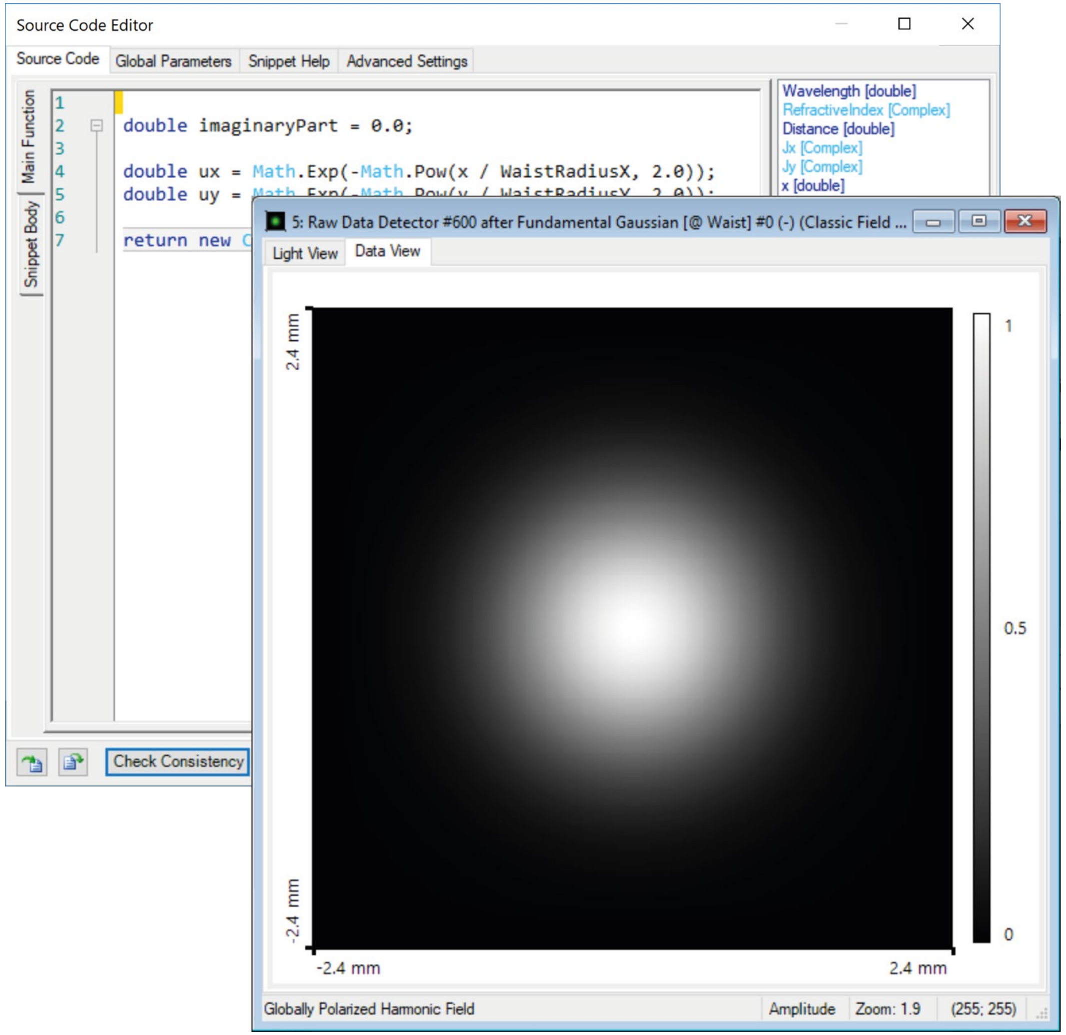
How to Work with the Programmable Light Source in VirtualLab Fusion and Example (Gaussian Beam)

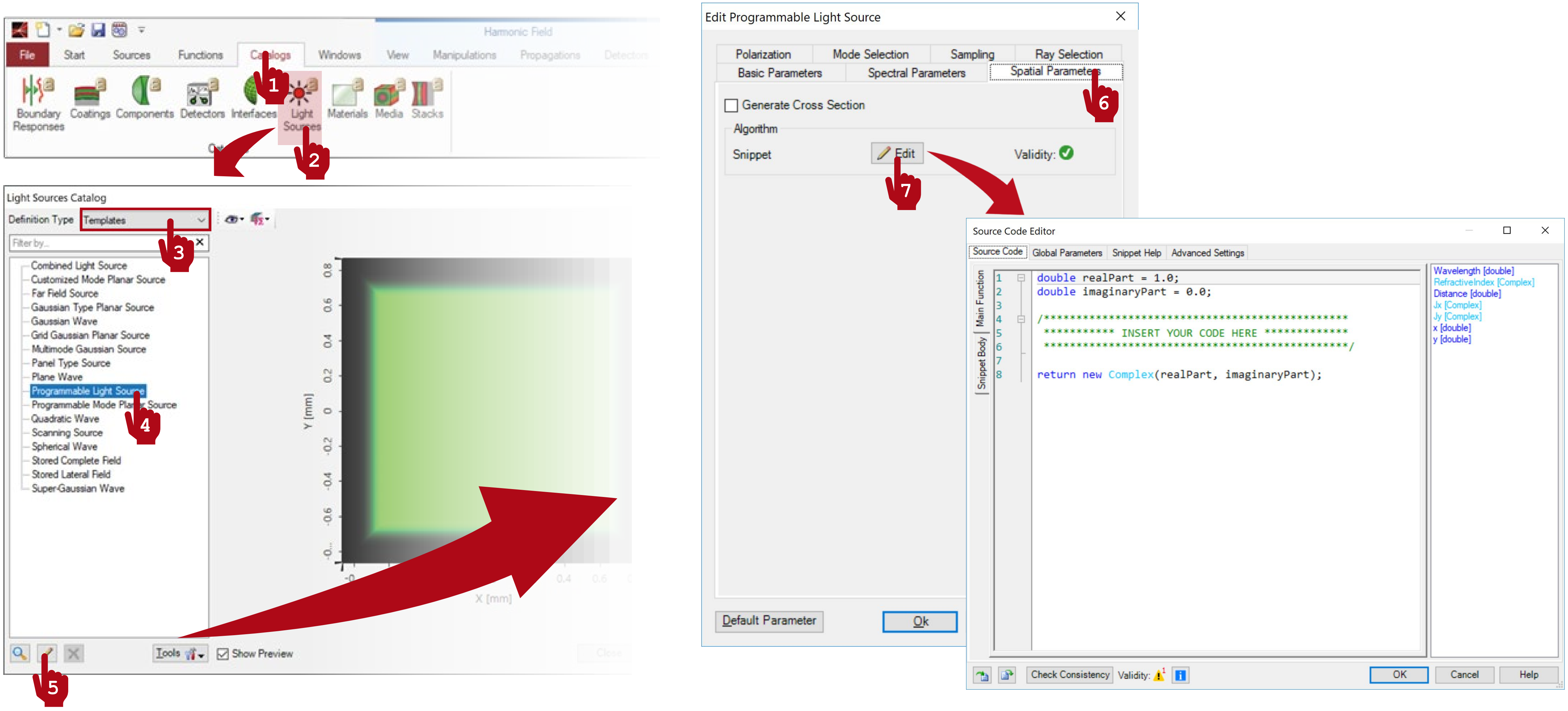
Abstract



```
double imaginaryPart = 0.0;

double ux = Math.Exp(-Math.Pow(x / WaistRadiusX, 2.0));
double uy = Math.Exp(-Math.Pow(y / WaistRadiusY, 2.0));

return new C
```

Providing maximum versatility for your optical simulations is one of our most fundamental objectives. In this document we show you how to work with the Programmable Source: a means to define the spatial dependence of a custom basic source mode which can then be used on its own, as a fully coherent, monochromatic source; or as a single mode in a more complex one (which is perhaps partially spatially coherent or polychromatic). Although the Gaussian beam is one of the source models included in VirtualLab by default, we use it here as a simple programming example.

Where to Find the Programmable Light Source: Catalog



Light Sources Catalog

Definition Type: Templates

Filter by...

- Combined Light Source
- Customized Mode Planar Source
- Far Field Source
- Gaussian Type Planar Source
- Gaussian Wave
- Grid Gaussian Planar Source
- Multimode Gaussian Source
- Panel Type Source
- Plane Wave
- Programmable Light Source
- Programmable Mode Planar Source
- Quadratic Wave
- Scanning Source
- Spherical Wave
- Stored Complete Field
- Stored Lateral Field
- Super-Gaussian Wave

☑ Show Preview

Edit Programmable Light Source

□ Generate Cross Section

Algorithm

Snippet — Edit — Validity: ✓

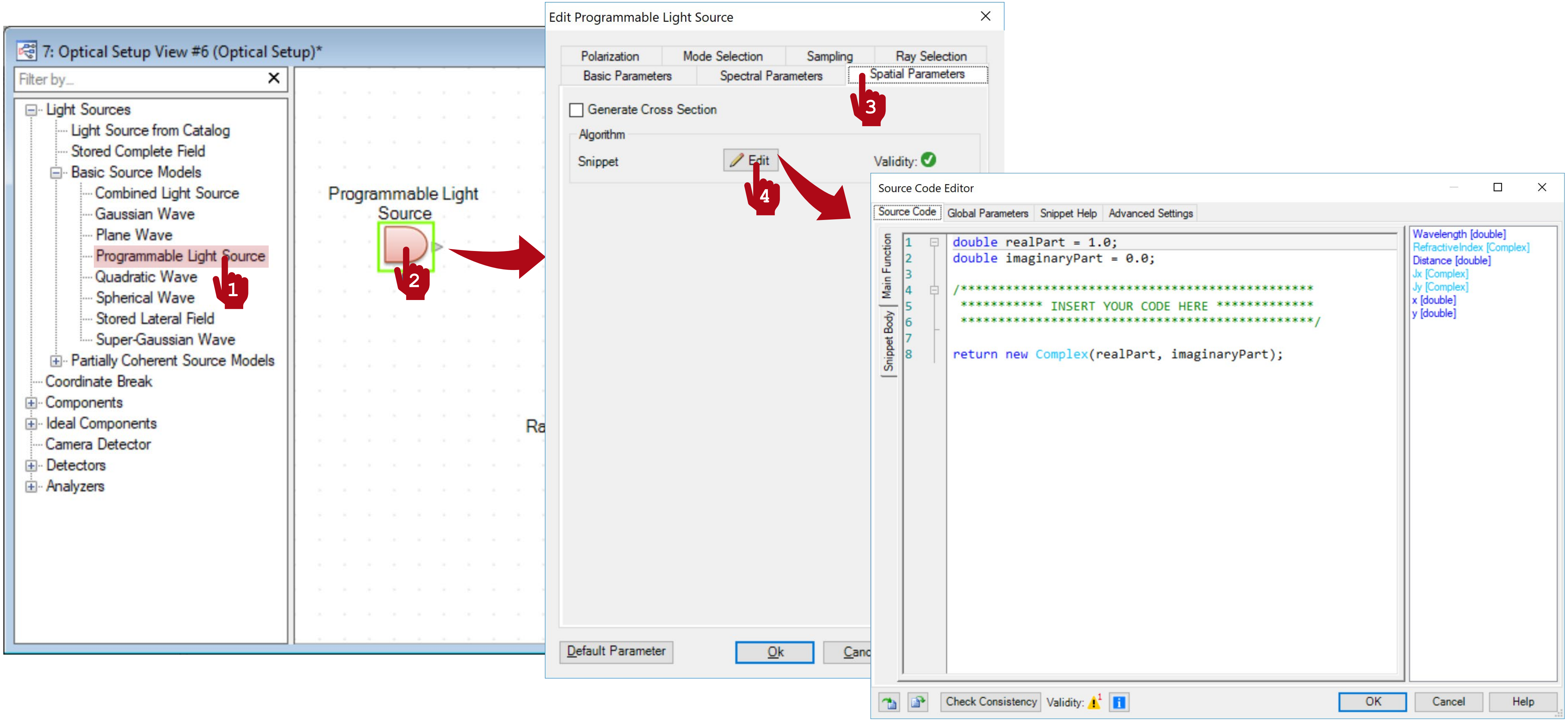
Source Code Editor

```
double realPart = 1.0;
double imaginaryPart = 0.0;

/*********************************************
 *********** INSERT YOUR CODE HERE *************
 *********************************************/

return new Complex(realPart, imaginaryPart);
```

Wavelength [double]
RefractiveIndex [Complex]
Distance [double]
Jx [Complex]
Jy [Complex]
x [double]
y [double]

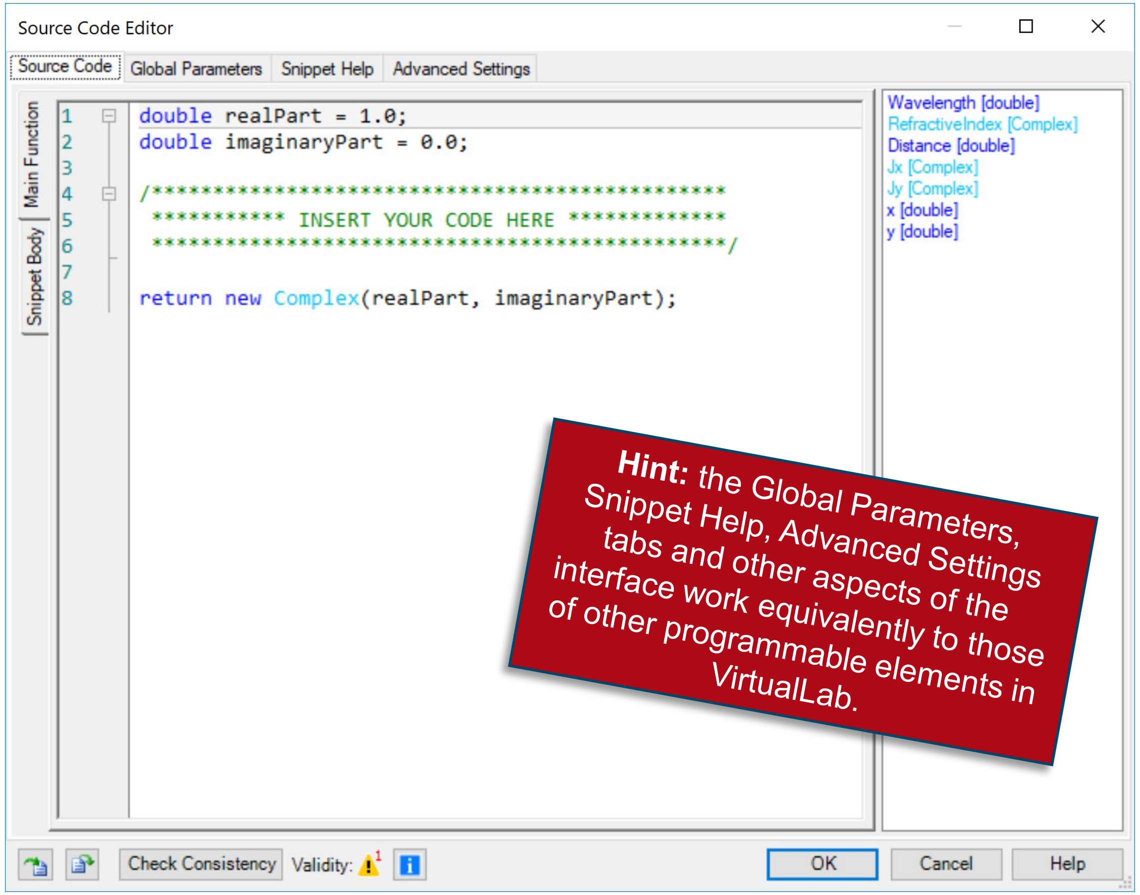
Validity: ⚠

Where to Find the Programmable Light Source: Optical Setup



```
double realPart = 1.0;
double imaginaryPart = 0.0;

/*********************************************
 *********** INSERT YOUR CODE HERE *************
 *********************************************/

return new Complex(realPart, imaginaryPart);
```

Writing the Code



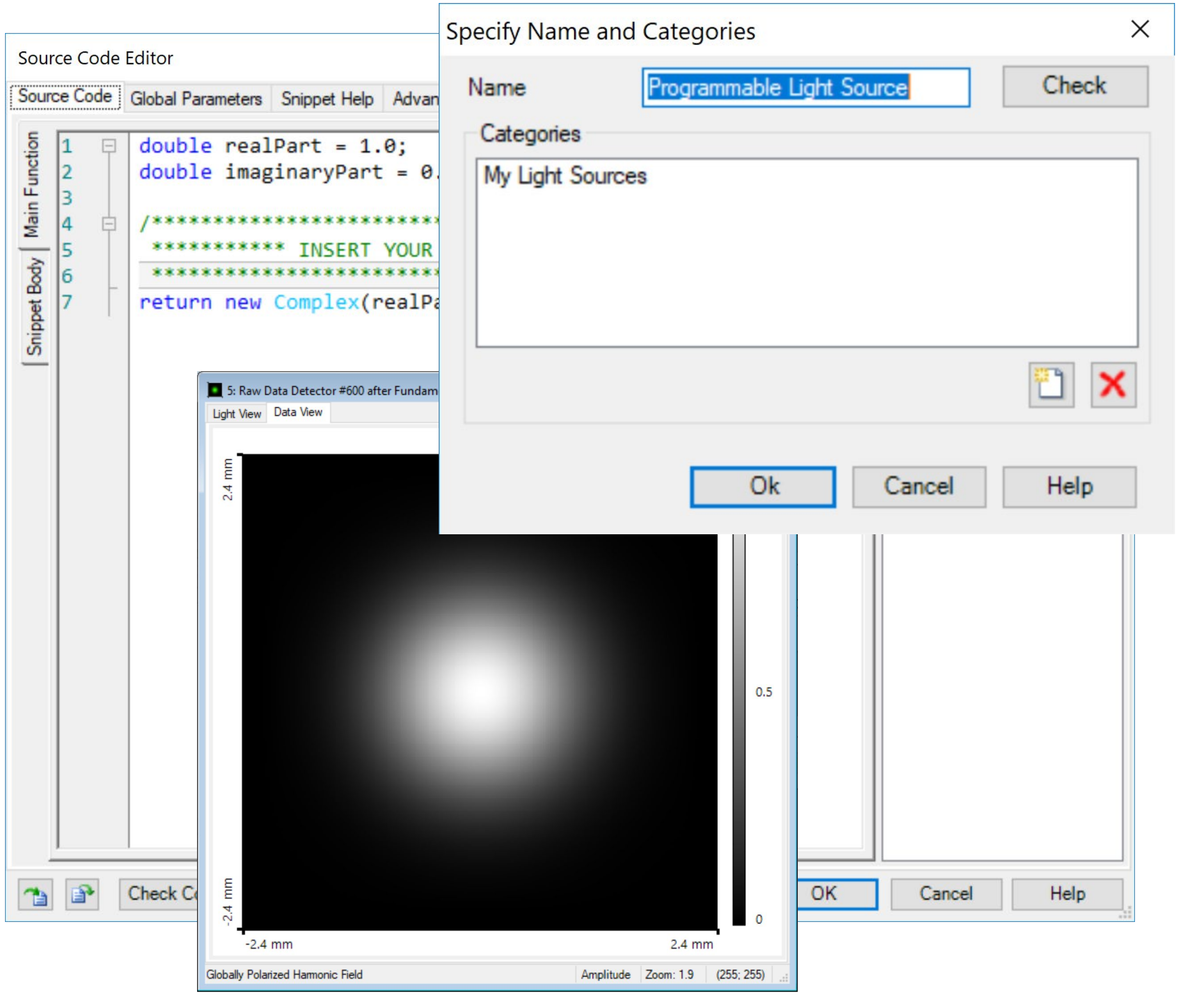
```
double realPart = 1.0;
double imaginaryPart = 0.0;

/**********************************************
 *********** INSERT YOUR CODE HERE *************
 **********************************************/

return new Complex(realPart, imaginaryPart);
```

Hint: the Global Parameters, Snippet Help, Advanced Settings tabs and other aspects of the interface work equivalently to those of other programmable elements in VirtualLab.

- The panel on the right shows a list of available independent parameters.
- `Wavelength` reads in the values of the wavelength or wavelengths specified in the *Spectral Parameters* tab of the configuration dialogue of the source.
- `RefractiveIndex` reads in the complex-valued refractive index of the embedding medium for the specific wavelength of the mode. The embedding medium is defined in the Basic Parameters tab.
- `Distance` reads in another parameter from the configuration dialogue, this time from the Basic Parameters tab: the Distance to Input Plane. This is an important parameter, for instance, in the case of a point source, where the source field cannot be defined exactly at the emitting point.
- `Jx` and `Jy` are the complex-valued components of the Jones polarization vector. If we represent the function which is defined in the code as *U(x, y)*, then the electric components which finally emanate from the source plane are $E_x = J_x\, U(x, y)$ and $E_y = J_y\, U(x, y)$.
- `x` and `y` represent the two-dimensional source plane; they are the coordinates spanning this plane.
- The code in the Main Function must return a `Complex` value per `x, y` point. All these values put together conform the function *U(x, y)*.
- Use the Snippet Body to group parts of the code in support functions.

Output

- The output is a complex-valued function which represents the spatial part of an eventual electromagnetic field component, *U(x, y)*.
- It is a conclusion of Maxwell's equations that, in a homogeneous medium, it is enough to fix two out of the six electromagnetic components, the other four follow from the equations. In VirtualLab E_x and E_y are, without loss of generality, selected to be these independent components. In the Programmable Light Source, they are fixed as $E_x = J_x\, U(x, y)$ and $E_y = J_y\, U(x, y)$.
- The output of the custom source is then an electromagnetic field whose spatial part is defined according to the code, and with a spectral composition as per the Spectral Parameters tab.
- The resulting field can be used as a standalone source in an Optical Setup, it can be saved in the catalog, or it can be employed as a basic mode in a more complex source.

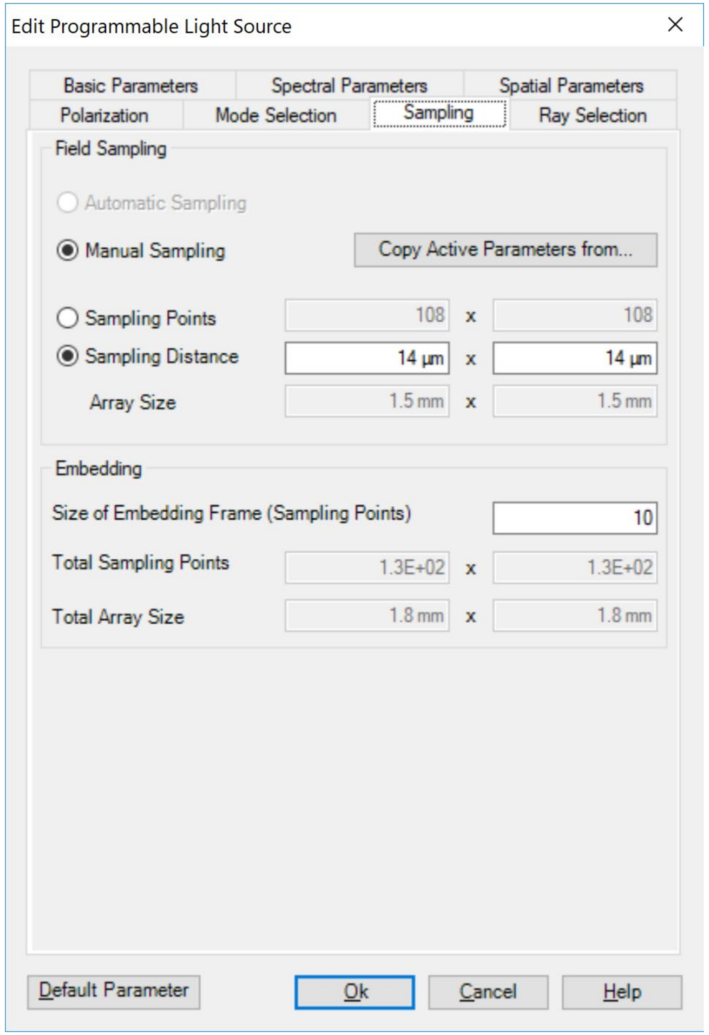


```
double realPart = 1.0;
double imaginaryPart = 0.
/***********************
 *********** INSERT YOUR
 ***********************
return new Complex(realPa
```

Name: Programmable Light Source

Categories: My Light Sources

Sampling



Field Sampling

○ Automatic Sampling

● Manual Sampling

○ Sampling Points 108 x 108

● Sampling Distance 14 µm x 14 µm

Array Size 1.5 mm x 1.5 mm

Embedding

Size of Embedding Frame (Sampling Points) 10

Total Sampling Points 1.3E+02 x 1.3E+02

Total Array Size 1.8 mm x 1.8 mm

- The code defines the source field function analytically, so the accuracy of the programmed function is only limited by double precision.
- The user must ensure that the sampling of the field is fine enough to resolve the function they have implemented.
- Use the Sampling tab for this purpose.
- Please note that the sampling may depend on the actual values of the defined global parameters.

Programming a Gaussian Beam

The Gaussian Beam

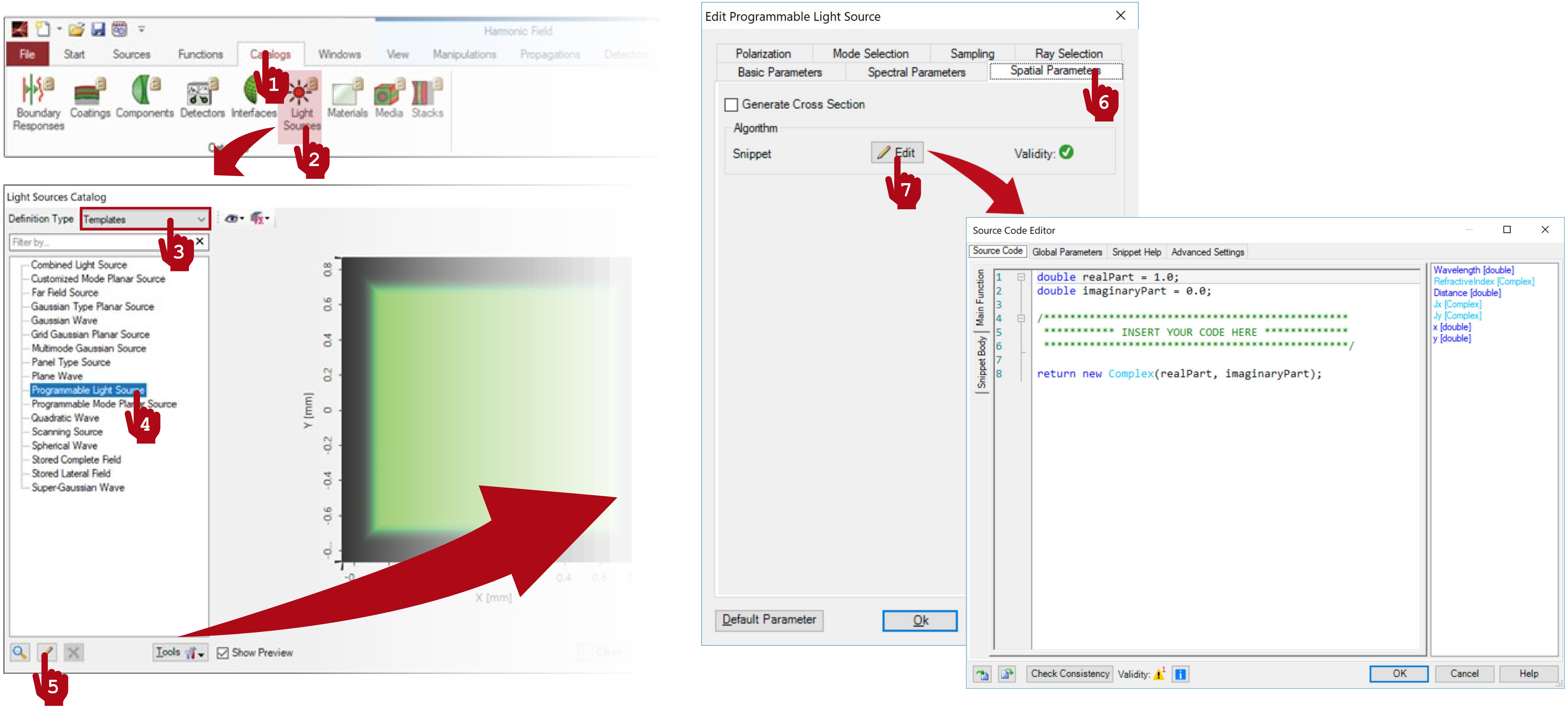
An electromagnetic field is described as a fundamental Gaussian beam when the electric component normal to the main propagation direction is given, at its waist, by a mathematical expression of the form:

$$V_\ell\left(x,y\right) \propto \exp\left(-\frac{x^2}{w_x^2}\right)\exp\left(-\frac{y^2}{w_y^2}\right) \tag{1}$$

$V_\ell\left(x,y\right) \rightarrow$ One of the six electromagnetic components, which takes a Gaussian form

$w_x, w_y \rightarrow$ Waist radius in x and y.

Where to Find the Programmable Light Source: Catalog



Definition Type: Templates

☑ Show Preview

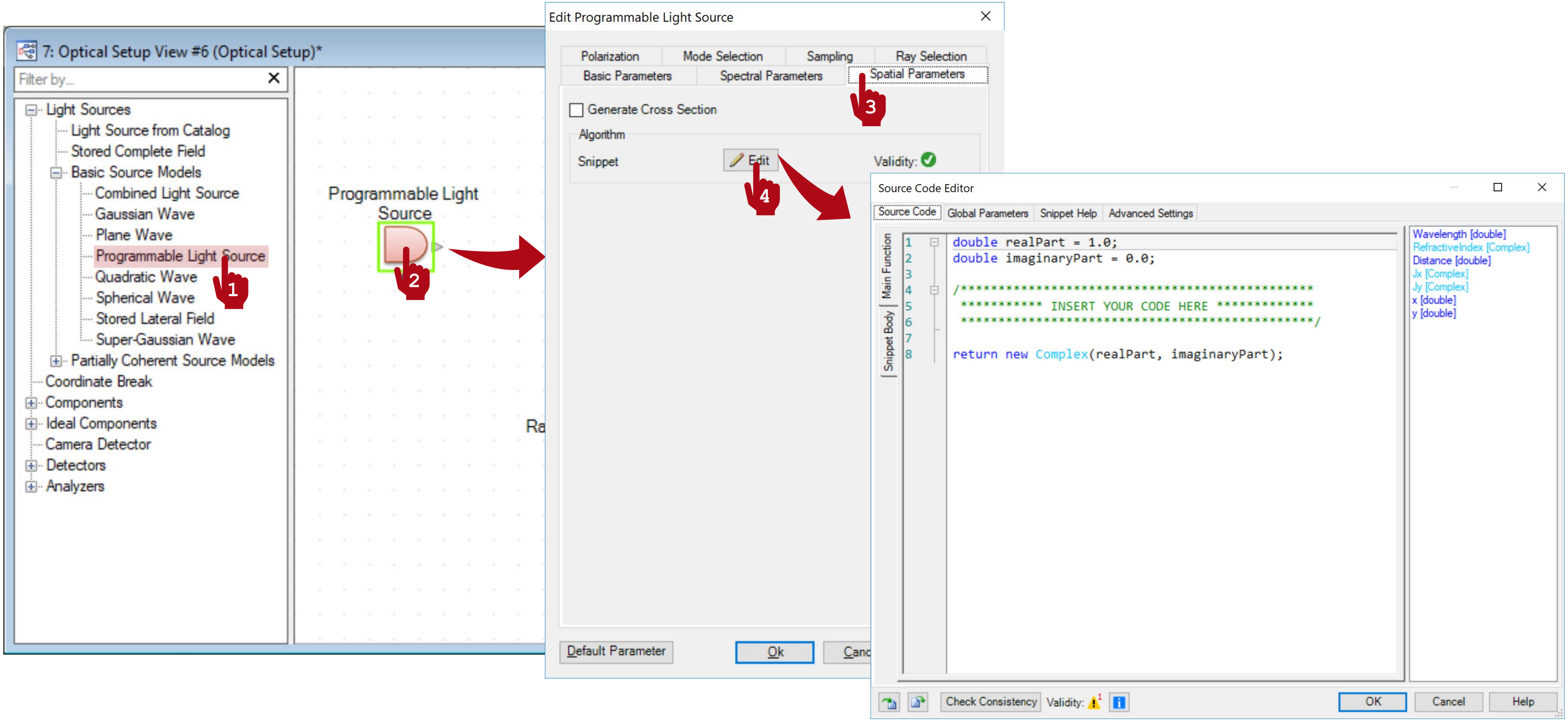
☐ Generate Cross Section

Validity: ✓

```
double realPart = 1.0;
double imaginaryPart = 0.0;

/*******************************************
 *********** INSERT YOUR CODE HERE *************
 *******************************************/

return new Complex(realPart, imaginaryPart);
```

Validity: ⚠

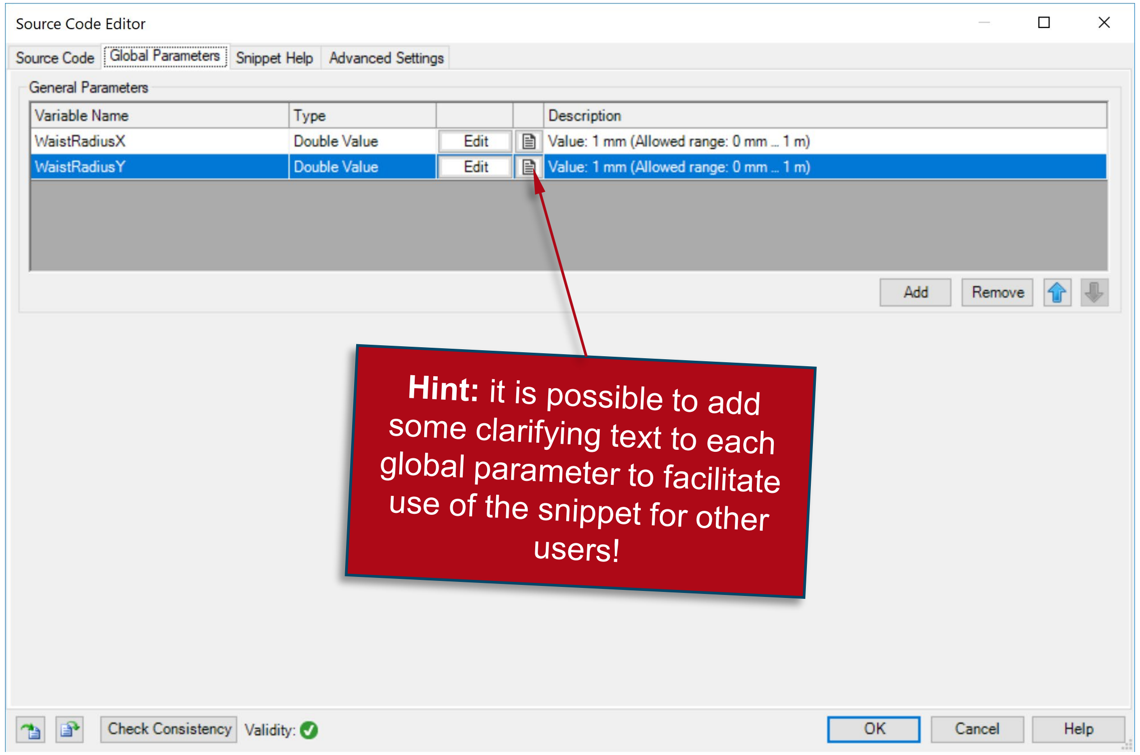
Where to Find the Programmable Light Source: Optical Setup



```
double realPart = 1.0;
double imaginaryPart = 0.0;

/*********************************************
 *********** INSERT YOUR CODE HERE ************
 *********************************************/

return new Complex(realPart, imaginaryPart);
```

Programmable Light Source: Global Parameters

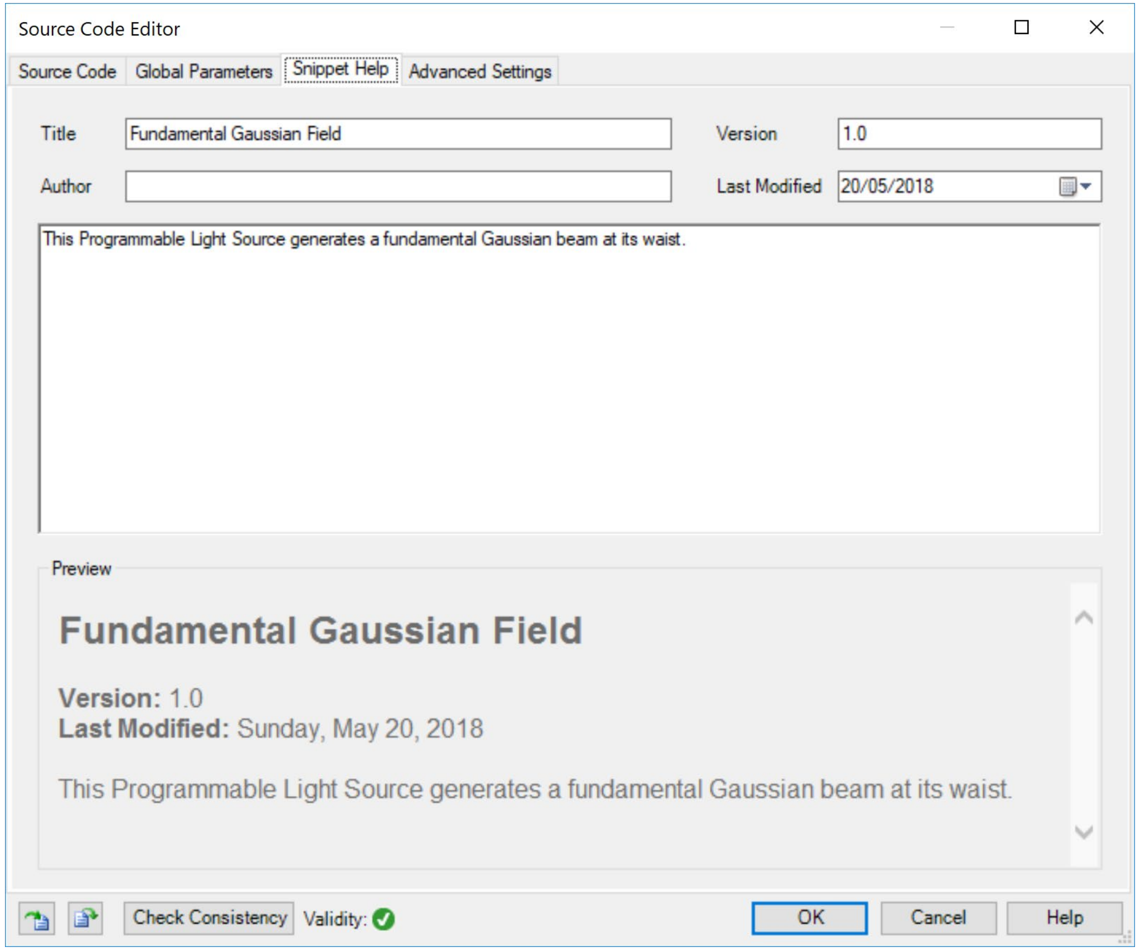
- Once you have triggered open the Edit dialogue, go to the Global Parameters Tab.
- There, Add and Edit two global parameters:
  - `double WaistRadiusX = 1 mm (0 mm, 1 m)`: the radius of the Gaussian beam, in *x* direction, at the waist.
  - `double WaistRadiusY = 1 mm (0 mm, 1 m)`: the radius of the Gaussian beam, in *y* direction, at the waist.



Variable Name	Type			Description
WaistRadiusX	Double Value	Edit	🗎	Value: 1 mm (Allowed range: 0 mm .. 1 m)
WaistRadiusY	Double Value	Edit	🗎	Value: 1 mm (Allowed range: 0 mm .. 1 m)

Hint: it is possible to add some clarifying text to each global parameter to facilitate use of the snippet for other users!

Programmable Light Source: Snippet Help



Title: Fundamental Gaussian Field

Version: 1.0

Author:

Last Modified: 20/05/2018

This Programmable Light Source generates a fundamental Gaussian beam at its waist.

Preview

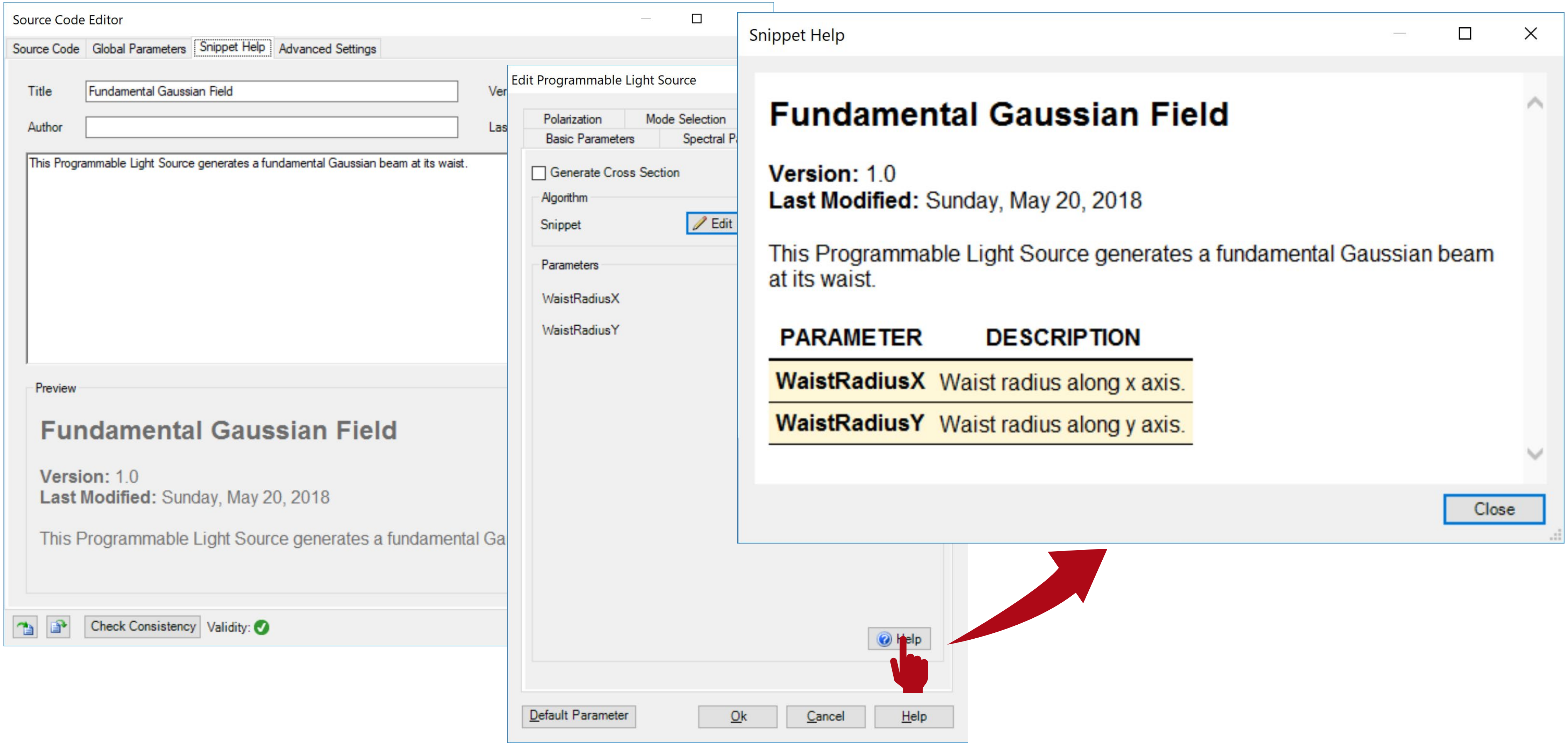
Fundamental Gaussian Field

Version: 1.0
Last Modified: Sunday, May 20, 2018

This Programmable Light Source generates a fundamental Gaussian beam at its waist.

- **Optional:** you can use the Snippet Help to write instructions, clarifications, and some metadata associated to your snippet.
- This option is very helpful to keep track of your progress with a programmable element.
- It is especially useful when the programmable element is later disseminated to be handled by other users!

Programmable Light Source: Snippet Help



Title: Fundamental Gaussian Field

Author:

This Programmable Light Source generates a fundamental Gaussian beam at its waist.

Preview

Fundamental Gaussian Field

Version: 1.0
Last Modified: Sunday, May 20, 2018

This Programmable Light Source generates a fundamental Ga

☐ Generate Cross Section

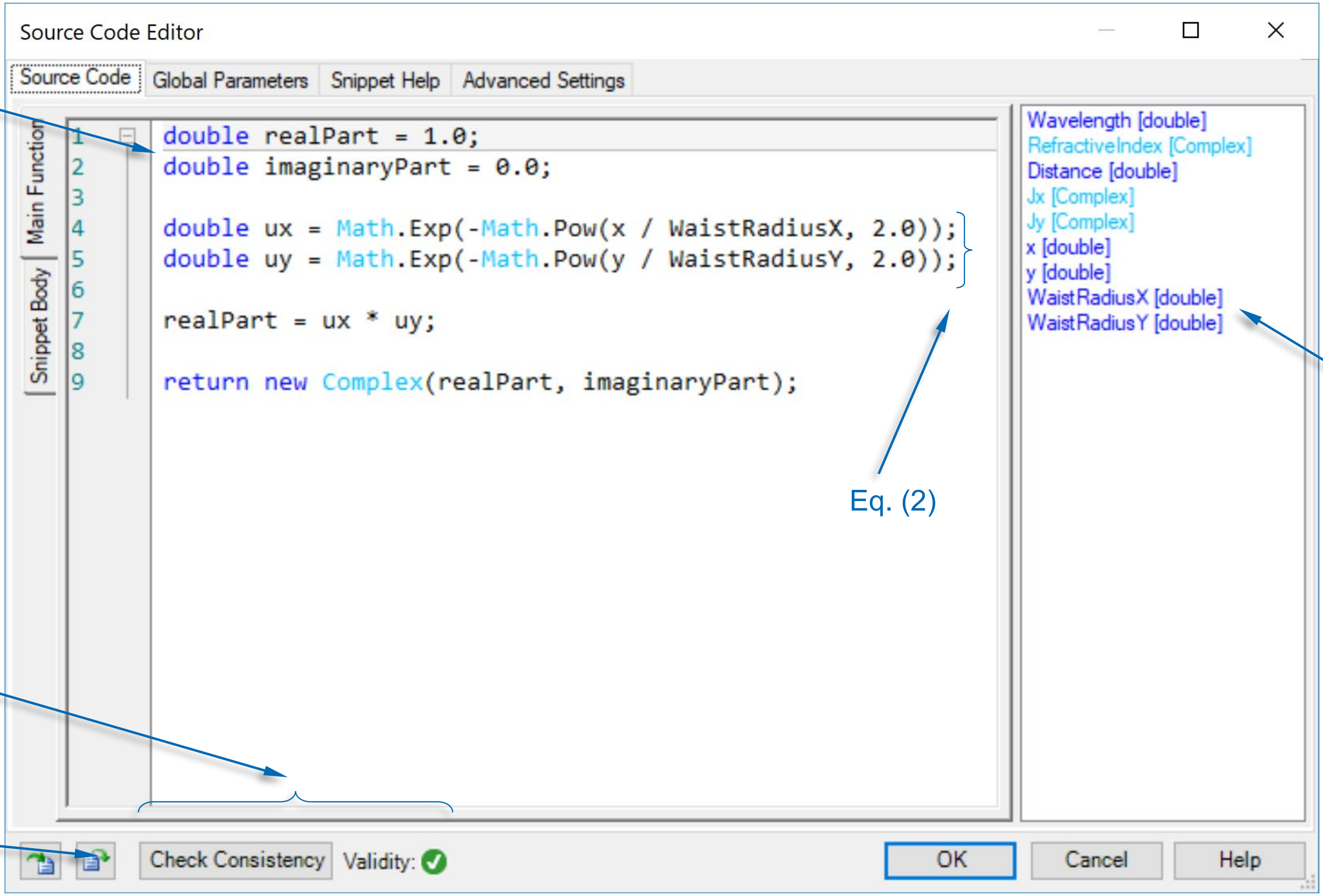
Fundamental Gaussian Field

Version: 1.0
Last Modified: Sunday, May 20, 2018

This Programmable Light Source generates a fundamental Gaussian beam at its waist.

PARAMETER	DESCRIPTION
WaistRadiusX	Waist radius along x axis.
WaistRadiusY	Waist radius along y axis.

Programmable Light Source: Writing the Code



```
double realPart = 1.0;
double imaginaryPart = 0.0;

double ux = Math.Exp(-Math.Pow(x / WaistRadiusX, 2.0));
double uy = Math.Exp(-Math.Pow(y / WaistRadiusY, 2.0));

realPart = ux * uy;

return new Complex(realPart, imaginaryPart);
```

Declaration of output variable given by default

Default global parameters/variables

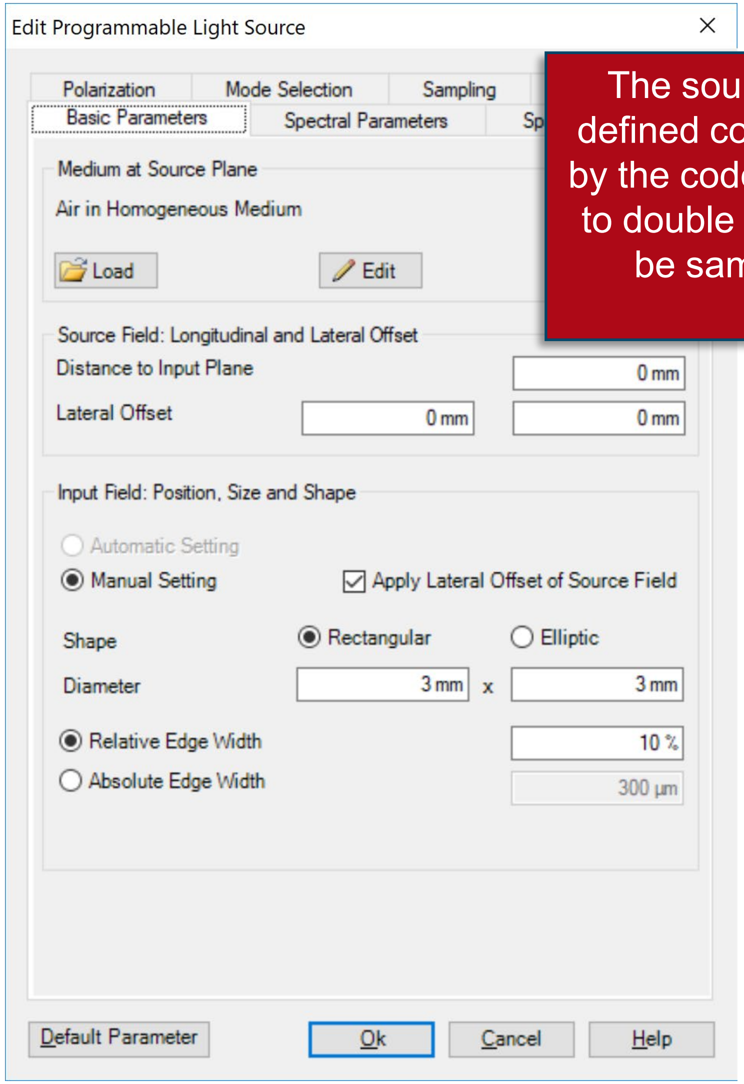
Global parameters defined by user in Global Parameters tab

Eq. (2)

Are there errors in your code?

Export Snippet to save your work!

Programmable Light Source: Adjusting Sampling and Window



Edit Programmable Light Source

Polarization | Mode Selection | Sampling | Sp

Basic Parameters | Spectral Parameters | Sp

Medium at Source Plane

Air in Homogeneous Medium

Load | Edit

Source Field: Longitudinal and Lateral Offset

Distance to Input Plane: 0 mm

Lateral Offset: 0 mm, 0 mm

Input Field: Position, Size and Shape

○ Automatic Setting

● Manual Setting ☑ Apply Lateral Offset of Source Field

Shape: ● Rectangular ○ Elliptic

Diameter: 3 mm x 3 mm

● Relative Edge Width: 10 %

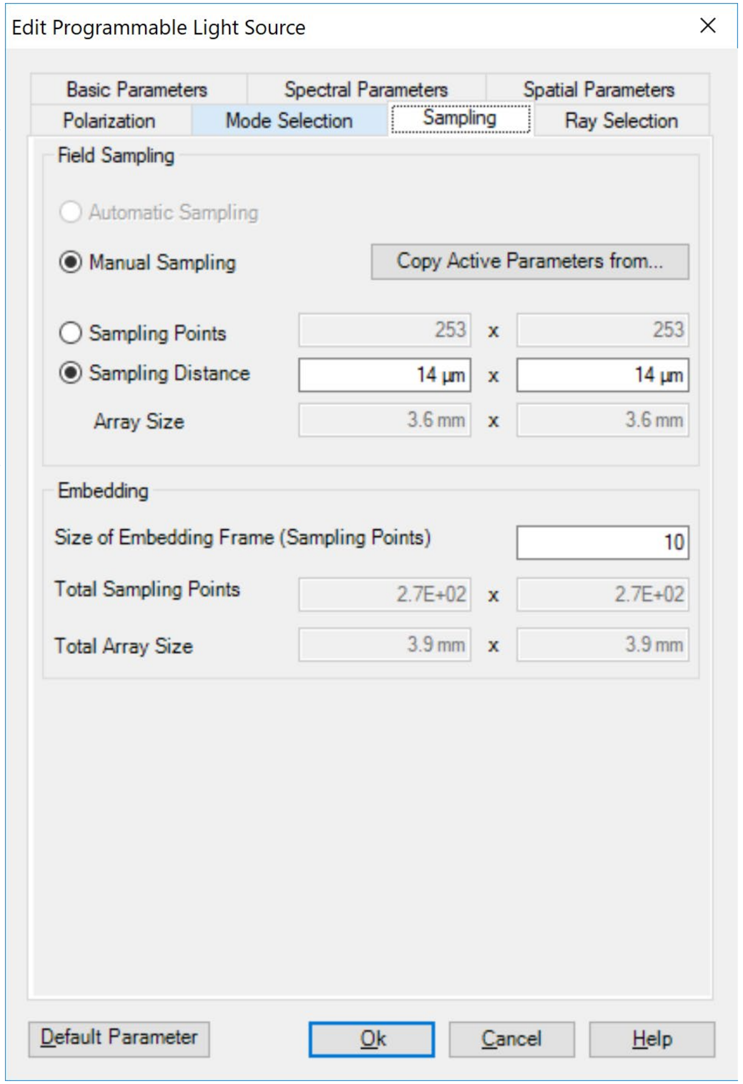
○ Absolute Edge Width: 300 µm

Default Parameter | Ok | Cancel | Help

The source field function is defined completely analytically by the code—**full accuracy** (up to double precision)—and can be sampled as finely as required!

It is up to the user to define a suitable area of definition in the *x*, *y* plane for their custom source in the Basic Parameters tab

It is up to the user to define a suitable sampling distance for their custom source in the Sampling tab



Edit Programmable Light Source

Basic Parameters | Spectral Parameters | Spatial Parameters

Polarization | Mode Selection | Sampling | Ray Selection

Field Sampling

○ Automatic Sampling

● Manual Sampling — Copy Active Parameters from...

○ Sampling Points: 253 x 253

● Sampling Distance: 14 µm x 14 µm

Array Size: 3.6 mm x 3.6 mm

Embedding

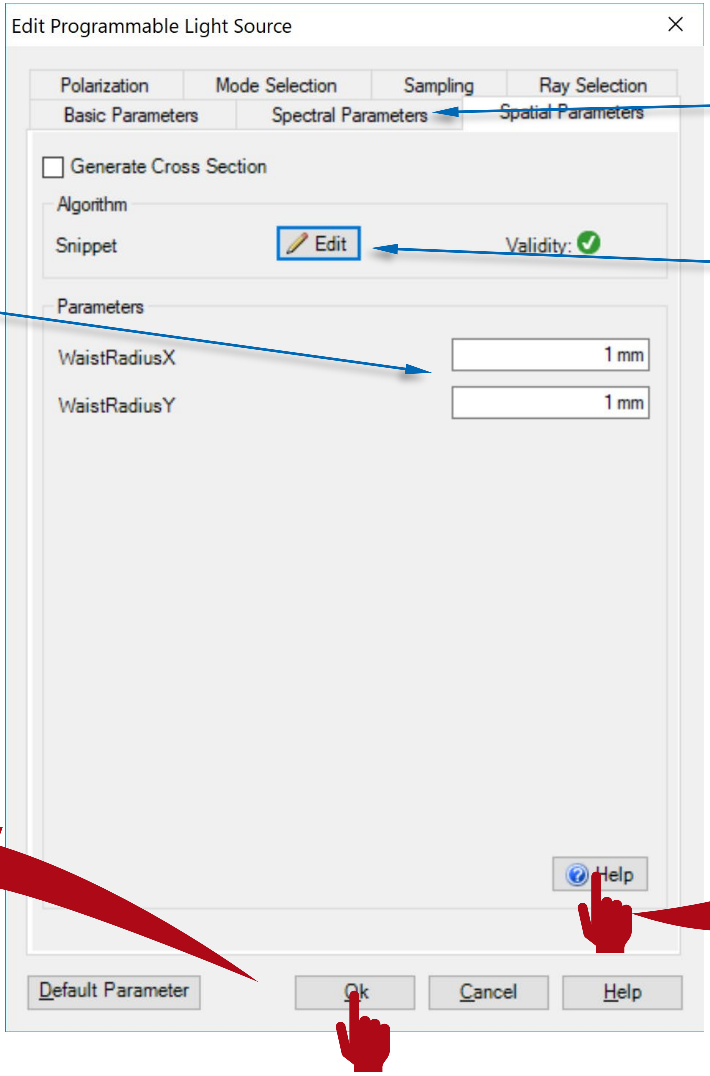
Size of Embedding Frame (Sampling Points): 10

Total Sampling Points: 2.7E+02 x 2.7E+02

Total Array Size: 3.9 mm x 3.9 mm

Default Parameter | Ok | Cancel | Help

Programmable Light Source: Using Your Snippet



Edit Programmable Light Source

☐ Generate Cross Section

Algorithm

Snippet — Edit — Validity: ✓

Parameters

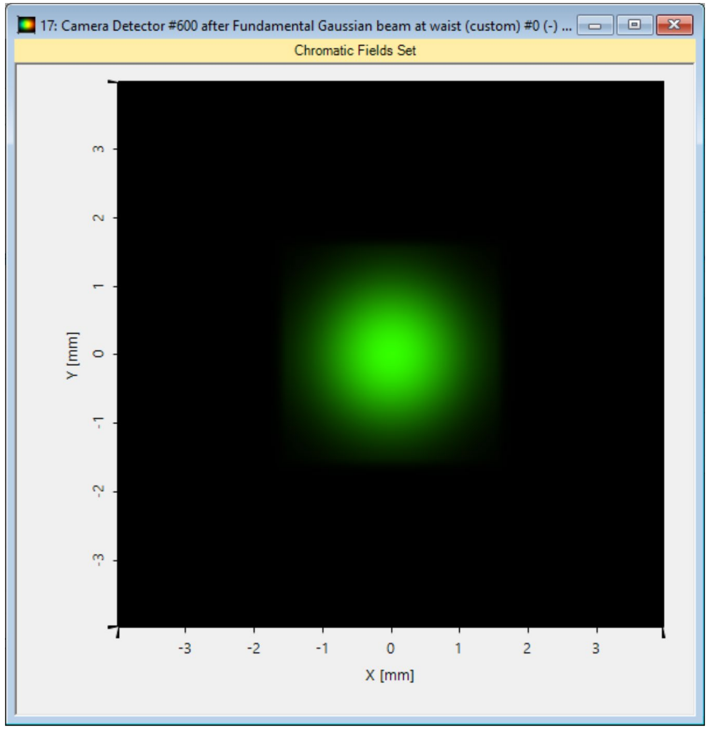
WaistRadiusX: 1 mm

WaistRadiusY: 1 mm

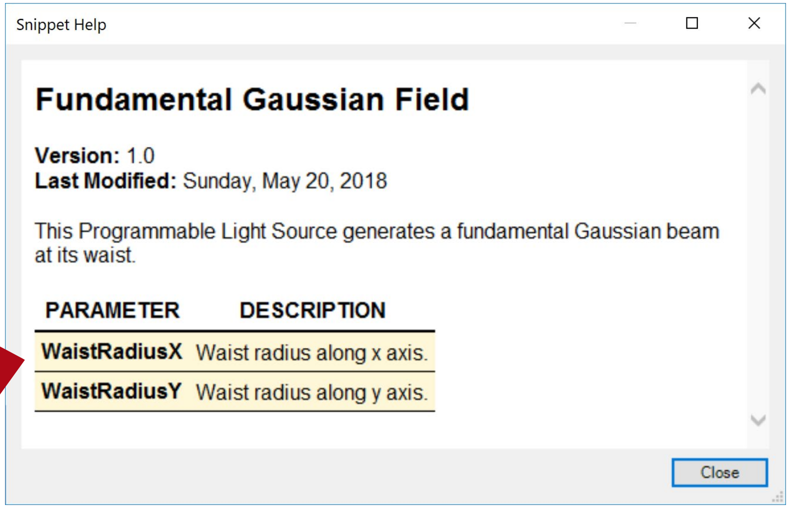
You can modify the value of the global parameters you defined here

Configure the spectral make-up of the source independently in the Spectral Parameters tab

Modify your snippet by clicking on Edit



17: Camera Detector #600 after Fundamental Gaussian beam at waist (custom) #0 (-) ...



Snippet Help

Fundamental Gaussian Field

Version: 1.0
Last Modified: Sunday, May 20, 2018

This Programmable Light Source generates a fundamental Gaussian beam at its waist.

PARAMETER	DESCRIPTION
WaistRadiusX	Waist radius along x axis.
WaistRadiusY	Waist radius along y axis.

Test the Code!

Main Function (Height Profile)

```
double realPart = 1.0;
double imaginaryPart = 0.0;

double ux = Math.Exp(-Math.Pow(x / WaistRadiusX, 2.0));
double uy = Math.Exp(-Math.Pow(y / WaistRadiusY, 2.0));

realPart = ux * uy;

return new Complex(realPart, imaginaryPart);
```

Document Information

title	How to Work with the Programmable Light Source in VirtualLab Fusion and Example (Gaussian Beam)
document code	CZT.0097
version	1.0
toolbox(es)	Starter Toolbox
VL version used for simulations	7.4.0.49
category	Feature Use Case
further reading	- Customizable Help for Programmable Elements - Programming Radially & Azimuthally Polarized Sources