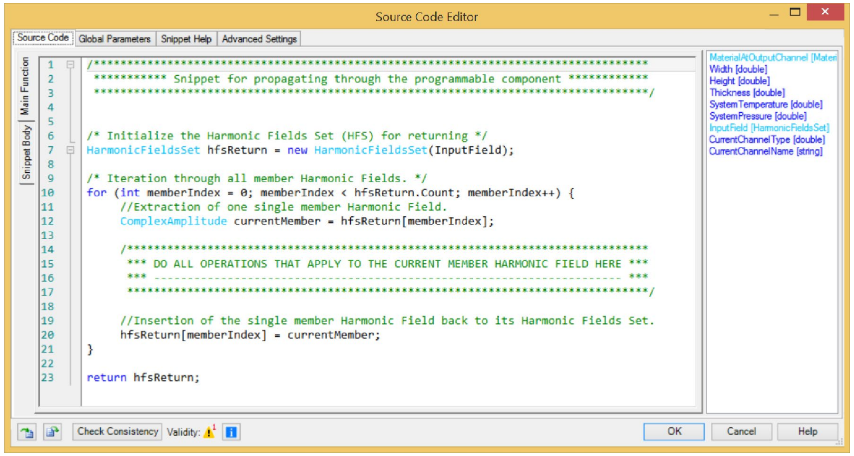
The Programmable Component

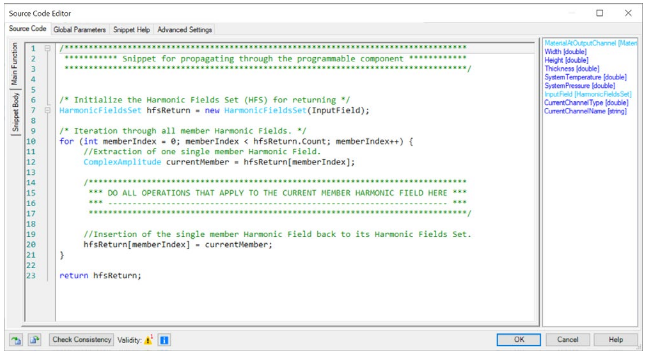
Abstract



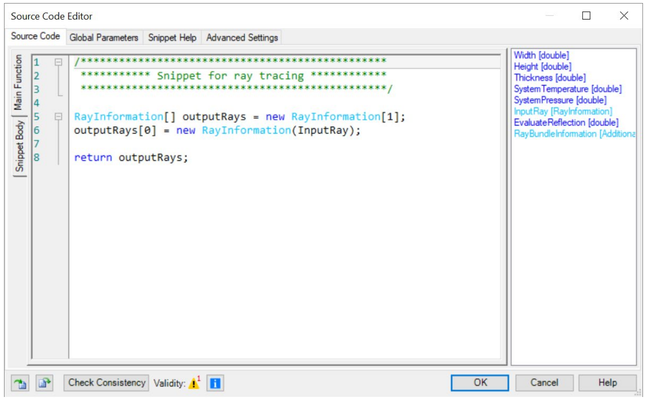
There are a various possibilities for customization of the functionality in VirtualLab Fusion. Besides generating modules for automatization and importing external DLLs, different programmable objects help the user to create tailored solutions to certain applications. This use case shows how to specify arbitrary effects to equidistant or non-equidistant field data as an optical component within optical systems. This kind of customization is realized by the programmable component.

Task

- Usage of equidistant code editor



- Usage of non-equidistant code editor

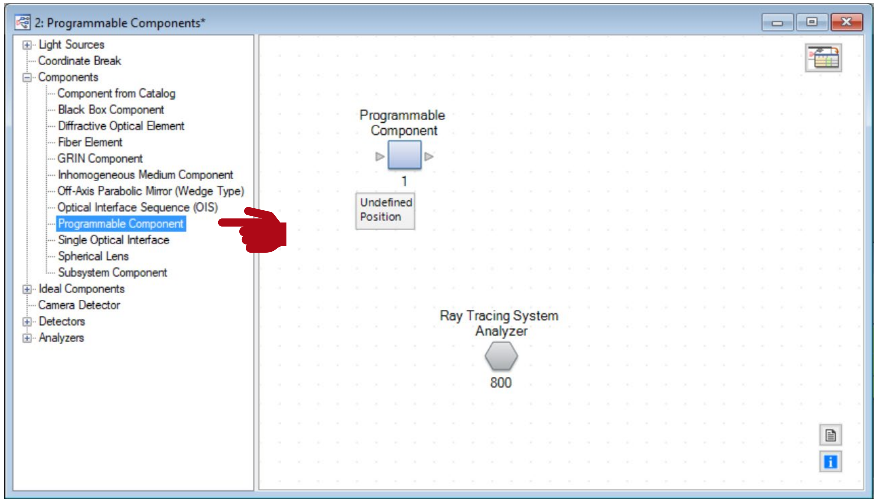


Programmable Component Initialization

Initialization

Components =>

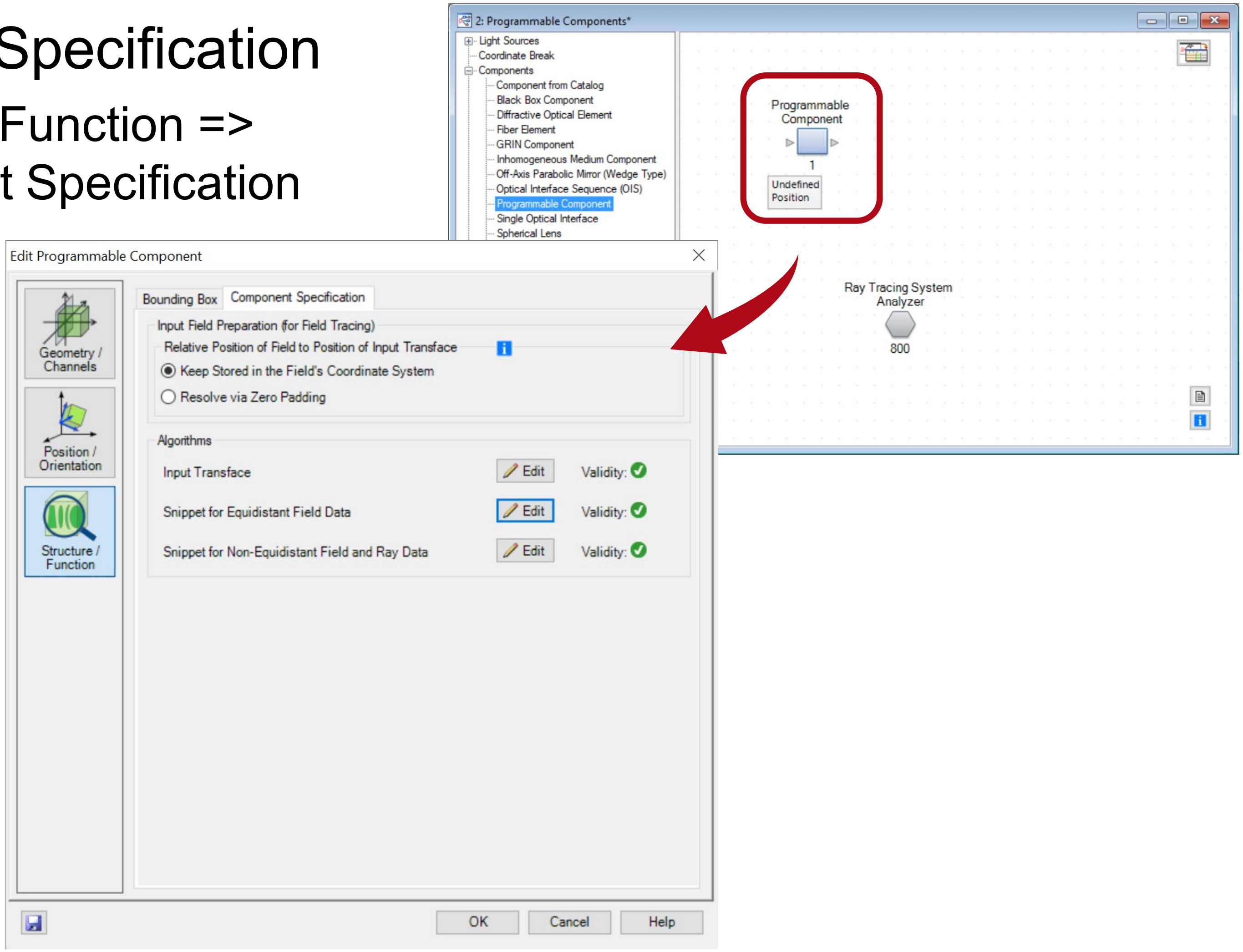
Programmable Component



Programmable Component Specification

Component Specification

Structure / Function =>
Component Specification

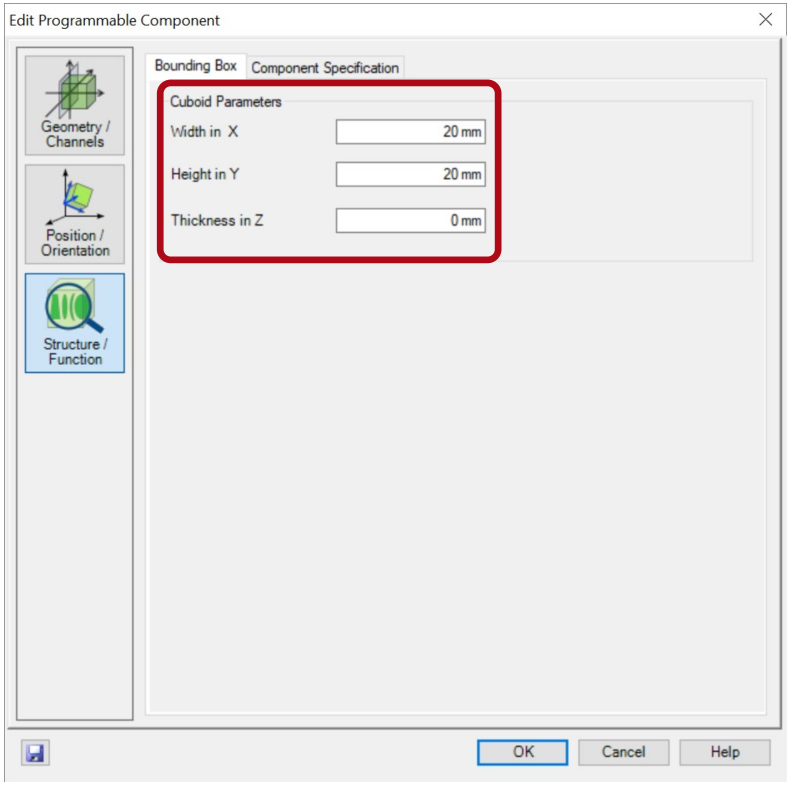


Programmable Component Specification

Specification of the Bounding Box

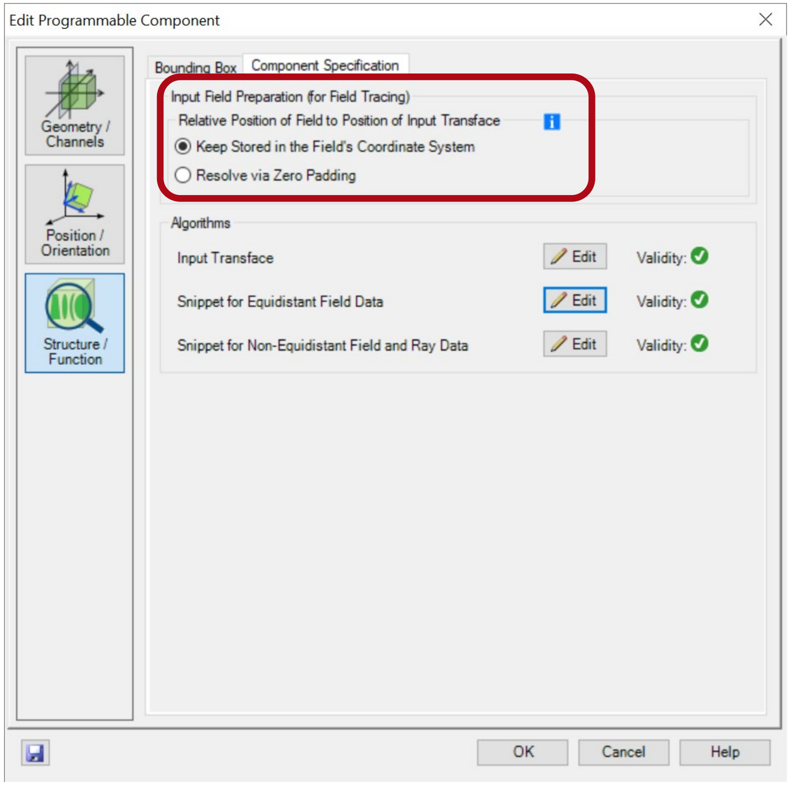
Specify the dimension of the component within a rectangle bounding box by

- width in x
- height in y
- thickness in z



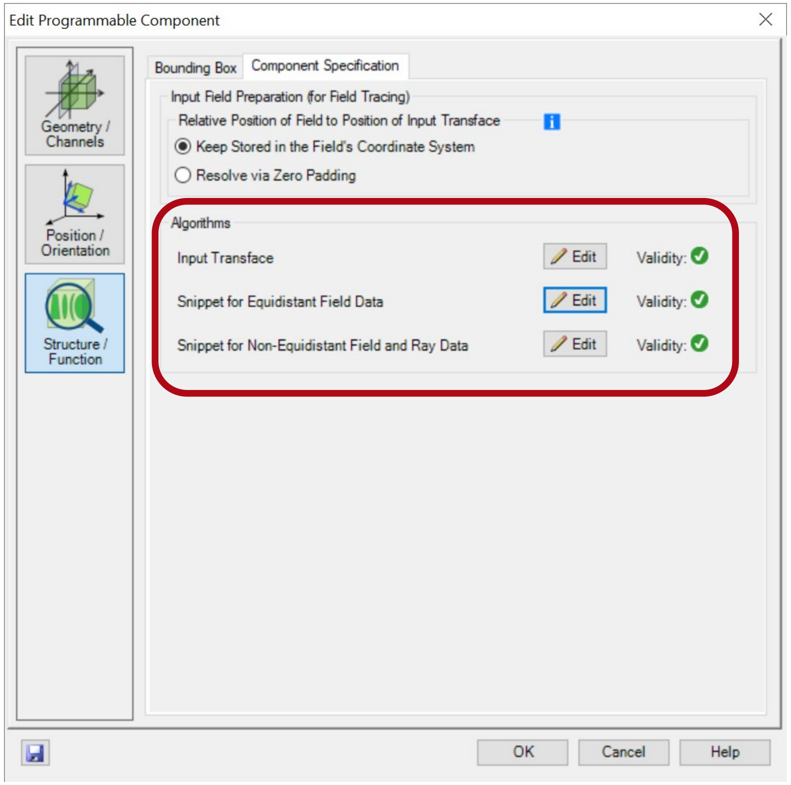
Programmable Component Specification

- Input Field Preparation
  - There are two options to define how equidistant fields should be propagated to the input transface of the component:
    - ➢ *Keep Stored in the Fields Coordinate System*
    - ➢ *Resolve via Zero Padding*



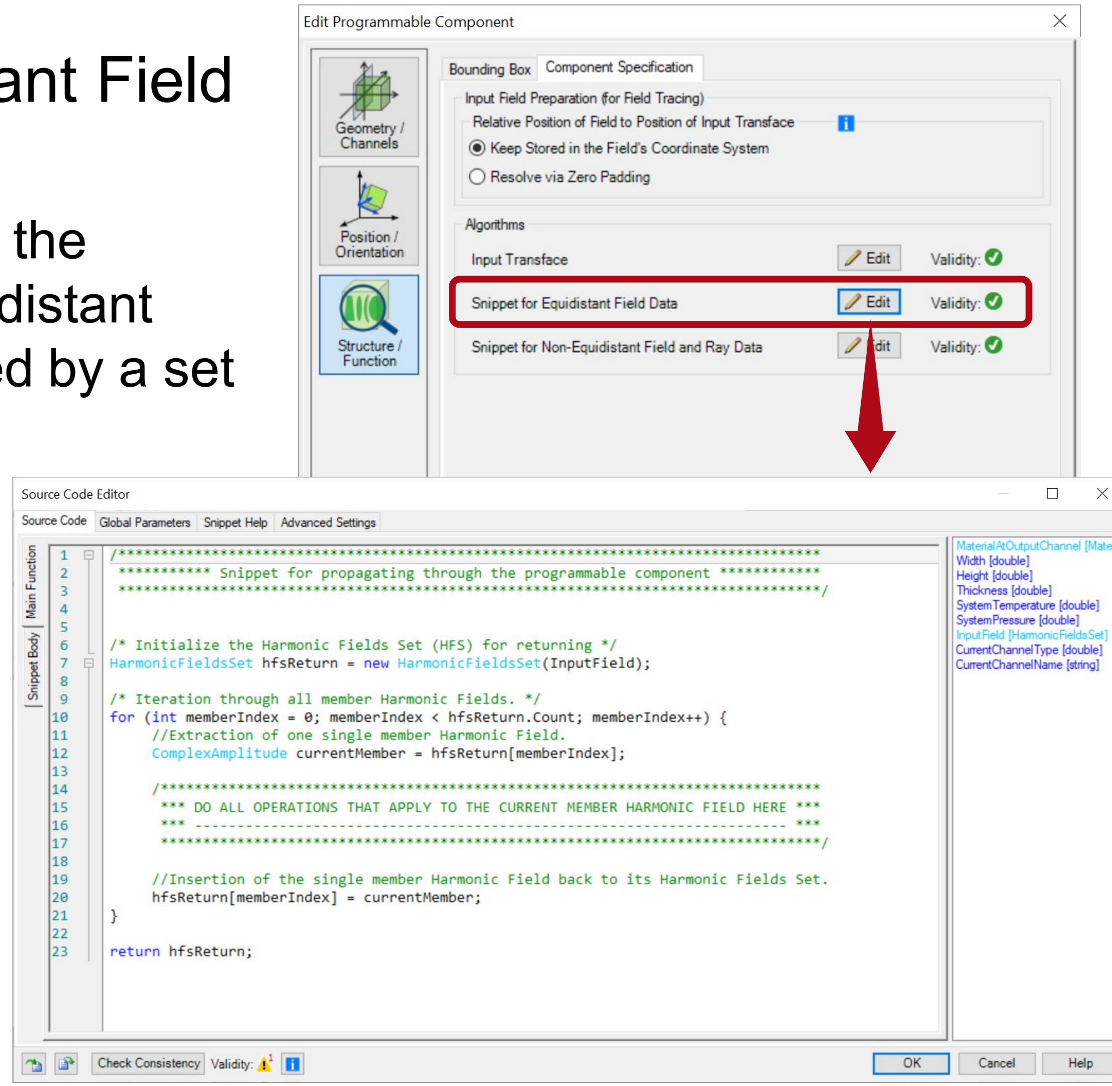
Programmable Component Specification

- Algorithms
  - The edit button provides access to the source code editor to specify the
    - *Input Transface*
    - *Snippet for Equidistant Field Data*
    - *Snippet for Non-equidistant Field and Ray Data*



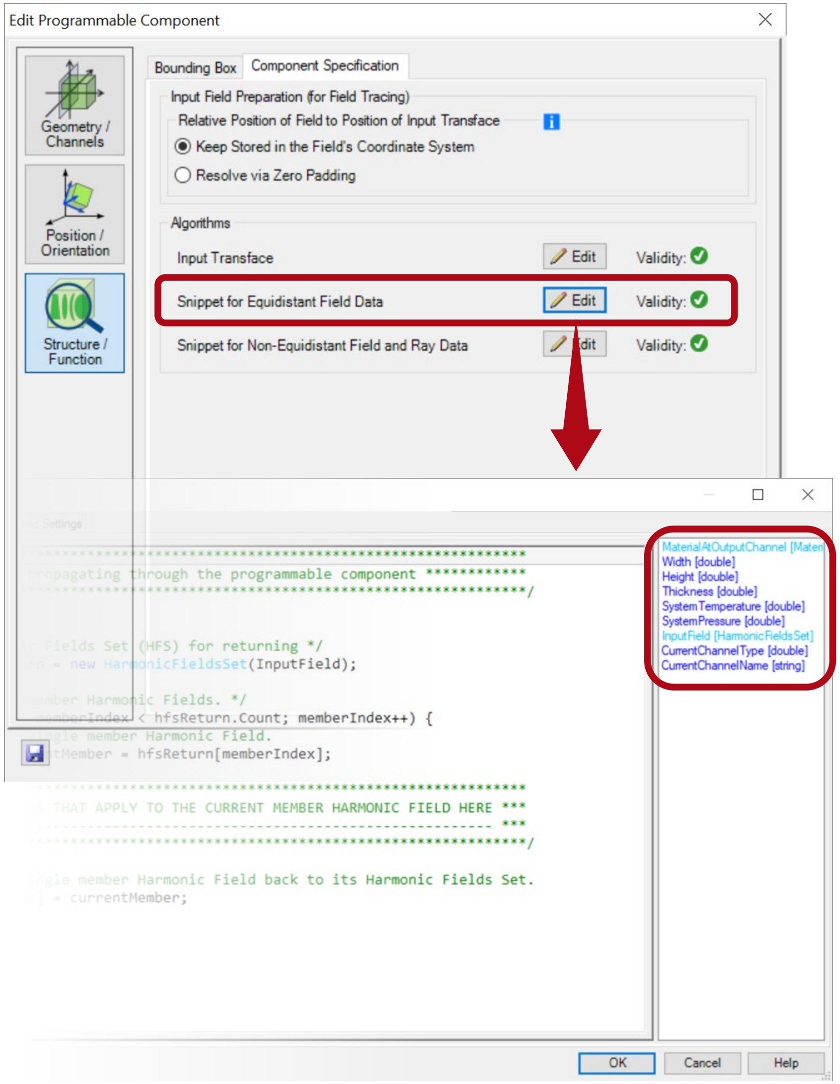
Programmable Component Algorithms

- Snippet for Equidistant Field Data
  - The snippet enables the manipulation of equidistant field data represented by a set of harmonic fields



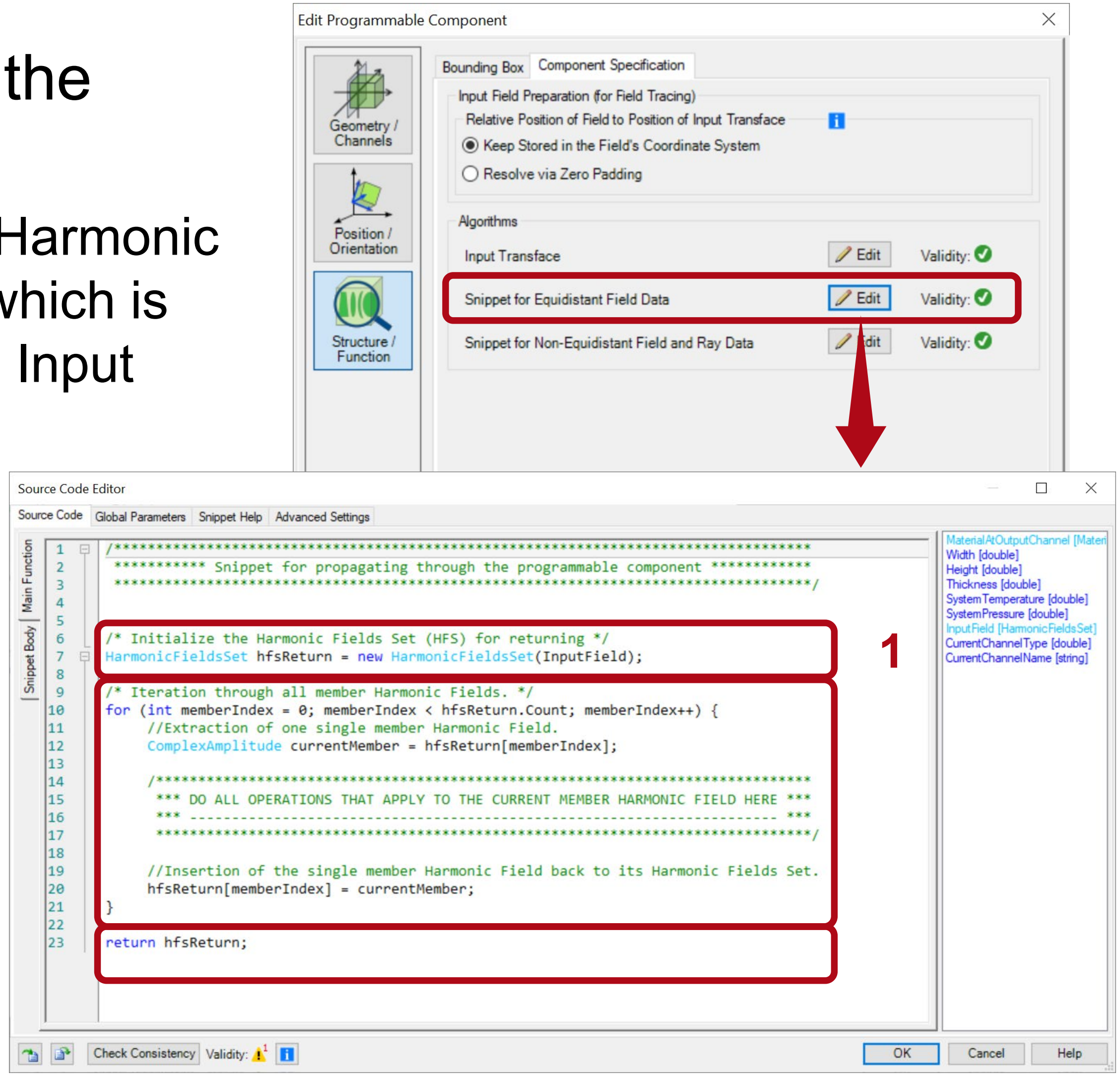
Programmable Component Algorithms

- Initial Global Parameters of the Snippet
  - Width, height and thickness of the bounding box provide the specified data from the component specification.
  - The parameter temperature and pressure are global parameters of the system.
  - Parameters of the channel specification like channel material, channel type and channel name help to define the interaction with different optical channels like e.g. transmission and reflection.



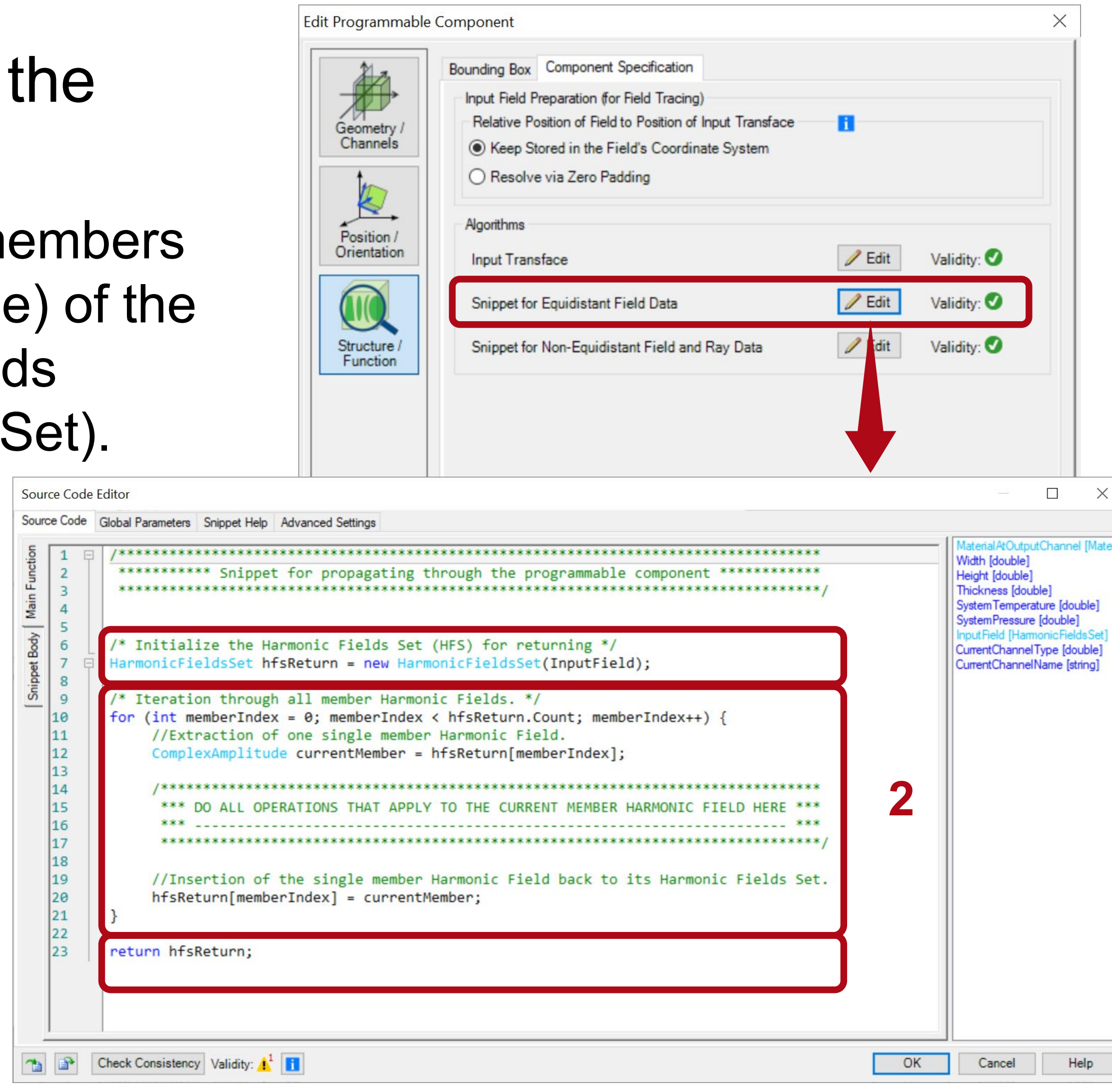
Programmable Component Algorithms

- General Concept of the Snippet
  1. Initialization of the Harmonic Fields Set output, which is initialized using the Input Field.



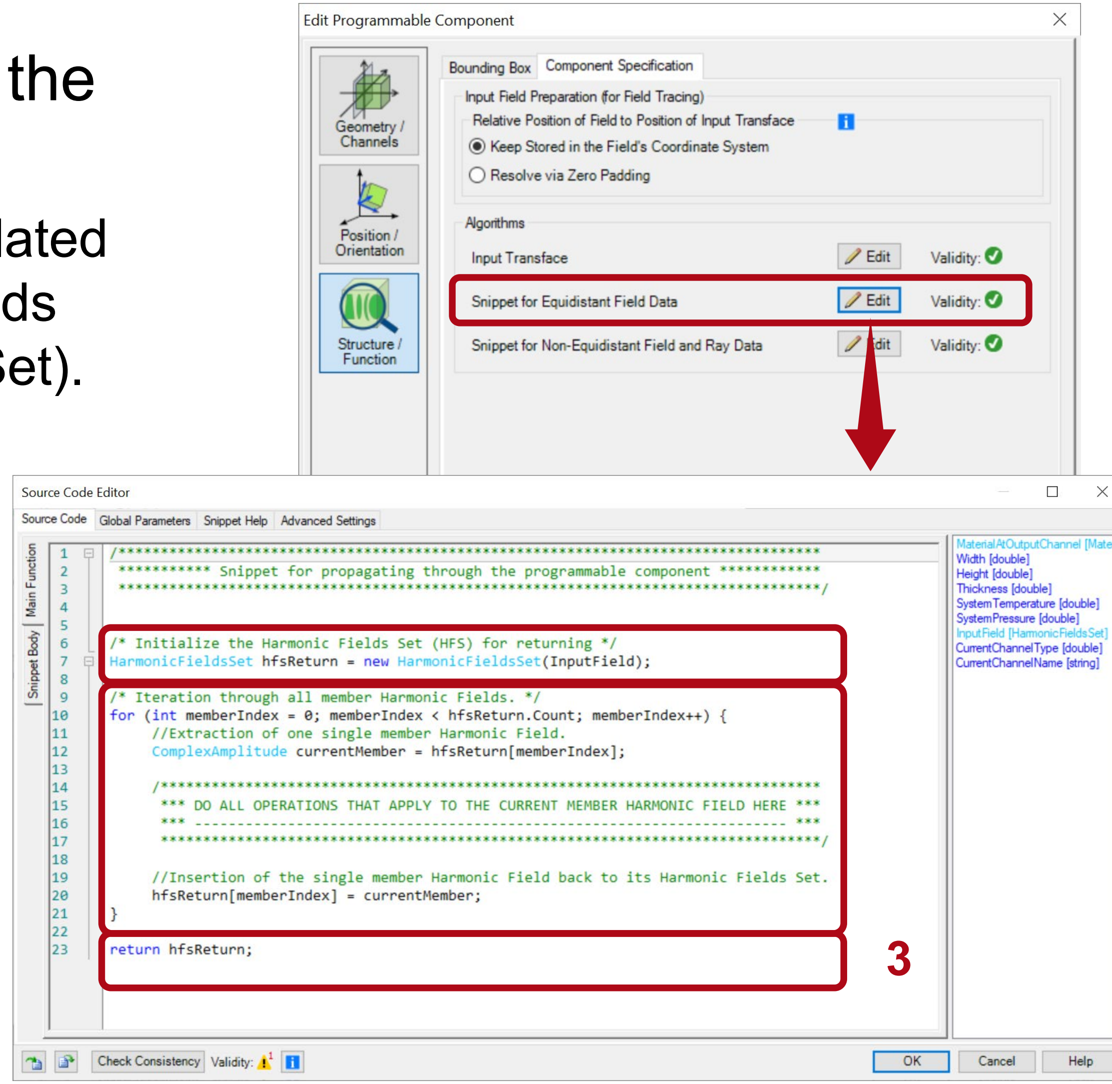
Programmable Component Algorithms

- General Concept of the Snippet
  2. Loop over all the members (Complex Amplitude) of the set of harmonic fields (Harmonics Fields Set).



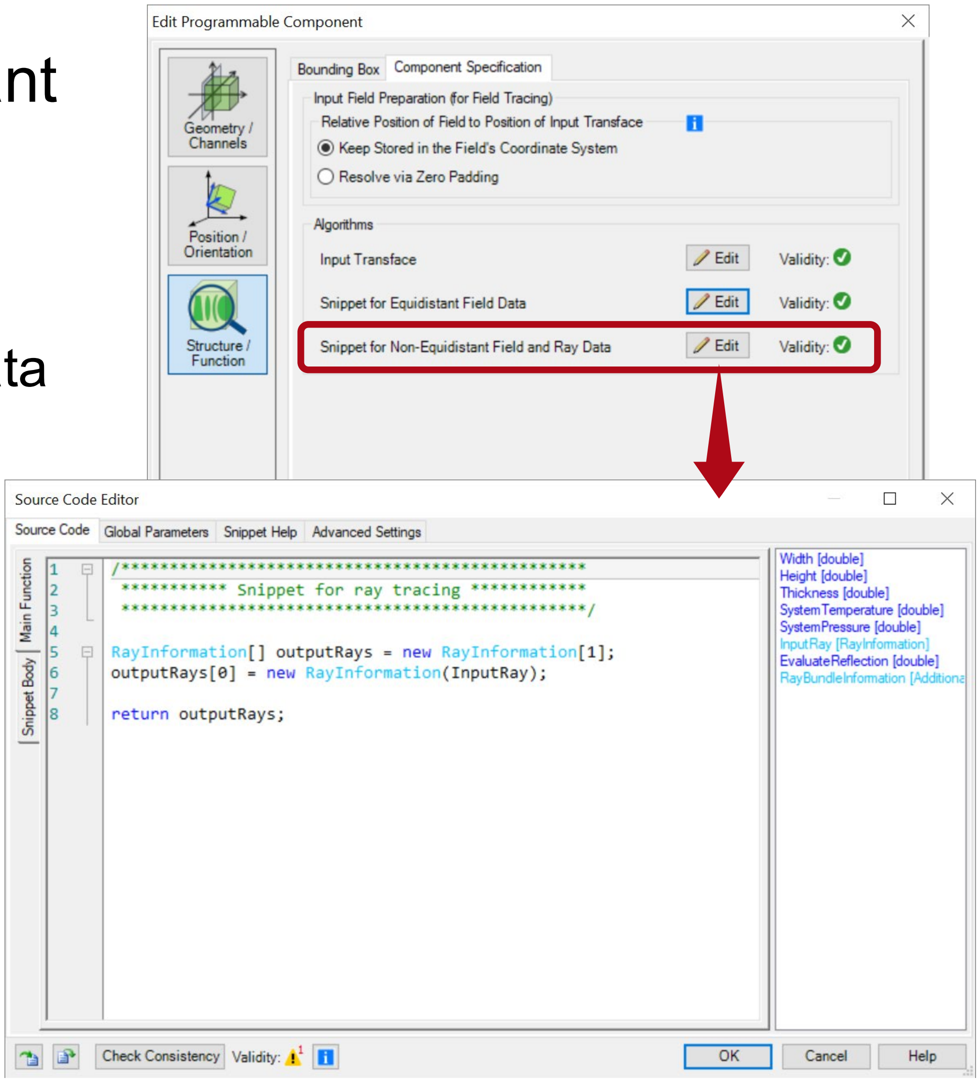
Programmable Component Algorithms

- General Concept of the Snippet
  3. Return the manipulated set of harmonic fields (Harmonic Fields Set).



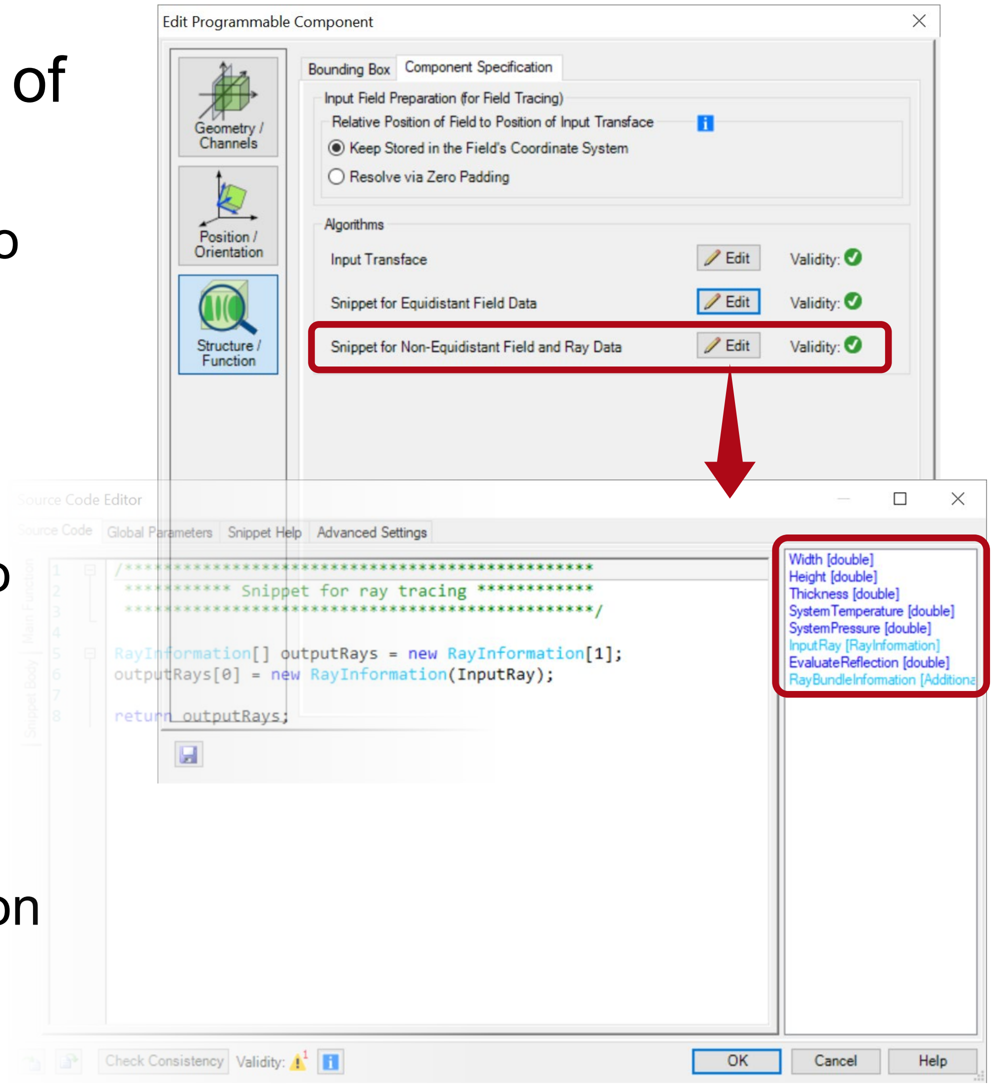
Programmable Component Algorithms

- Snippet for Non-equidistant Field Data
  - The snippet enables the manipulation of non-equidistant field and ray data represented by a set of ray bundles.



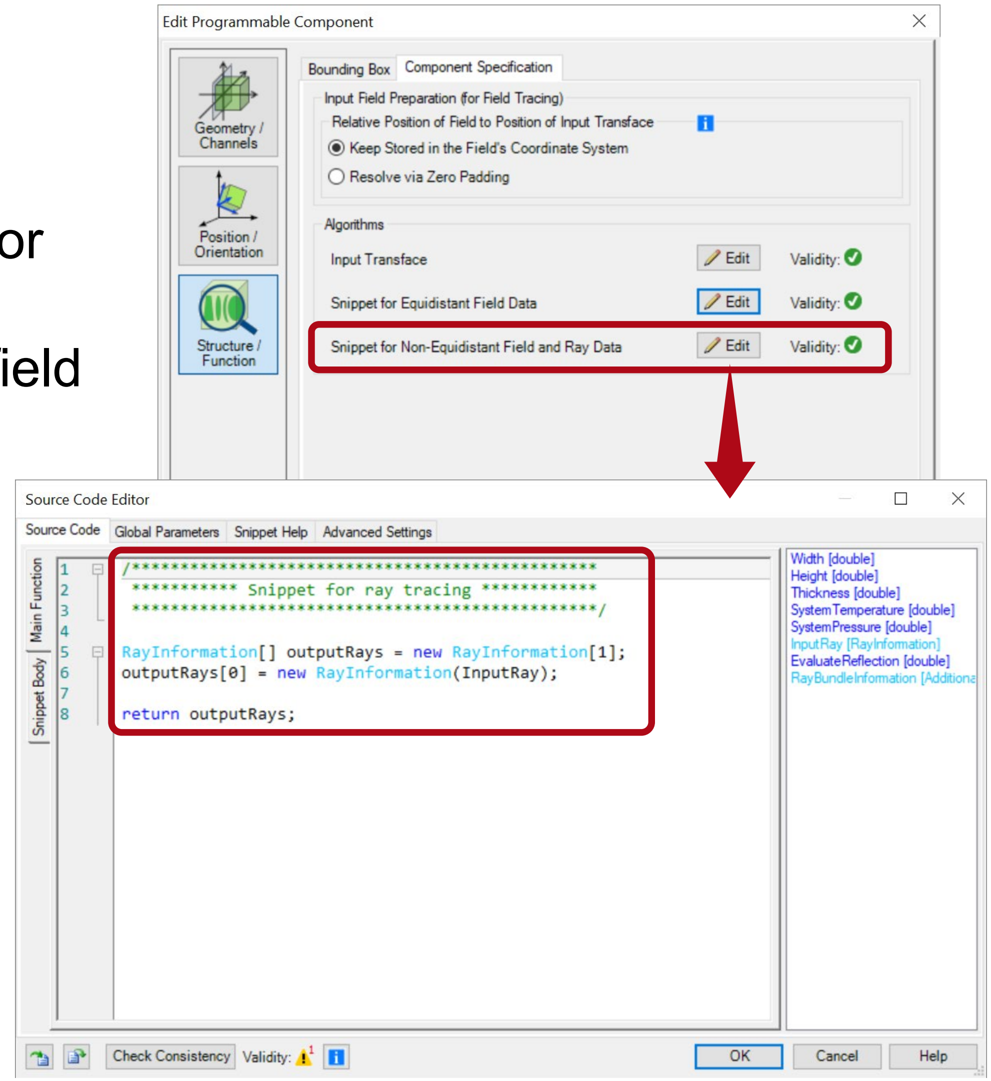
Programmable Component Algorithms

- Initial Global Parameters of the Snippet
  - The snippet can be used to describe the change of an input ray to a set of output rays.
  - The global parameter to Evaluate Reflection help to distinguish between the interaction with the optical channel's reflection or transmission.
  - The Ray Bundle Information contains the data of the corresponding ray bundle.



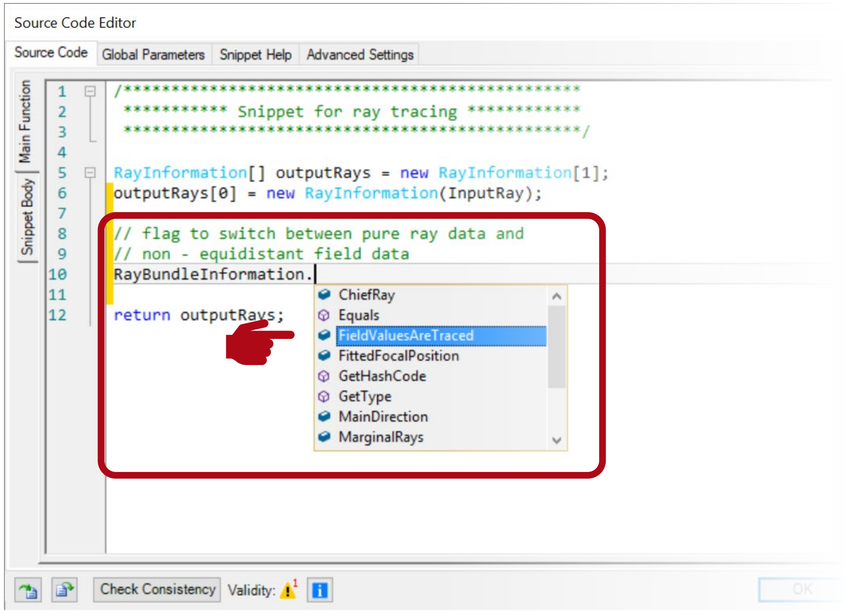
Programmable Component Algorithms

- General Concept of the Snippet
  - The snippet is performed for each ray (InputRay) of the incoming non-equidistant field data.



Programmable Component Algorithms

- General Concept of the Snippet
  - The RayBundleInformation parameter enables access to several information of the corresponding ray bundle of the current input ray.
  - For example, one can check if field values are traced to switch between the handling of pure ray data (using ray tracing engine) and non-equidistant field data (using field tracing 2nd generation engine).



Document Information

title	The Programmable Component
version	1.0
VL version used for simulations	7.0.3.4
category	Feature Use Case