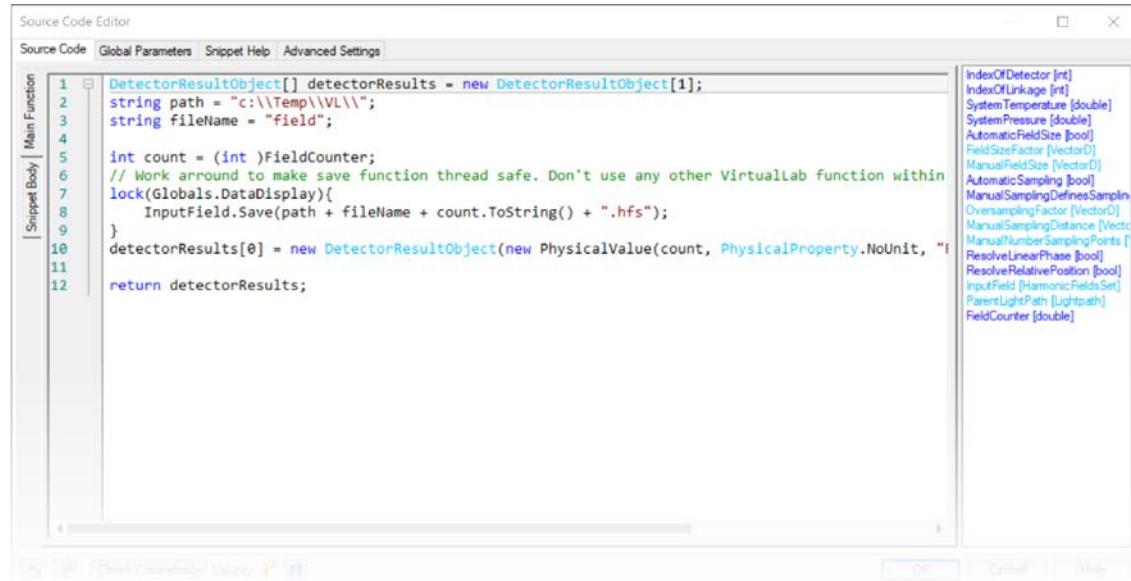




Programming a Detector to Save Fields Automatically

Abstract



The screenshot shows a Source Code Editor window with the following C# code:

```
1 DetectorResultObject[] detectorResults = new DetectorResultObject[1];
2 string path = "c:\\Temp\\VLA\\";
3 string fileName = "field";
4
5 int count = (int)FieldCounter;
6 // Work around to make save function thread safe. Don't use any other VirtualLab function within
7 lock(Globals.DataDisplay){
8     InputField.Save(path + fileName + count.ToString() + ".hfs");
9 }
10 detectorResults[0] = new DetectorResultObject(new PhysicalValue(count, PhysicalProperty.NoUnit, "I"));
11
12 return detectorResults;
```

The code defines a method that saves a series of detector results to a file on the hard disk. The file name is automatically generated by appending the counter value to the base file name.

When performing series of time-consuming optical simulation tasks, it is helpful to have the function of saving the results automatically. In this example, we construct a Programmable Detector which enables the automated saving of a light distribution (harmonic fields set) to the desired file path on the hard disk. The saved file name can be automatically generated using the counter value in the detector.

Task Description & Sample Code

Task:
Use Programmable Detector to automatically save light distribution (harmonic fields set) to the hard disk.

The screenshot shows a Source Code Editor window with the following code:

```
DetectorResultObject[] detectorResults = new DetectorResultObject[1];
string path = "c:\\Temp\\VL\\";
string fileName = "field";

int count = (int )FieldCounter;
lock(Globals.DataDisplay){
    InputField.Save(
        path + fileName + count.ToString() + ".hfs");
}

detectorResults[0] = new DetectorResultObject(
    new PhysicalValue(count, PhysicalProperty.NoUnit,
    "File Index"), "Field Save Detector");
return detectorResults;
```

Global Parameters (User Defined)

Variable	Value	Allowed range
double FieldCounter	0	0 - 100000000

Main Function

```
DetectorResultObject[] detectorResults = new
DetectorResultObject[1];
string path = "c:\\Temp\\VL\\";
string fileName = "field";

int count = (int )FieldCounter;
lock(Globals.DataDisplay){
    InputField.Save(
        path + fileName + count.ToString() + ".hfs");
}

detectorResults[0] = new DetectorResultObject(
    new PhysicalValue(count, PhysicalProperty.NoUnit,
    "File Index"), "Field Save Detector");
return detectorResults;
```

The screenshot shows a Windows File Explorer window with the following file list:

Name	Date modified	Type
field0	9/21/2018 5:43 PM	HFS File
field1	9/21/2018 5:43 PM	HFS File
field2	9/21/2018 5:43 PM	HFS File
field3	9/21/2018 5:43 PM	HFS File
field4	9/21/2018 5:43 PM	HFS File
field5	9/21/2018 5:43 PM	HFS File
field6	9/21/2018 5:43 PM	HFS File
field7	9/21/2018 5:43 PM	HFS File
field8	9/21/2018 5:43 PM	HFS File
field9	9/21/2018 5:43 PM	HFS File
field10	9/21/2018 5:43 PM	HFS File

3

Document Information

title	Programming a Detector to Save Fields Automatically
document code	CZT.0063
version	1.0
toolbox(es)	Starter Toolbox
VL version used for simulations	7.4.0.49
category	Feature Use Case
further reading	<ul style="list-style-type: none">- Programming a Degree of Coherence Detector