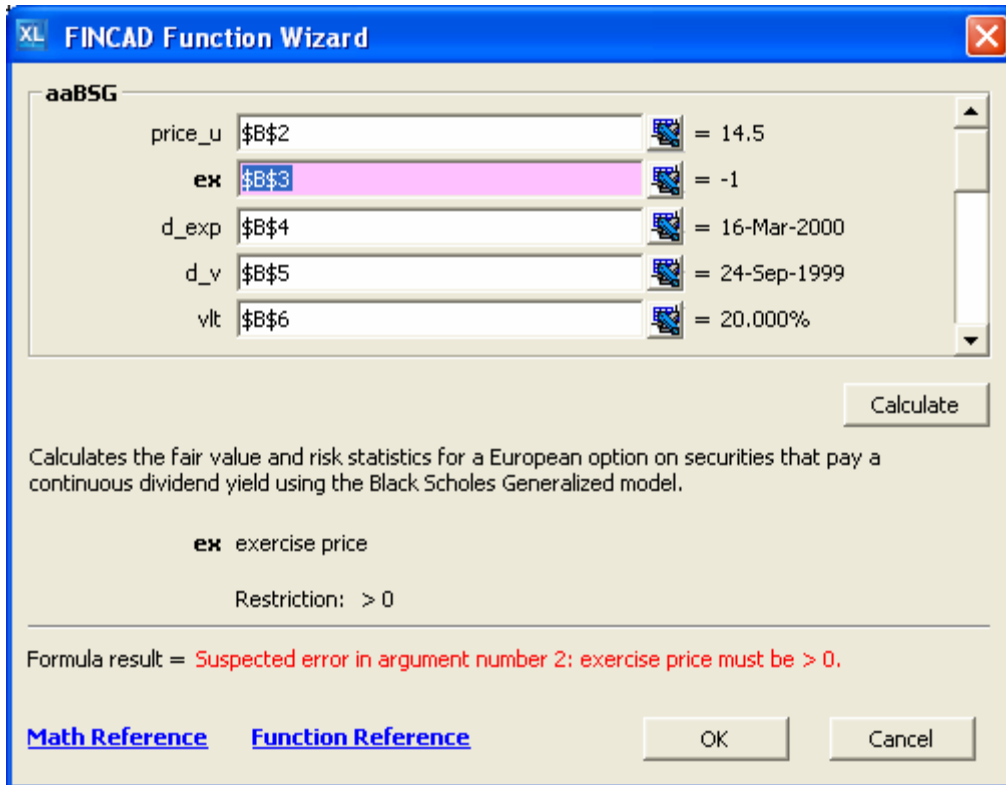
In its current form, the Validate Inputs feature does not handle nested function calls, i.e., functions embedded within other functions. The best way to work around this limitation is to go to the formula and see if it is possible to separate some of the nested functions. If there are multiple FINCAD functions combined together, you may separate the various functions and then individually validate them.

FINCAD Function Wizard

Input problems can also be diagnosed using the new "Function Wizard". FINCAD has developed its own Function Wizard. It provides more information and features than the standard Excel Function Wizard. You can access the FINCAD Function Wizard in two ways. The first way is through the FINCAD XL menu (FINCAD XL->Function Wizard) and like the "Validate Inputs" feature, you can also access this tool by right clicking and using the context menu (as you can see from the first screenshot in this article).

When the formula that is being viewed by the Function Wizard is failing to calculate, you can use the Calculate button to calculate the function from within the Wizard and it will display the relevant error text and also highlight the problem input in pink.

The example below shows a failed calculation for the aaBSG function after the Calculate button was pressed.



More information and features than the XL Function Wizard

The middle section of this Function Wizard dialog contains a long description of the input that is currently selected as well as information about restrictions for the input. In the case above, we can see that the input should be greater than zero and the information in red from the formula result tells us that this is not the case. Also, the Fincad Function Wizard provides shortcuts which allow you to quickly open the related Math or Function Reference page without having to use the Function Finder.

What's new with validation in Fincad Developer

Now that we have gone through the new Input Validation features in the Fincad XL product, let us look next at how you can diagnose an input problem using the Fincad Developer product. The first thing to note about Input Validation in Fincad Developer is that the Function Reference pages in Fincad Developer contain content that is not in the Fincad XL Function Reference pages. Let us start by looking at a typical Function Reference page in Developer. Below is a screenshot which shows part of a Function Reference page for the function aaAccrued

C# Example
 iStatus = FincadFunctions.aaAccrual_factor(d_e, d_t, acc, ref return_acc_fac)

Returns	Type	Description	Sample	Restriction
acc_fac	double	accrual factor - day basis/year basis		
Argument	Type	Description	Sample	Restriction
d_e	double	effective date	3-Jan-2000	(#1000)
d_t	double	terminating date	9-Mar-2000	>= d_e (#2503)
acc	int	accrual method (sw_331) 1 - actual/365 (fixed)	2	1..15 (#3795)

Error codes on the Function Reference pages

As you may notice, there are restriction codes (i.e. error codes) displayed along the right side. For example, if the terminating date is less than the effective date then the function will return an error code with the number 2503. When the function returns this error code you can go to the Function Reference page to find out what it means. Also, the first digit of the error code tells us that the problem relates to the second input for the function. The remaining three digits are used by FINCAD to identify more specifically, what is wrong with the input. A list of the internal three digit codes can be accessed through the help menu in the Function Finder dialog (Help->Developer Reference->Error Codes).

Getting the error text for an error code at runtime

In v10 it is also possible to retrieve the error text associated with the error code by passing the error code into the function `GetErrorString`. `GetErrorString` is currently available in the interfaces for C, C++, C#, VB.NET and Java.

The C version of the declaration for this routine is below:

```
void GetErrorString(unsigned int errorCode, const unsigned int len, char * errorStr);
```

You need to create a buffer (i.e. string) which will hold the error string and pass a pointer to this buffer along with its length when calling `GetErrorString`. The error text will be truncated if it turns out to be longer than your buffer.

Here is an example of what a call to `GetErrorString` looks like in C:

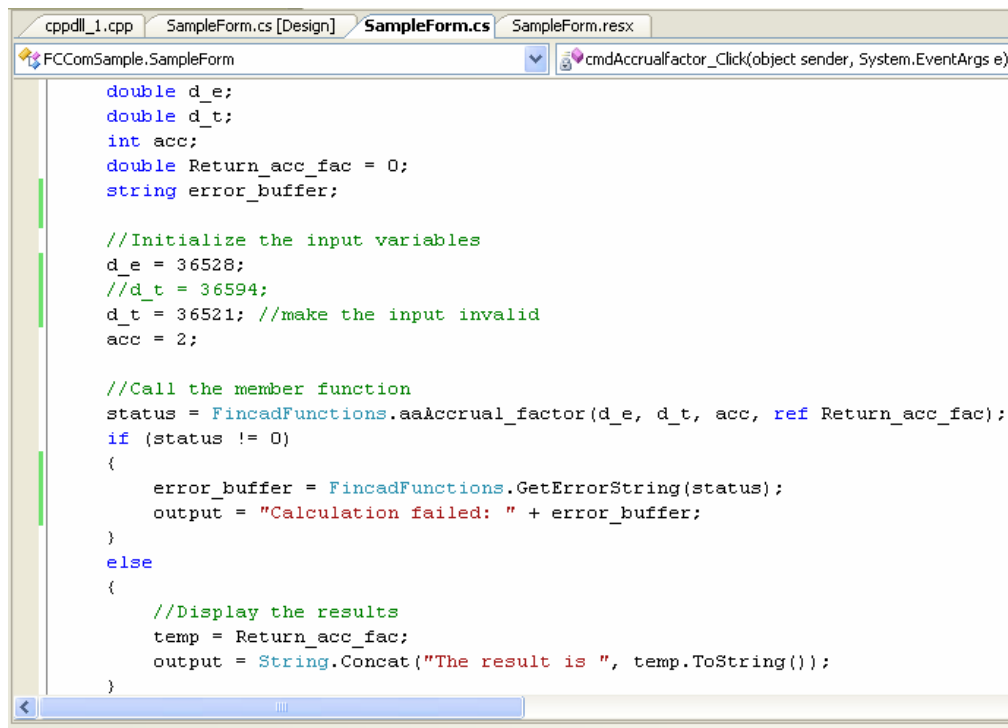
```
GetErrorString((int)(Return_cf_tbl->val.err), 4095, strOutput);
```

Using GetErrorString in the object oriented languages

For object oriented languages the string object handles sizing and resizing of the buffer and therefore the length input is not needed and in C++, `GetErrorString` is used internally by the FINCAD C++ wrapper library when a routine throws an exception. It provides the mechanism for setting the exception text. So, you don't need to call it directly in C++, when the C++ wrapper throws an exception you will get the error text.

Error handling example for C#

Below is an example of a simple C# program which uses `GetErrorString` to display an error message. The terminating date in this code example has been intentionally set to an invalid value, in order to demonstrate the `GetErrorString` routine.



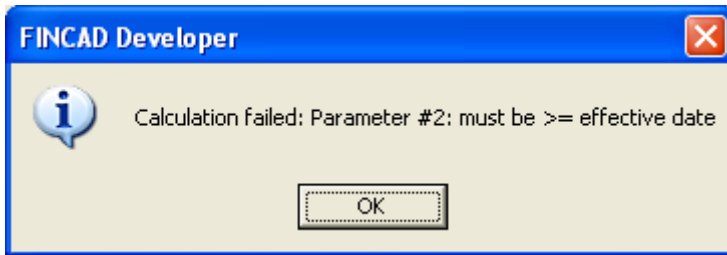
```
cppdll_1.cpp  SampleForm.cs [Design]  SampleForm.cs  SampleForm.resx
FComSample.SampleForm  cmdAccrualfactor_Click(object sender, System.EventArgs e)

double d_e;
double d_t;
int acc;
double Return_acc_fac = 0;
string error_buffer;

//Initialize the input variables
d_e = 36528;
//d_t = 36594;
d_t = 36521; //make the input invalid
acc = 2;

//Call the member function
status = FincadFunctions.aaAccrual_factor(d_e, d_t, acc, ref Return_acc_fac);
if (status != 0)
{
    error_buffer = FincadFunctions.GetErrorString(status);
    output = "Calculation failed: " + error_buffer;
}
else
{
    //Display the results
    temp = Return_acc_fac;
    output = String.Concat("The result is ", temp.ToString());
}
```

Running the code example above will display a message like the one below.



Run-time error messages like this can help users of your application diagnose and fix input problems quickly.

Disclaimer

Your use of the information in this article is at your own risk. The information in this article is provided on an "as is" basis and without any representation, obligation, or warranty from FINCAD of any kind, whether express or implied. We hope that such information will assist you, but it should not be used or relied upon as a substitute for your own independent research.

Copyright © 2007 FinancialCAD Corporation. All rights reserved. FinancialCAD® and FINCAD® are registered trademarks of FinancialCAD Corporation. Other trademarks are the property of their respective holders. This email is for informational purposes only. FinancialCAD Corporation MAKES NO WARRANTIES, EXPRESSED OR IMPLIED, IN THIS SUMMARY.