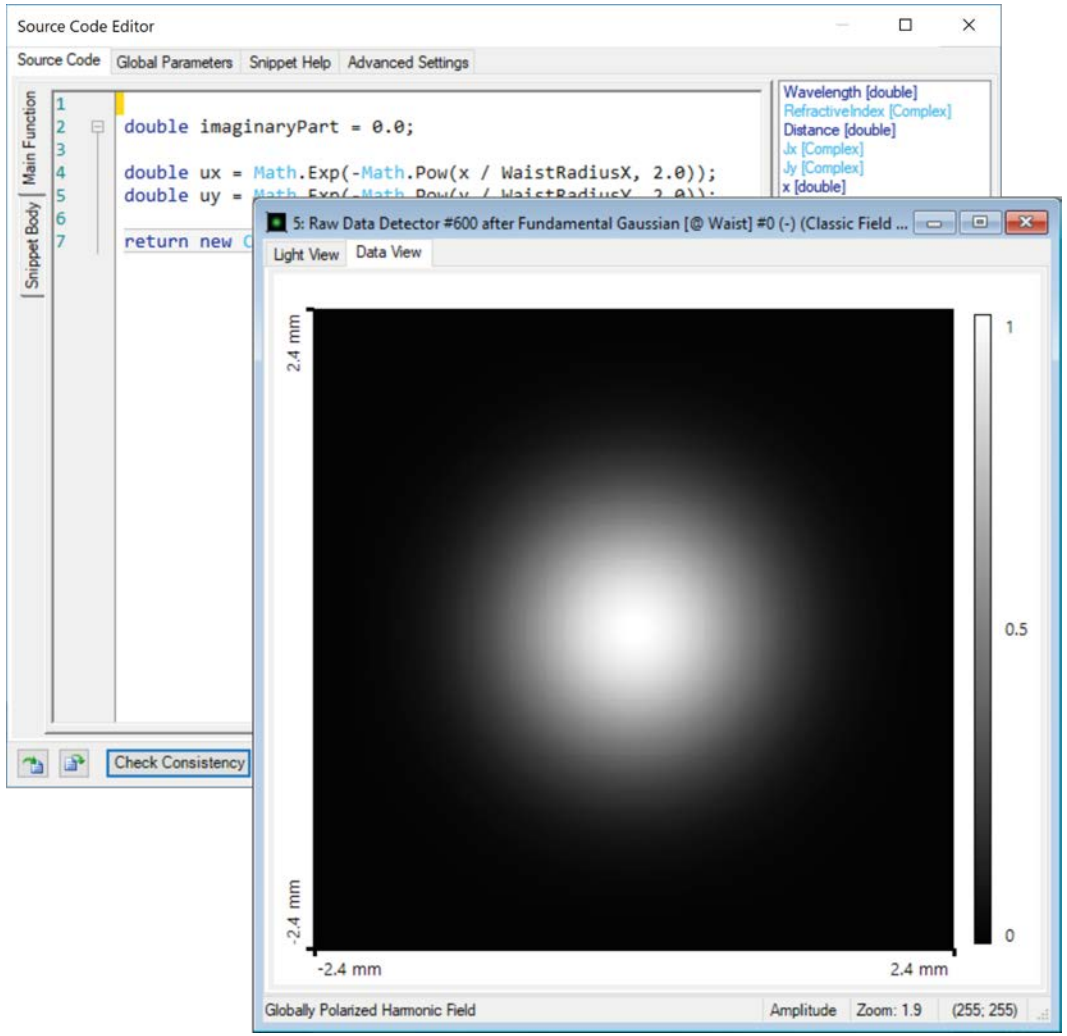
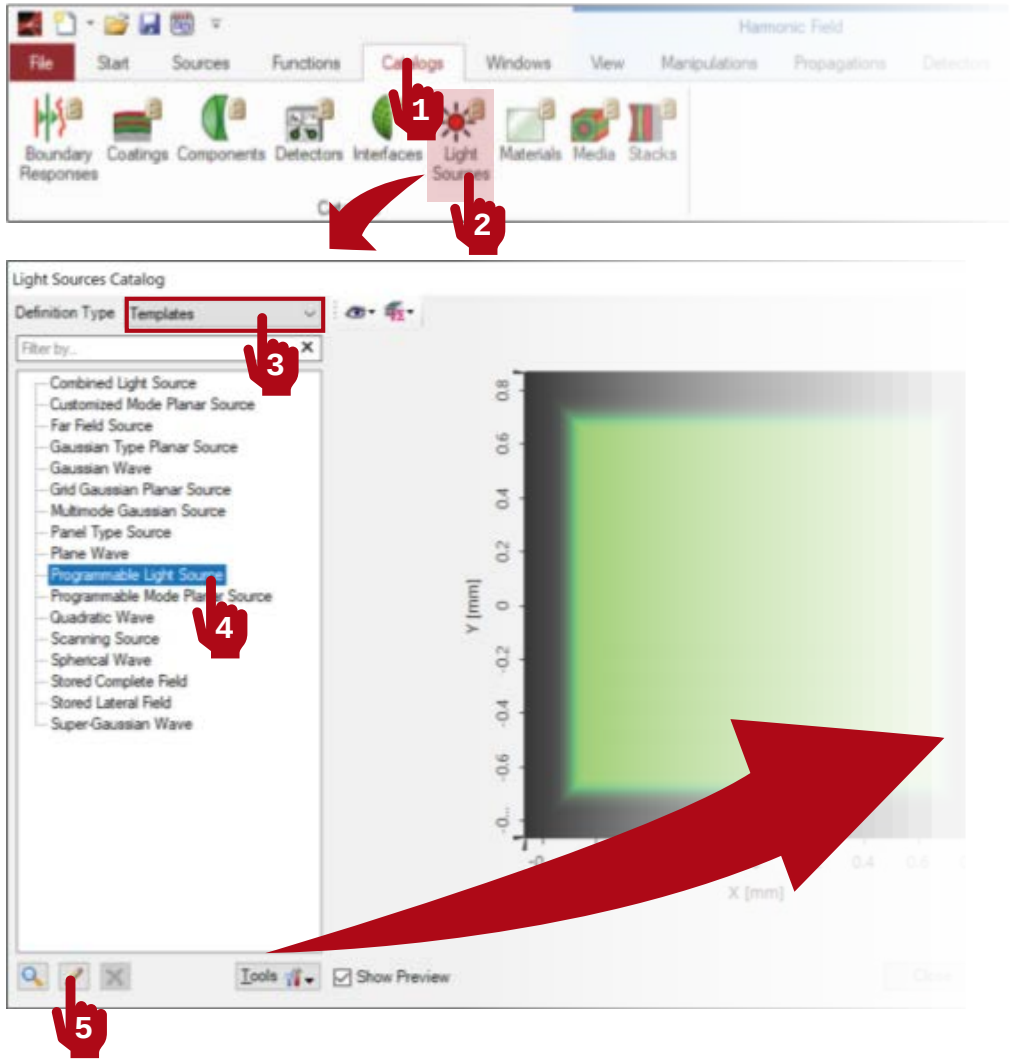
How to Work with the Programmable Light Source in VirtualLab Fusion and Example (Gaussian Beam)

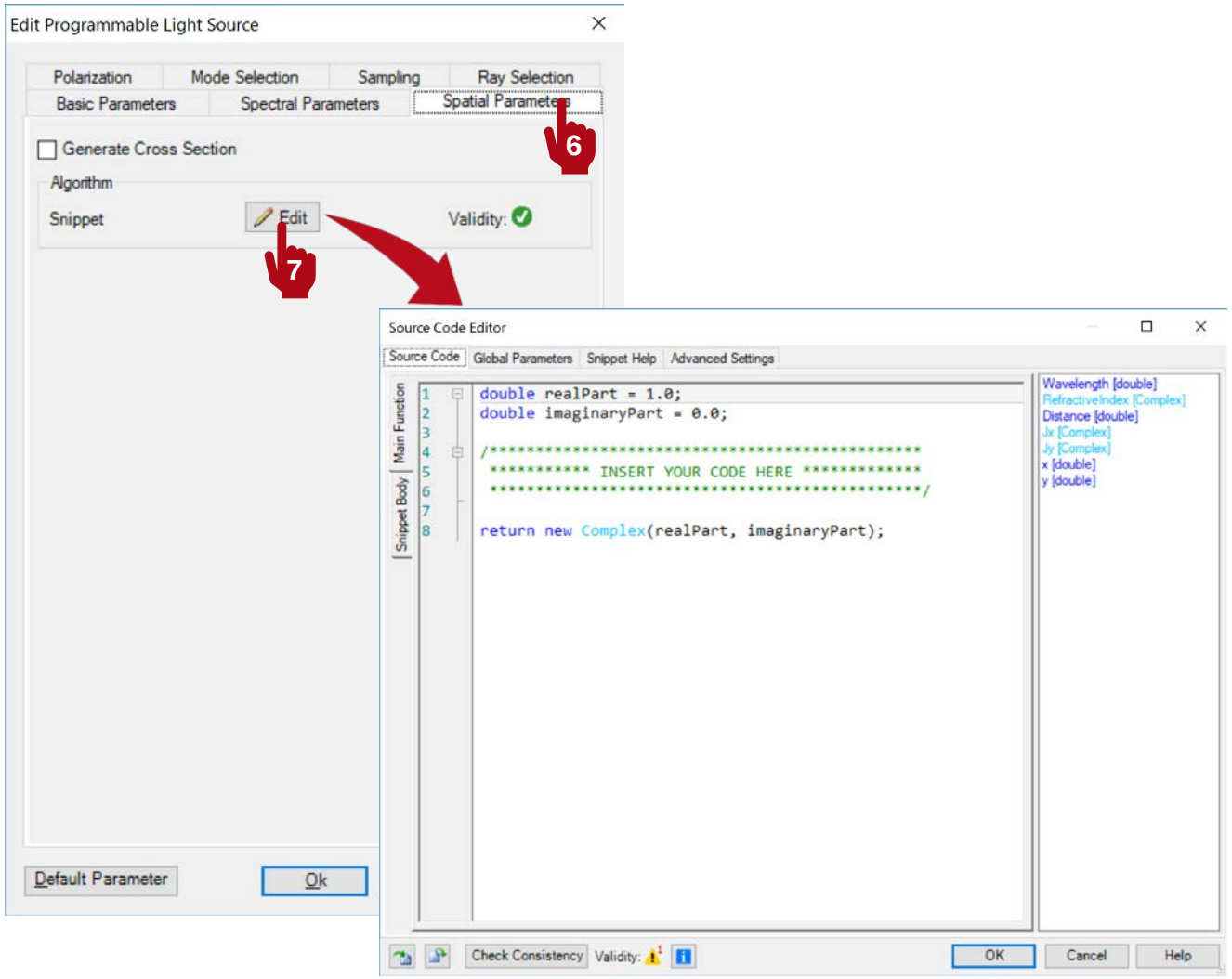
Abstract



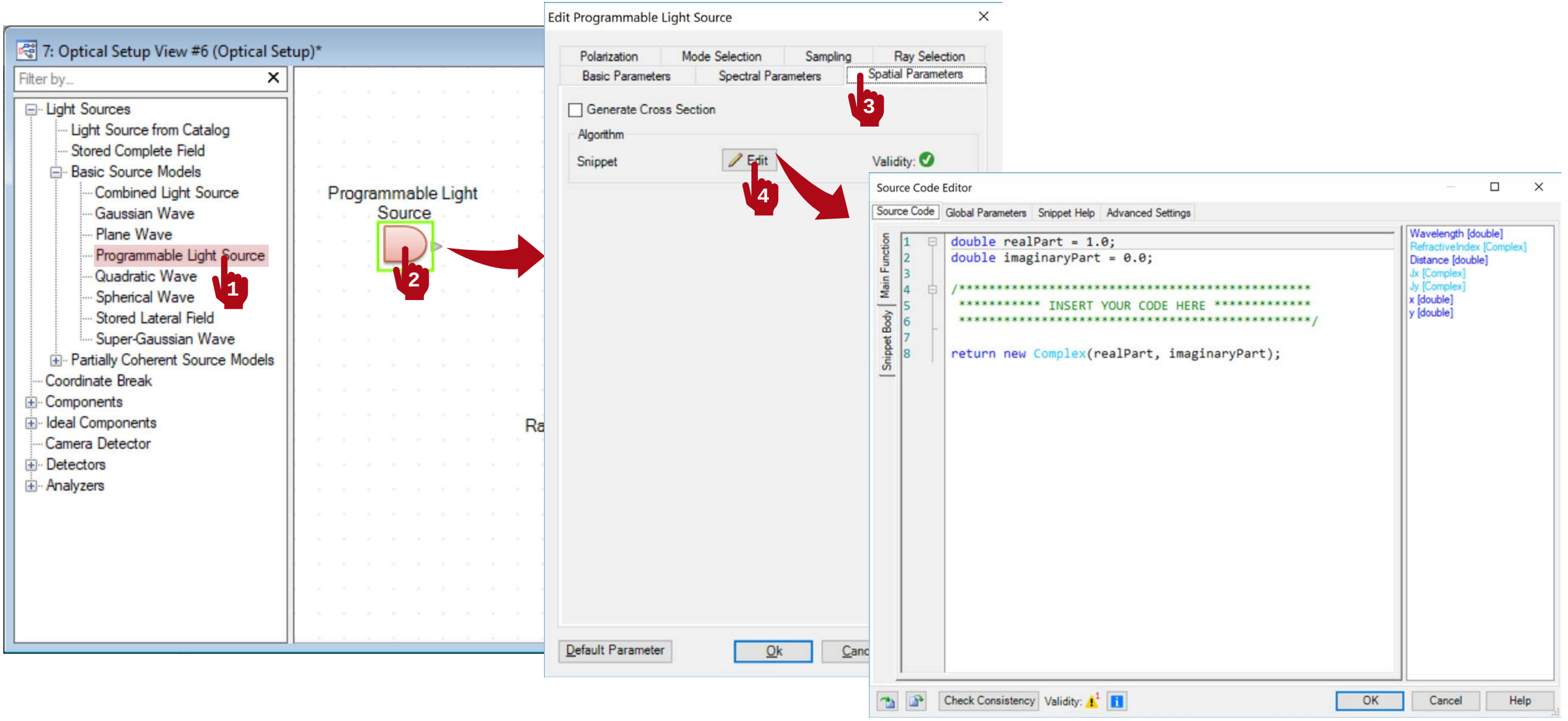
Providing maximum versatility for your optical simulations is one of our most fundamental objectives. In this document we show you how to work with the Programmable Source: a means to define the spatial dependence of a custom basic source mode which can then be used on its own, as a fully coherent, monochromatic 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

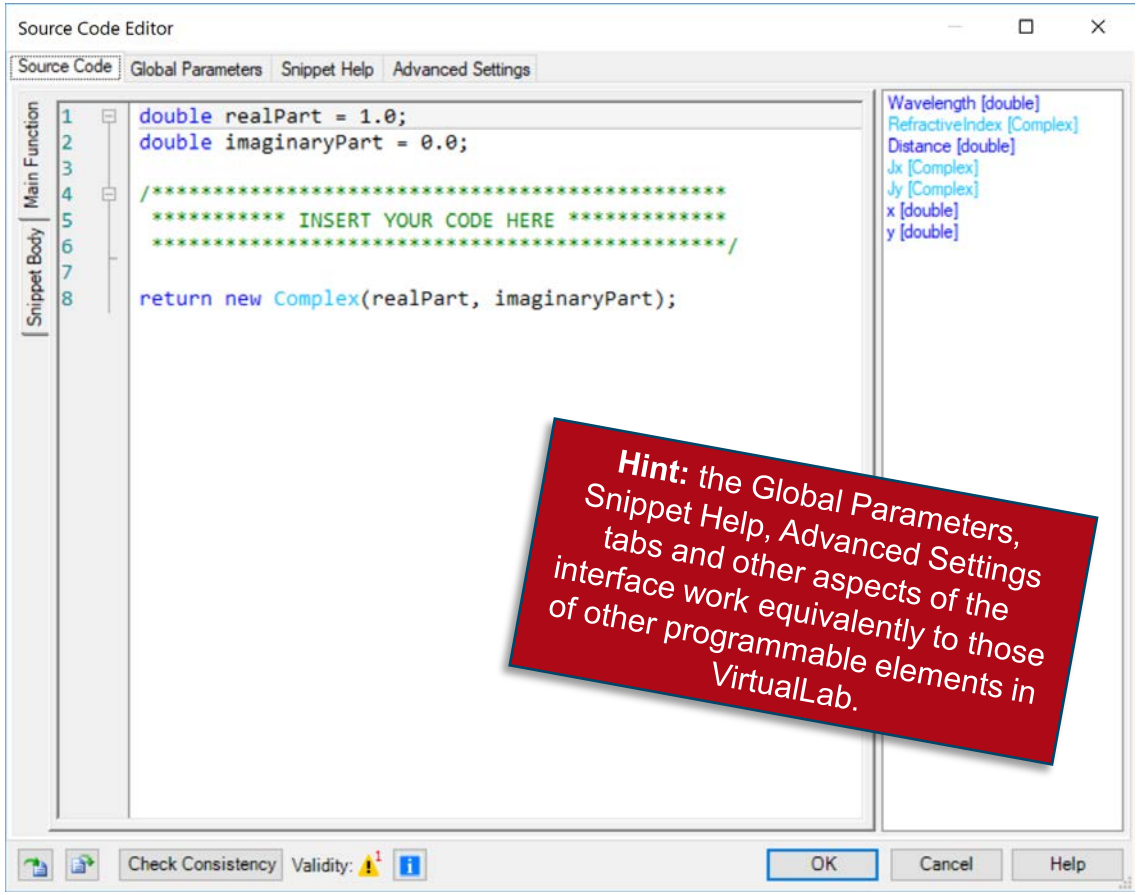




Where to Find the Programmable Light Source: Optical Setup



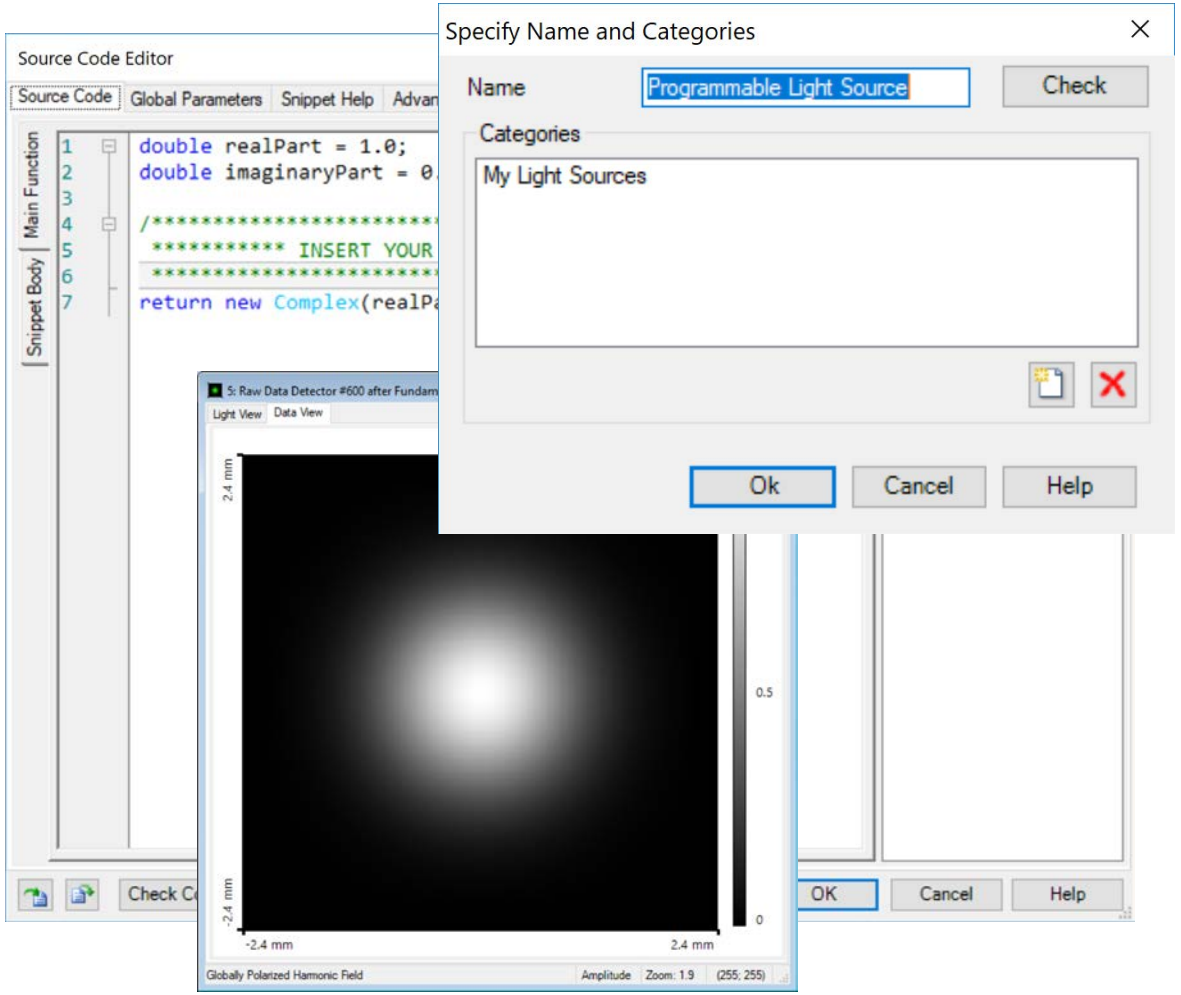
Writing the Code



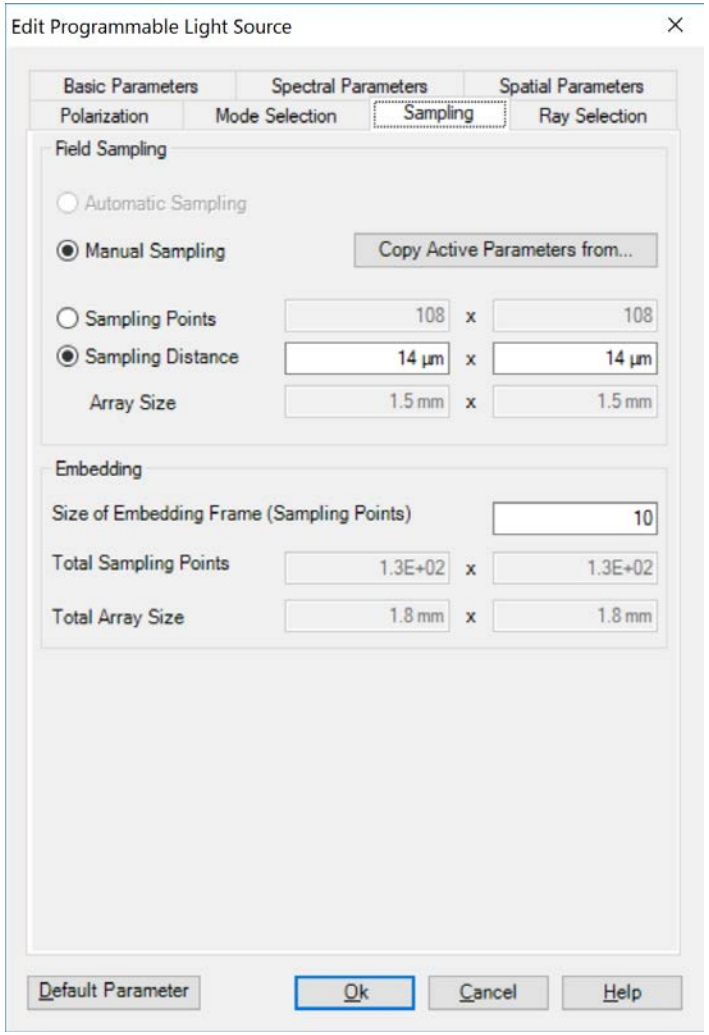
- The panel on the right shows a list of available independent parameters.
- `Wavelength` reads in the values of the wavelength or wavelengths specified in the *Spectral Parameters* tab of the configuration dialogue of the source.
- `RefractiveIndex` reads in the complex-valued refractive index of the 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.



Sampling



- 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

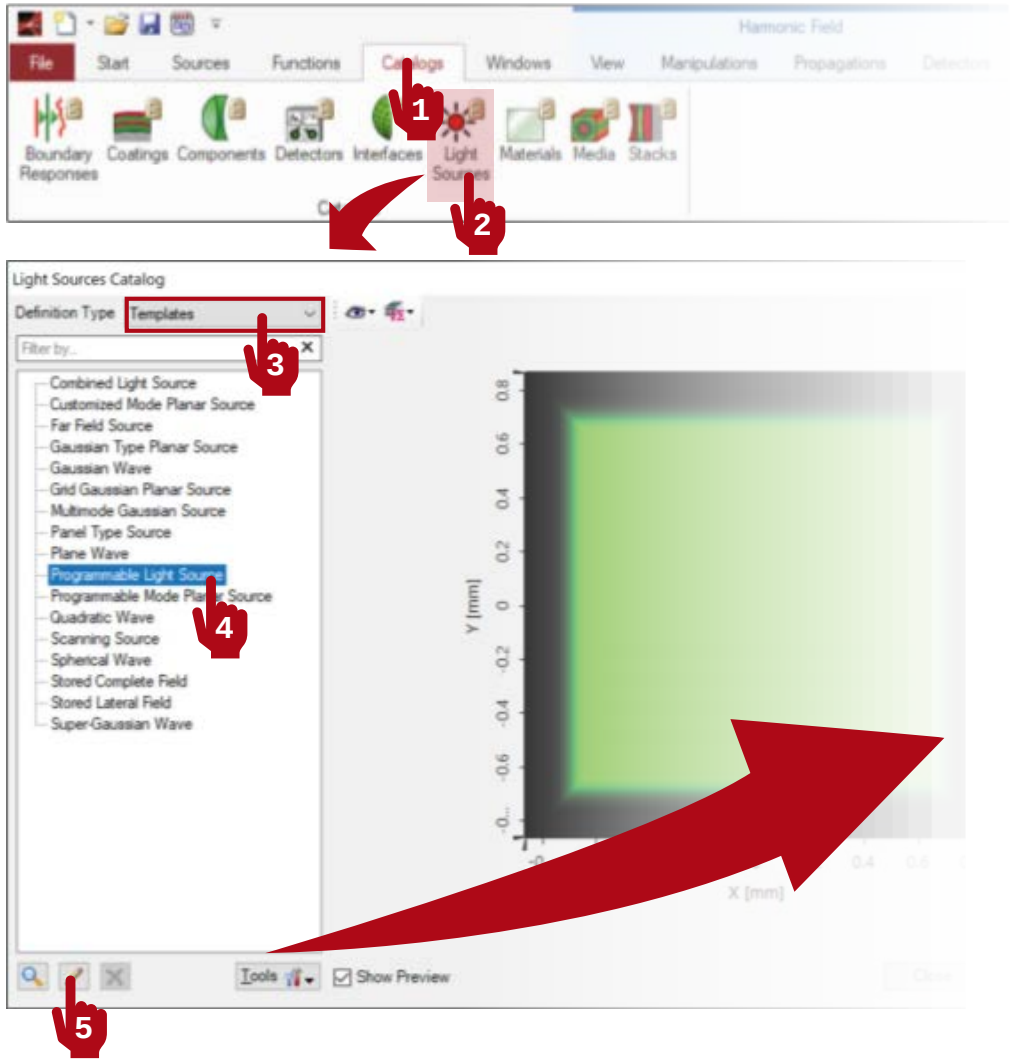
An electromagnetic field is described as a fundamental Gaussian beam when the electric component normal to the main propagation direction is given, at its waist, by a mathematical expression of the form:

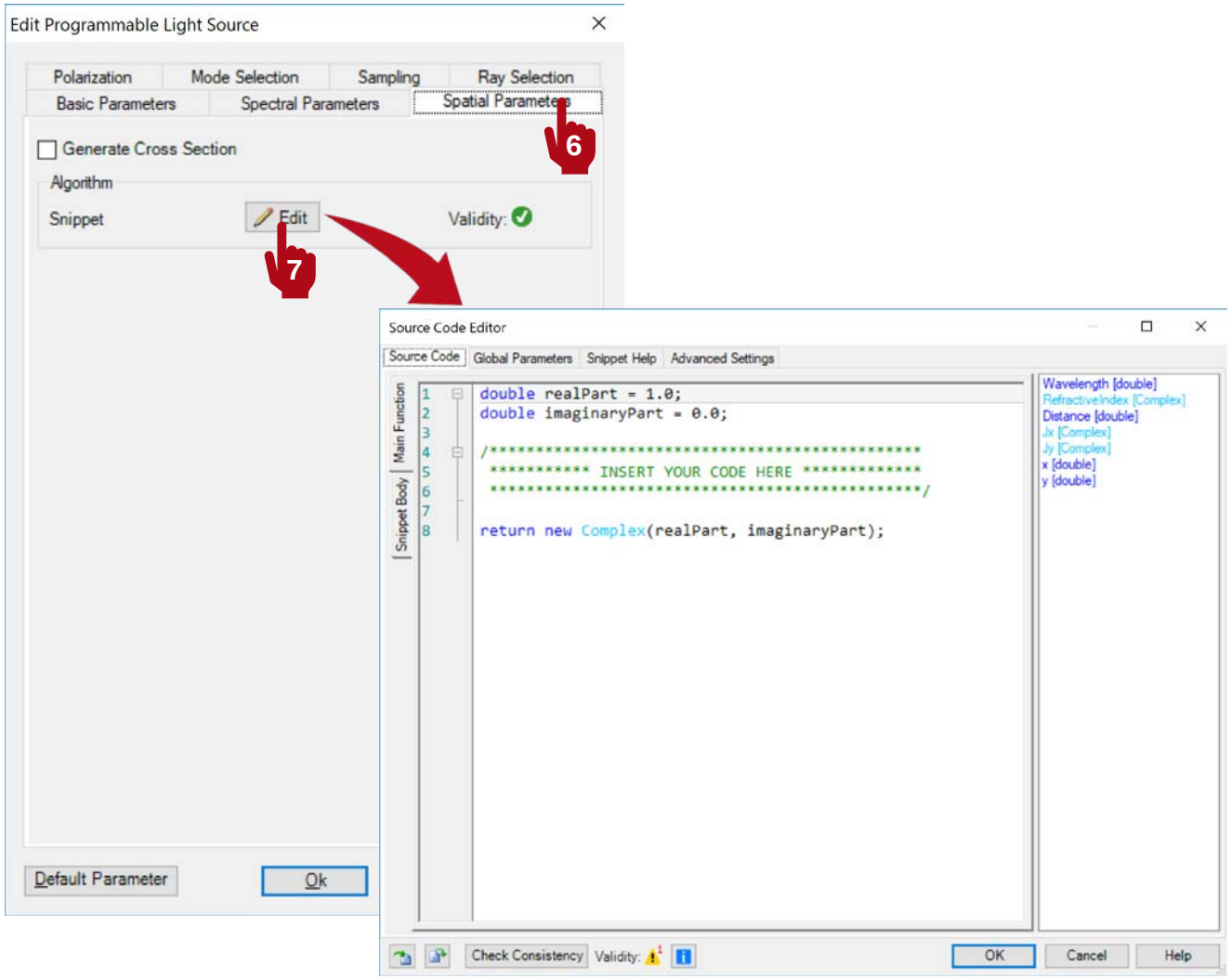
$$V_\ell(x,y) \propto \exp\left(-\frac{x^2}{w_x^2}\right) \exp\left(-\frac{y^2}{w_y^2}\right) \qquad (1)$$

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

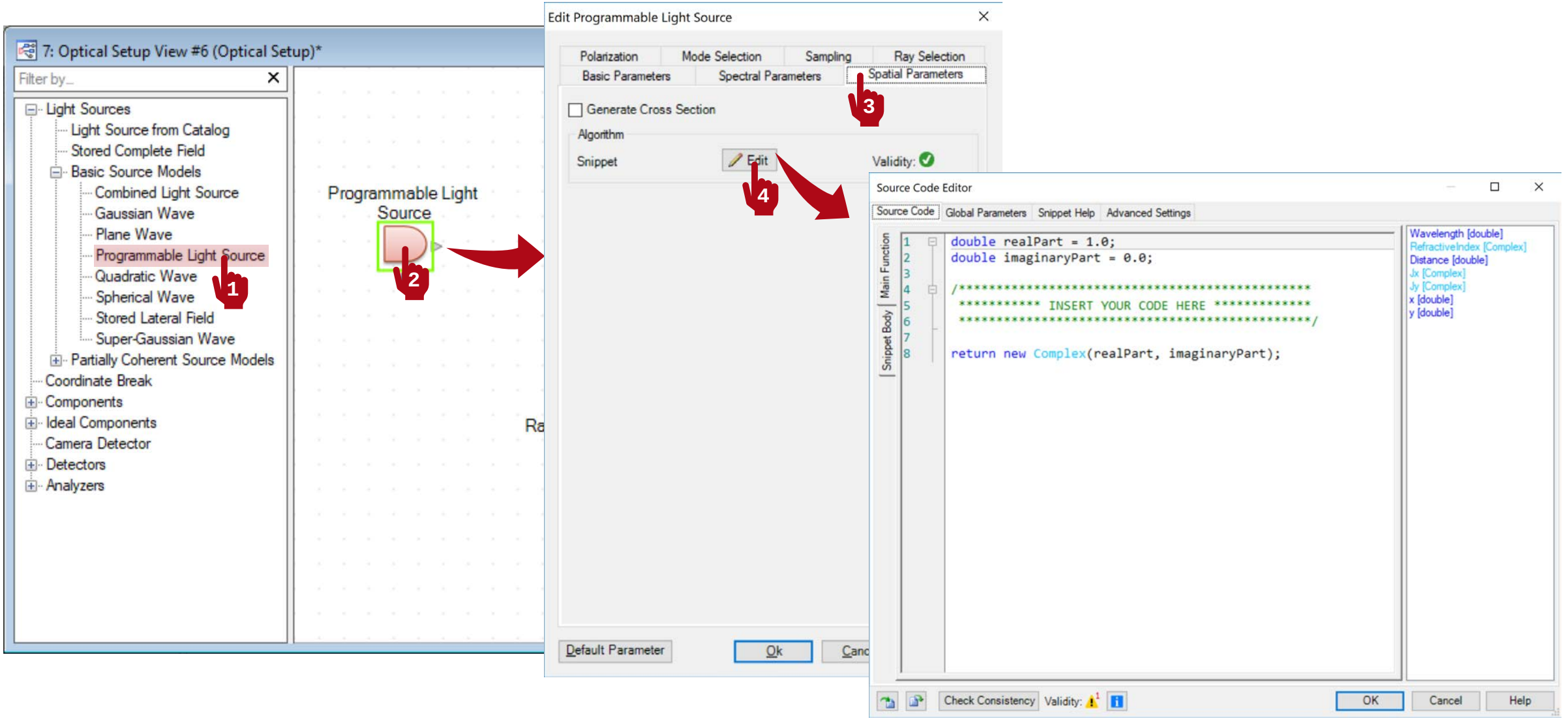
$w_x, w_y \rightarrow$ Waist radius in x and y.

Where to Find the Programmable Light Source: Catalog



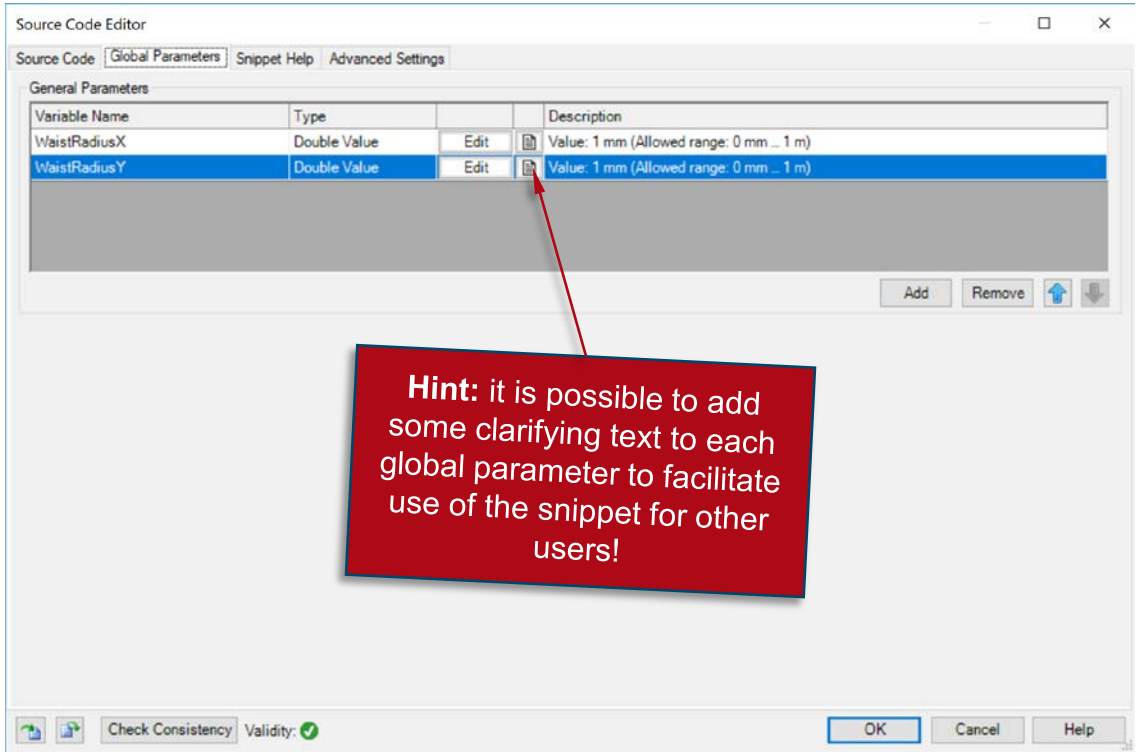


Where to Find the Programmable Light Source: Optical Setup

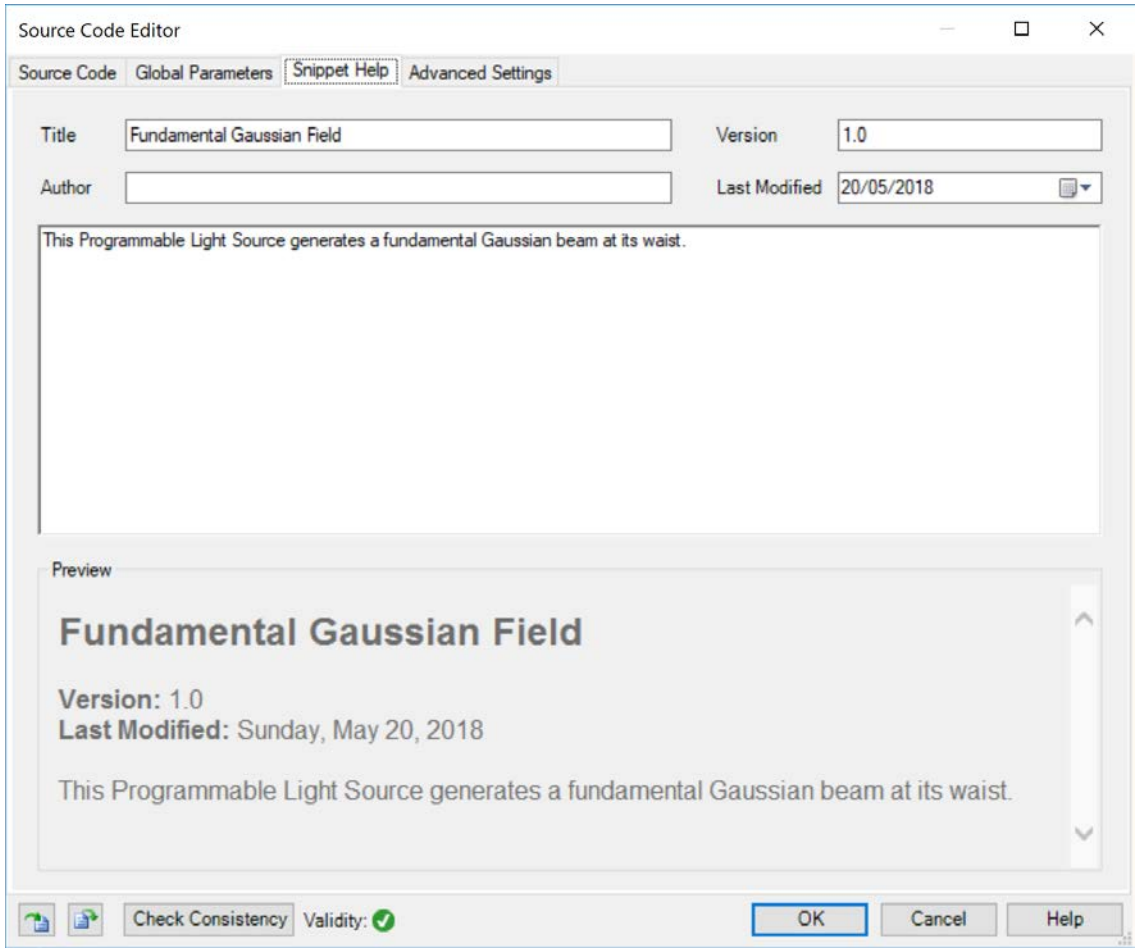


Programmable Light Source: Global Parameters

- 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.

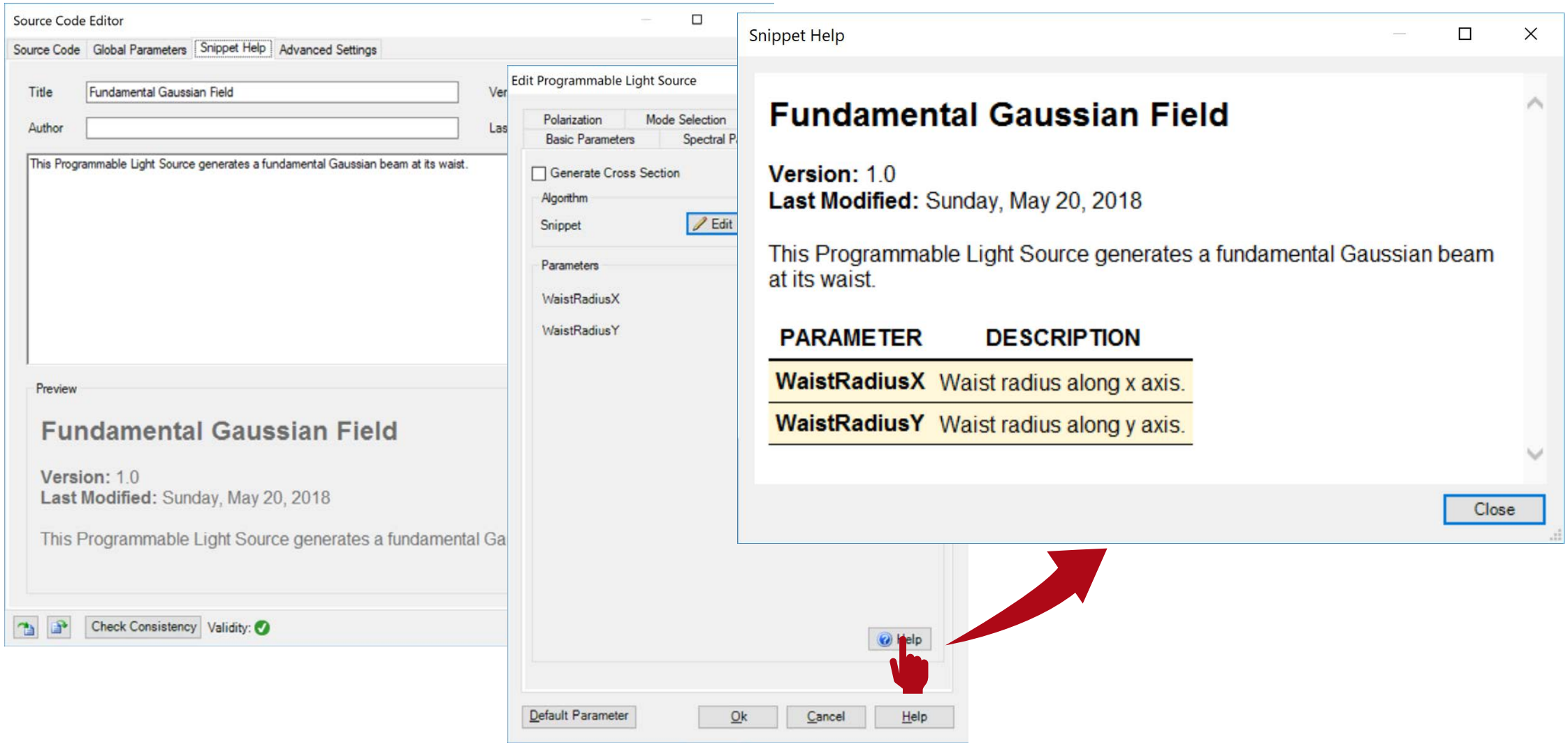


Programmable Light Source: Snippet Help

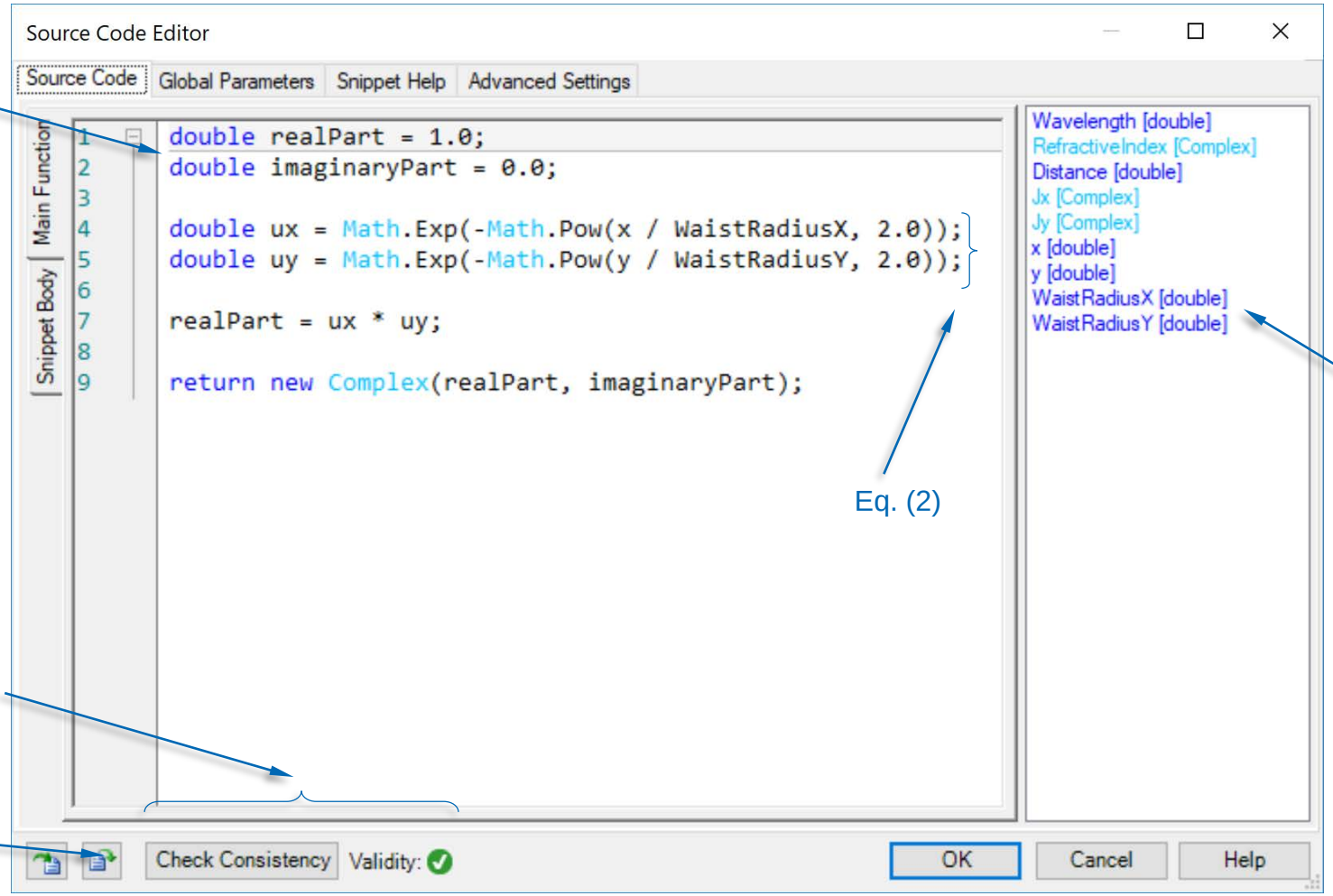


- **Optional:** you can use the Snippet Help to write instructions, clarifications, and some metadata associated to your snippet.
- This option is very helpful to keep track of your progress with a programmable element.
- It is especially useful when the programmable element is later disseminated to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

Programmable Light Source: Snippet Help



Programmable Light Source: Writing the Code



Declaration of output variable given by default

Default global parameters/variables

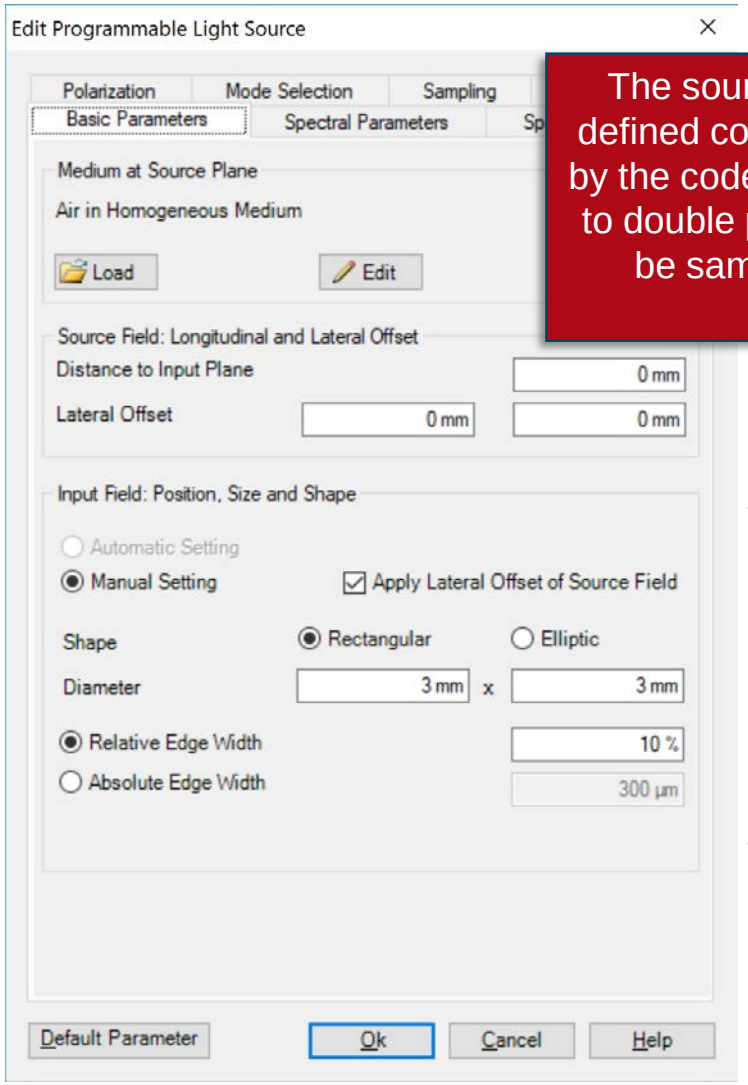
Global parameters defined by user in Global Parameters tab

Eq. (2)

Are there errors in your code?

Export Snippet to save your work!

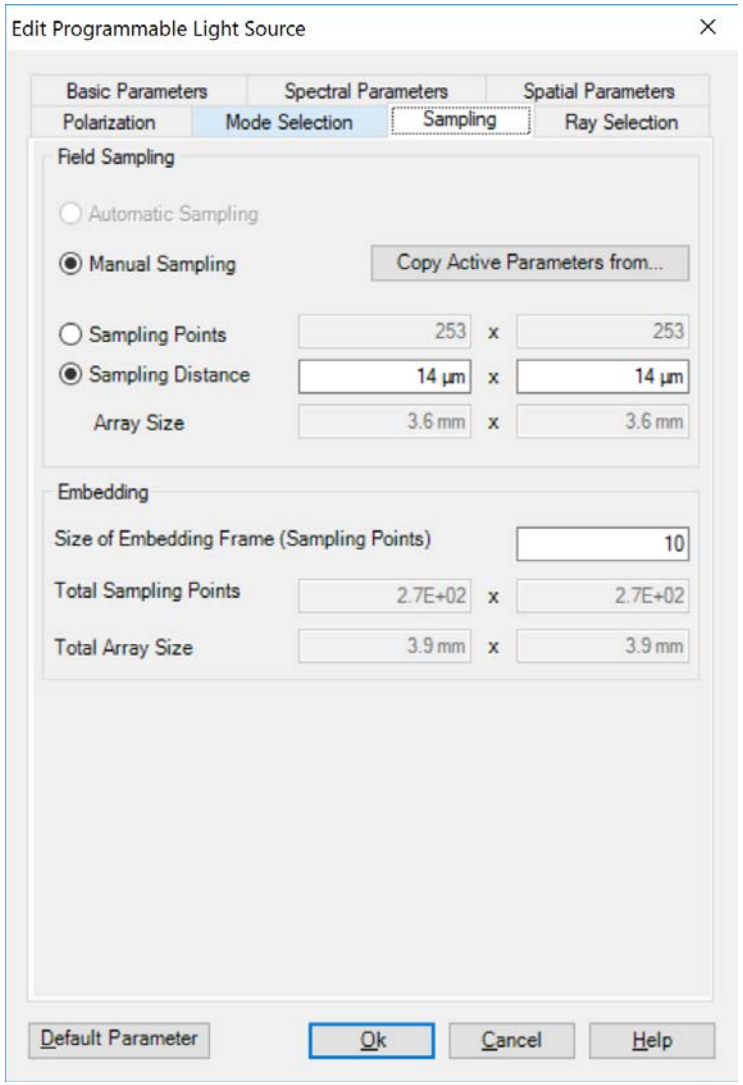
Programmable Light Source: Adjusting Sampling and Window



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

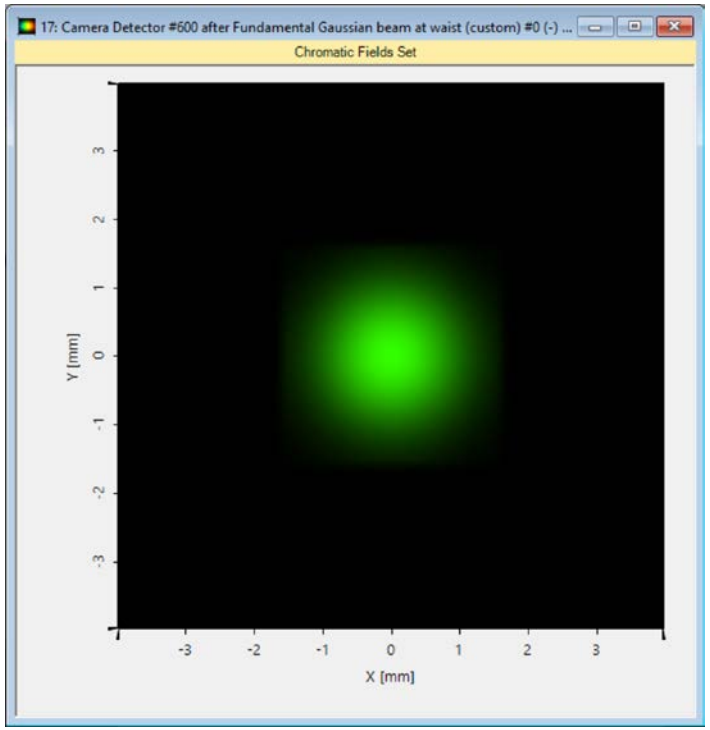


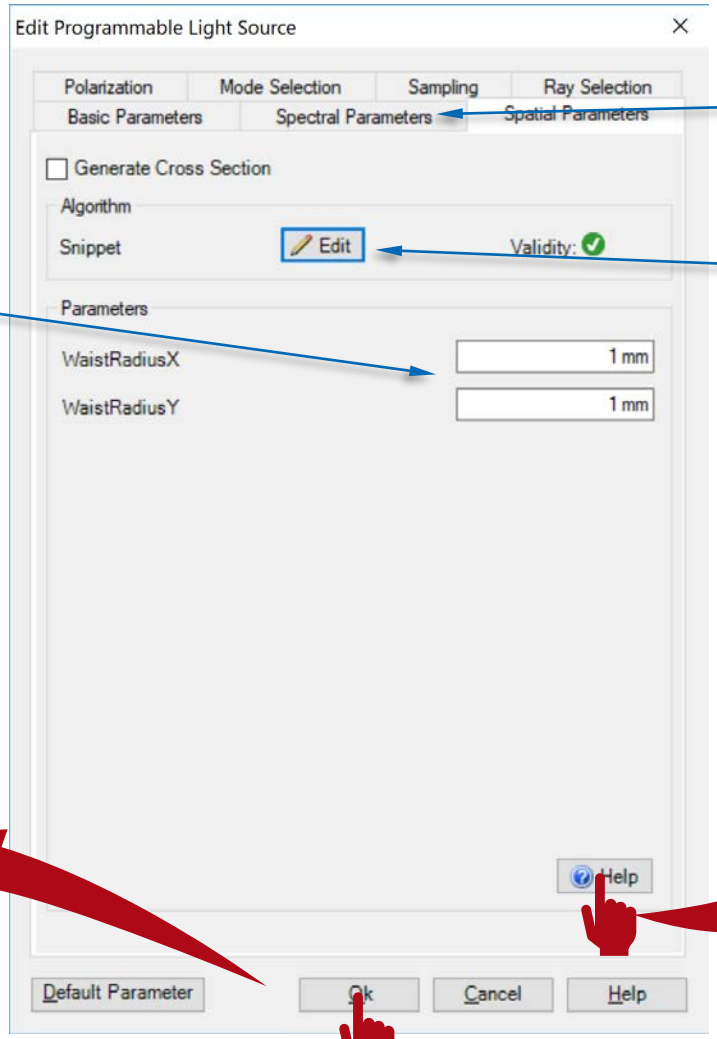
Programmable Light Source: Using Your Snippet

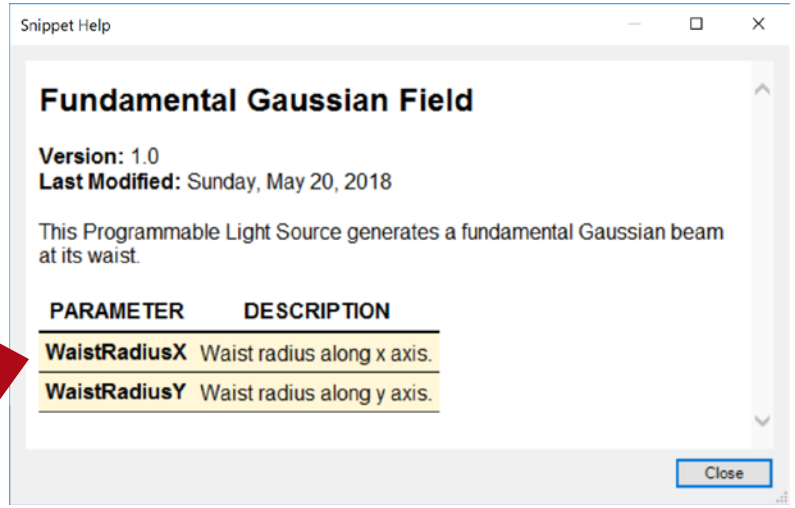
Configure the spectral make-up of the source independently in the Spectral Parameters tab

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

Modify your snippet by clicking on Edit







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 - Programmable Light Source, Function, Interface and Medium - Programming Radially & Azimuthally Polarized Sources