
Using Metrology.NET[®] to Calculate ISO/IEC 17025 Uncertainties for Fluke MET/CAL[®]

Michael L. Schwartz
Cal Lab Solutions, Inc.

Writing automated calibrations and controlling test equipment is easy. But once you have the automation working, now it's time to calculate the uncertainties. Often calculating uncertainties and documenting the results takes longer than writing the actual procedure. Then there are the changes to the calculations, changes in customer requirements, as well as industry regulations and standards. All of this takes time away from creating automation.

This paper will show you how to offload the task of calculating measurement uncertainties to a Metrology.NET service call, and how this division of tasks in software allows the programmer to focus on writing automation and the quality/metrology engineer to focus on uncertainties. By using a REST based service call, the quality engineer doesn't have to learn programming, nor is the programmer required to learn all the details about the uncertainty calculation. Best of all, this REST call technique can be added to any software language, such as MET/CAL[®], SureCal[®], LabView[®] or any Metrology.NET[®] system.

Introduction

First we have to start with "What is a REST call, and why it is important?"

Representational state transfer (REST) or RESTful web services is a way of providing interoperability between computer systems on the Internet. REST-compliant Web services allow requesting systems to access and manipulate textual representations of web resources using a uniform and predefined set of stateless operations... In a RESTful Web service, requests made to a resource's URL will elicit a response that may be in XML, HTML, JSON or some other defined format. The response may confirm that some alteration has been made to the stored resource, and it may provide hypertext links to other related resources or collections of resources. Using HTTP, as is most common, the kind of operations available include those predefined by the HTTP verbs GET, POST, PUT, DELETE and so on. [1]

So what does all this mean to a metrologist, in layman's terms? Most people view the internet through a web browser and think of all the content on the worldwide web as documents and web pages. But to a computer all of that data is just 1 and 0s. With Web services and RESTful services, we can greatly increase precision and accuracy of things like uncertainty calculations, while at the same time decrease the manpower required. By adding just one simple RESTful service call for calculation uncertainties, we can centralize all of our uncertainty calculations in one single

place. Then all of the lab's metrology software can call the same uncertainty calculator. This centralized REST based approach makes it extremely simple to verify all reported uncertainties comply with the lab's Scope of Accreditation.

This brings us to the core system-of-systems design of Metrology.NET and how it embraces a distributed computing model of shareable resources. By sharing a resource, other systems and other software packages are able to consume the shared resource. This allows an organization to consolidate all of their uncertainty calculations in a single location, no matter the programming language.

Creating a Standard

The most important part of a system-of-systems design is its modularity and ability to keep up with changing technologies! The architecture has been prebuilt, allowing labs to replace our uncertainty calculator with any future uncertainty tool that complies with the REST based communication standard. Think about that for a moment: Who designs a system that from day one the team is thinking about how to replace it with something better, something that hasn't even been invented; that is a true system-of-systems design.

The creation of a metrology based communications standard has been the goal of the key members of the National Conference of Standards Laboratories International (NCSLI) Measurement Information Infrastructure (MII) group for a number of years. We are the first company to define a Metrology.NET communication standard that

Sym	Contributor	A/B	Nominal	Limits	Units	Distribution	Divisor	Sensitivity	Standard Unc Value	Standard Unc Units
STD	Fluke 5720 Error	B	10.00E+0	73.5E-6	V	Normal	2	1	36.75E-6	V
TRC	Fluke 5720 Trace Error	B		9.63E-06	V	Normal	2	1	4.81E-6	V
RES	Fluke 5720 Resolution	B		70.0E-12	V	Rectangle	1.732	1	40.42E-12	V
UUT_Res	UUT Resolution	B		100.0E-9	V	Rectangle	1.732	1	57.74E-9	V
REP	UUT Repeatability	A		000.0E+0	V	Normal	2	1	000.00E+0	V
ENV	Env	B		0.00E+0	V	Rectangle	1.732	1	000.00E+0	V

Combined Standard Uncertainty :	37.1E-6	V
Coverage Factor k :	2	V
Expanded Uncertainty :	74.1E-6	V

Divisor and Distribution	Type
6 = 2.449 Triangle	B
3 = 1.732 Rectangle	B
2 = 1.414 U-Shaped	B
1 = 1.000 Sigma	A
2 = 2.000 Normal (k=2)	A/B
3 = 3.000 Normal (k=3)	B

Figure 1.

performs RESTful service calls to an uncertainty server. When we started this project, we wanted something very simple and easy to implement for any software platform. We wanted to simply tell the server the calculator to use and pass it a set of name value pairs. If the server is able to successfully calculate the uncertainties, it will return the uncertainty and the revision number of the calculator that it uses.

The first objection I always get is “Each uncertainty calculator is different! You can’t standardize it, because it is different!” Correct, each uncertainty calculation is different and will require a different set of name value pair inputs, but that doesn’t mean we can’t standardize the interface! The key to this design is the simple name, value pairing to the interface for all uncertainty calculations. This interface is simple and easy, as well as infinitely expandable for all measurement disciplines. The consumer of the REST service can pass as few or as many name value pairs as they want. The server will either complete the calculation or fail by returning “39e39” as the result. SIMPLE!

Handling Uncertainty Calculations

One of the goals we had, when we were designing the system, was to address the disconnect between a lab’s scope of accreditation uncertainties and the uncertainties calculated by the automation software. Many labs have learned the hard way when an auditor writes them up because their automated

calibration software produces uncertainties less than their accredited values. Having all your uncertainties calculations in one system or systems could alleviate this problem!

In this example, I want to keep things pretty simple. I know there are several uncertainty gurus out there who know uncertainties forward and backwards. But, the goal of this paper is to provide an example of how to call external uncertainty calculations from a MET/CAL procedure. It is important to note, this example will work with any version of MET/CAL all the way back to version 6.0 and above. And, it does not require a copy of Microsoft Office or Excel to be installed on the local workstation.

First, we have to create our uncertainty calculation (Figure 1). We will use a very basic one we created around a Fluke 5720A DC Voltage Accuracy.

Once we have a spreadsheet with all the lab’s numbers, we have to add the Metrology.NET. To do this, we need to add a Metrology.NET interface tab to the spreadsheet.

Contributors	Value	Contributor Type
Volts	10	Required
Repeatability	000.0E+0	Optional
Resolution	100.0E-9	Optional

Figure 2.

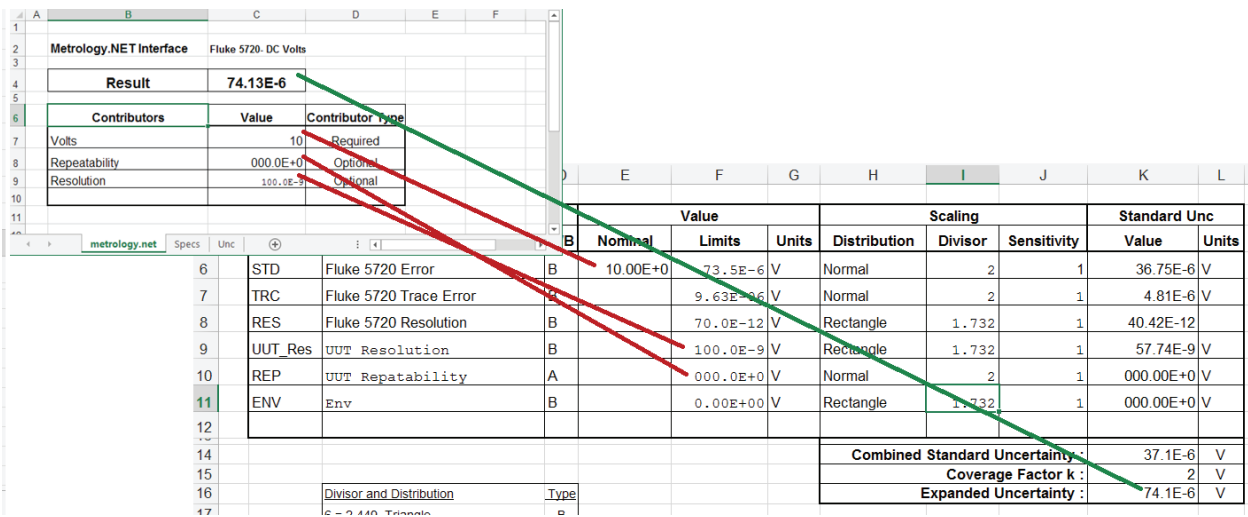


Figure 3.

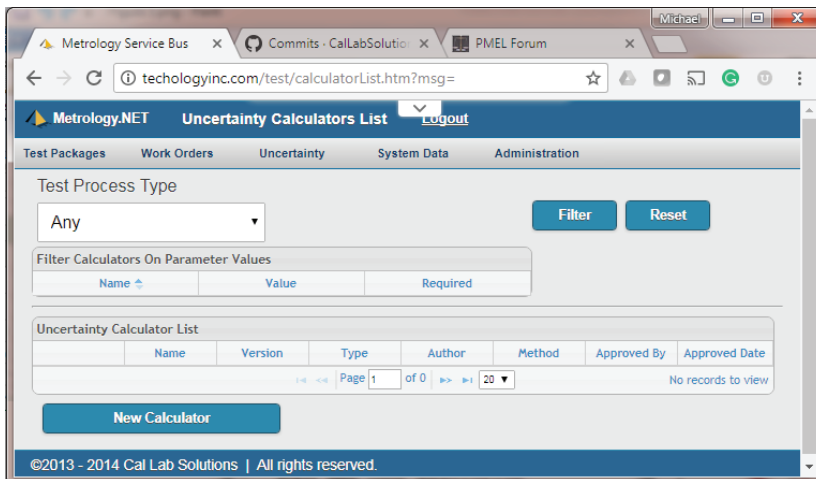


Figure 4.

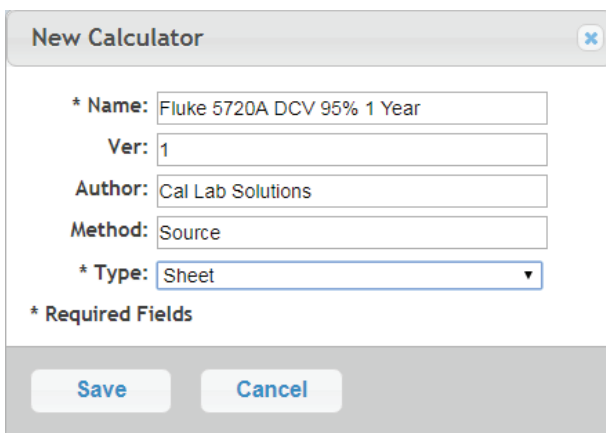


Figure 5.

When the spreadsheet is uploaded to the server, it will use this tab as the calculator's inputs, run the calculation, and return the results.

The inputs in the Metrology.NET tab (see Figure 2) are specific to the measurement process. They can be required or optional. Values can be text or numeric. If an input is required, then the consuming software must provide the value in the name value pairs passed to the calculator. If a required parameter is not provided, the server will return "39e39." If an input is listed as optional, a default value in the spreadsheet will be used. In this example, we are listing optional repeatability and resolution.

Next, we must wire the inputs in the Metrology.NET tab to the other cells in the spreadsheet. The server will enter the values in the input cells, and the values will propagate throughout the spreadsheet, until it eventually updates the calculation results field.

In the Fluke 5720A Example Spreadsheet (Figure 3), you can see the Volts Value in cell "Metrology.NET:C7" is linked to the "Unc:E6" cell. By changing the Volts Value, it updates and changes the E6 and F6 cells and recalculates the total uncertainties. The same thing is true for the Repeatability C8 and Resolution C9 cells. When the values in cells C7-C9 are updated, the calculations are propagated through the spreadsheet and the Expanded Uncertainty is updated in cell "Unc:K16," then copied to the resulting value to the linked cell "Metrology.NET:C4."

The Metrology.NET tab's purpose is to define a standard way of interfacing with any spreadsheet-based uncertainty calculator. There is a little more work involved in this methodology, but the advantage is the system can now use any spreadsheet with any number of Input Contributor Values, all standardized.

Creating a Calculator

Next, we need to upload the spreadsheet to the server so we can use it in multiple applications. To do this, we need to navigate to the Uncertainty Calculators List Page (Figure 4) then press the "New Calculator" button. Fill in the information for the New Calculator, being sure to select Sheet as the type (Figure 5).

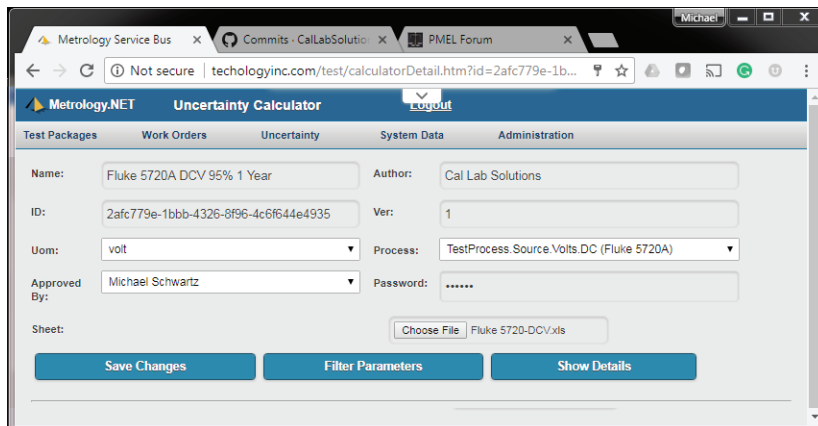


Figure 6.

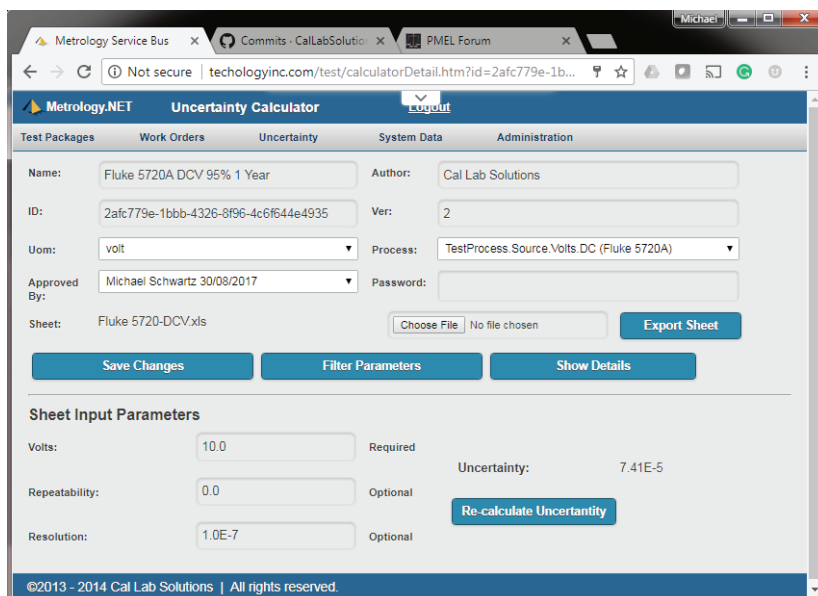


Figure 7.

Information about the spreadsheet will need to be uploaded to the server (Figure 6). The server needs to know the Units of Measure (UOM) for the spreadsheet and if it is attached to a specific metrology process—in this case "TestProcess.Source.Volts.DC (Fluke 5720A). Once you "Choose File" and select the spreadsheet to be uploaded, you can press "Save Changes."

The online uncertainty calculator is now ready for use. Notice the "Sheet Input Parameters" have been read from the spreadsheet (Figure 7). By changing the inputs and pressing "Re-Calculate Uncertainty," the server is able to use the spreadsheet to calculate any measurement uncertainty needed for the Fluke 5720A DC Voltage.

Now to save the Metrology.NET Calculator, we must make note of the calculator's ID. In this case the ID is "2afc779e-1bbb-4326-8f96-4c6f644e4935." We will need this ID later in our MET/CAL procedure. Note: These IDs are randomly created; they are a little ugly, but will guarantee you will always get the exact calculator when called. Also, make a mental note of the revision number of the calculator. All the changes made to this calculator are not under revision control. This is handy if you want to compare the results of two different versions of a calculator.

Performing Calculations

This example requires the McNetComm.exe and DosDoseDat.dll to communicate between MET/CAL and the Metrology.NET server. McNetComm.exe is a free software tool from Cal Lab Solutions, you can download it and many other MET/CAL tools from FTP://ftp.CalLabSolutions.com/FTP. These files must be placed in the workstation's User Programs Directory.

The first thing we need to do is setup a login and password for the MET/CAL station. This will allow the workstation to securely communicate with the server. In this example, an account is set up for the MET/CAL Station, using:

Login: MC_Station1
 Password: METCAL

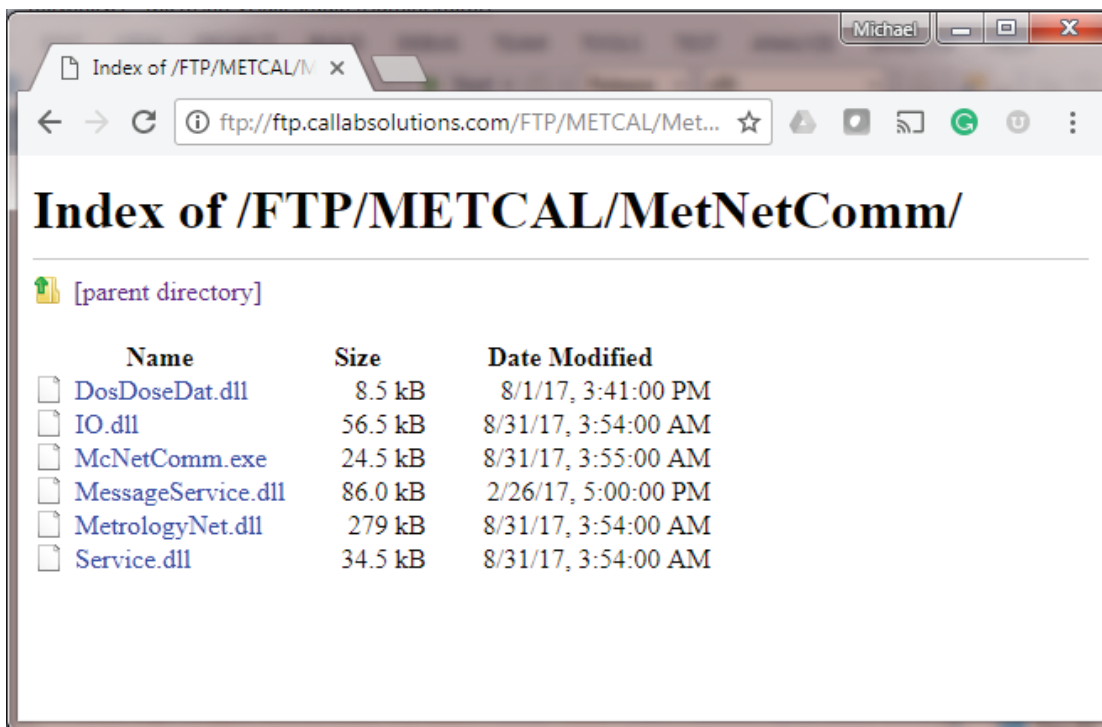


Figure 8.

The following lines of code will allow the MET/CAL workstation to log-in:

```

=====
#   Login in to the Metrology.NET Server
=====
1.001 MATH      MEM2=  "URI='http://xxx.Metrology.NET/
Server/'"
1.002 MATH      MEM2=MEM2&", Login=' MC_Station1"
1.003 MATH      MEM2=MEM2&", Password='METCAL"
1.004 DOS       McNetComm.exe Login
1.005 IF        Find(MEM2,"Error",1)
1.006 DISP      Error! Could not Connect the the Server
1.007 END
1.008 ENDIF
    
```

If the connection to the server is successful, MEM2 will contain the word "Connected" and the station can make any uncertainty calculations.

Now, all we have to do is load MEM2 with the Name Value pairs required by the uncertainty calculator. Simply separate Name Value pairs with commas: "Volts= 10, Resolution= 1e-5," etc. Note: You can pass more Name Value pairs than is required by the uncertainty calculator.

We also need to know the calculator's ID so we can select a specific calculator. In this example, we store that value in @UncCalcID. If we wanted a specific version of

the calculator, we would add it to the end with a space like this @UncCalcID="2afc779e-1bbb-4326-8f96-4c6f644e4935 2" We recommend leaving the specific version number off because it allows you to update uncertainty calculations without having to modify your code.

To call the calculator, we need the following lines of code:

```

=====
#   Calculate the Uncertainties for 10 V
=====
1.009 MATH      @UncCalcID ="2afc779e-1bbb-4326-8f96-
4c6f644e4935"
1.010 MATH      MEM2="Volts= 10"
1.011 MATH      MEM2=MEM2&", Resolution= 1e-5"
1.012 MATH      MEM2=MEM2&", Repeatability= 2.4e-4"
1.013 DOS       McNetComm.exe CalcUnc [V @UncCalcID]
1.014 MATH      MEM = MEM2
1.015 TSET      EXP_UNC = [MEM]
    
```

The resulting call to Metrology.NET will return the following string "0.0002515." We need to use that value in our MET/CAL code. I find the easiest way is with the TSET or TOL FSC. I prefer the TSET EXP_UNC, because it will override any and all MET/CAL uncertainty contributors, as well as allow you to use the TOL for UUT test limits.

Next, we can mix it with a Fluke 5720 call using the IEEE FSC. This example brings in all together:

```
#####
#   Setup the Fluke 5720 & Calculate Uncertainties
#####

1.016 MATH      @Volts = 10

1.017 IEEE      [@5720]OUT [V @Volts]V, 0Hz
1.018 IEEE      [@5720]OPER
1.019 IF        L[1]>23
1.020 IEEE      [@5720]*CLS
1.021 IEEE      [@5720]OPER
1.022 ENDIF
1.023 IEEE      [@5720]*OPC?[i!]

1.024 MATH      @UncCalcID ="2afc779e-1bbb-4326-8f96-
4c6f644e4935"
1.025 MATH      MEM2="Volts= " & @Volts
1.026 MATH      MEM2=MEM2&"", Resolution= 1e-5"
1.027 MATH      MEM2=MEM2&"", Repeatability= 2.4e-4"
1.028 DOS       McNetComm.exe CalcUnc [V @UncCalcID]
1.029 MATH      MEM = MEM2
1.030 TSET      EXP_UNC = [MEM]
```

I know many MET/CAL programmers out there like to use the 5720 & M5720 FSC, but we like to use IEEE FSCs. Mostly because it allows us to use any DC Voltage calibrator without having to modify the UUT Code. If you like using the Fluke 5720 FSC, you can use it in place of the IEEE calls—it will still all work.

Conclusion

Yes, I know MET/CAL is fully capable of calculating expanded uncertainties. And, yes, I know the LIB FSC can interface with Microsoft Excel. So why go through the trouble of setting up a Metrology.NET server? The bigger issue is how to keep your entire lab's uncertainty calculations up-to-date and in sync with the lab's Scope of Accreditation. The process we've explained in this article allows you to use the same spreadsheet you are showing your accreditation body. Plus, if you are using more than just MET/CAL, you can call the same uncertainty calculator from other applications. The below example is a VB.NET application making the same call.

```
If Agent.LogIn("MC_Station1", "METCAL") = True Then
```

```
    'Create Parameters List
    Dim Params As New List(Of Parameter)
    Params.Add(Parameter.NewParameter("Volts", "10"))
    Params.Add(Parameter.NewParameter("Resolution", "1e-5"))
    Params.Add(Parameter.NewParameter("Repeatability", "2.4e-4"))

    ' Create Link to Specific Calculator
    Dim UncCal As New Uncertainty("2afc779e-1bbb-4326-8f96-
```

```
4c6f644e4935", "", Params)
    UncCal.Calculate()
```

```
    ' See the Unc Calculation
    MsgBox("    Unc= " & UncCal.Uncertainty & vbCrLf & _
        "    Using= " & UncCal.UncertaintyID & vbCrLf & _
        "    Version= " & UncCal.Version, vbOK)

Else
    Throw New Exception("Error Logging on to the Server!" &
vbCrLf & _
        "Unknown User Name and/or Password!")
End If
```

The even bigger picture of Metrology.NET is the creation of a system-of-systems solution for metrology. This example was system-of-systems approach to calculating uncertainties, using Fluke MET/CAL and Excel, which can be expanded to use VB.NET, C#, LabView, etc. So, long after Windows and long after Excel or MET/CAL are viable products, we will be performing REST based uncertainty calculations.

References

- [1] Wikipedia, "Representational state transfer," https://en.wikipedia.org/wiki/Representational_state_transfer (accessed 08/31/2017).

Michael L. Schwartz (mschwartz@callabsolutions.com),
CEO, Cal Lab Solutions, Inc., Aurora, Colorado, USA.