

Programming a Detector to Save Fields Automatically

Abstract

```
Source Code Editor
Source Code Global Parameters Snippet Help Advanced Settings
Main Function
1 DetectorResultObject[] detectorResults = new DetectorResultObject[1];
2 string path = "c:\\Temp\\VL\\";
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;
Snippet Body
IndexOfDetector [int]
IndexOfLinkage [int]
SystemTemperature [double]
SystemPressure [double]
AutomaticFieldSize [bool]
FieldSizeFactor [VectorD]
ManualFieldSize [VectorD]
AutomaticSampling [bool]
ManualSamplingDefinesSampling [bool]
OversamplingFactor [VectorD]
ManualSamplingDistance [VectorD]
ManualNumberSamplingPoints [int]
ResolveLinearPhase [bool]
ResolveRelativePosition [bool]
InputField [HarmonicFieldsSet]
ParentLightPath [Lightpath]
FieldCounter [double]
```

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.

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;
```

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	- Programming a Degree of Coherence Detector