General Developer Tutorial for BRAPH 2.0

The BRAPH 2 Developers

October 15, 2023

...

Contents

Software architecture

The software architecture of BRAPH 2.0 provides a clear structure for developers to understand and extend the functionalities of the software. All objects in BRAPH 2.0 are derived from a base object called Element. The core code includes the compiler (`genesis`), the essential source code (`src`), and the GUI functionalities (`gui`). Developers can easily add new elements such as brain surfaces, atlases, example scripts, GUI pipelines, graphs, measures, data types, data importers, data exporters, and analyses. By writing new elements and recompiling the code, the new elements and their functionalities are immediately integrated into the GUI.

Genesis

explain genesis process and concept of elements with pseudo code

Need for two compilations and hardcoding

braph2genesis

regenerate + explain when recompile necessary

Elements

BRAPH 2.0 is a compiled object-oriented programming software. The base class for all elements is `Element`. Each element is essentially a container for a series of *properties*. Each propery has a *category* and a *format*. Even though it is possible to create instances of Element, typically one uses its subclasses. In this section, we will see how to implement a new element.

Overview

Element, NoValue, Callback, Concrete Element + FIG

Property Categories

The category determines for what and how a property. The possible categories are:

`CONSTANT` Static constant equal for all instances of the element. It allows incoming callbacks.

`METADATA` Metadata NOT used in the calculation of the results. It does not allow callbacks. It is not locked when a result is calculated.

`PARAMETER` Parameter used to calculate the results of the element. It allows incoming and outgoing callbacks. It is connected with a callback when using a template. It is locked when a result is calculated.

`DATA` Data used to calculate the results of the element. It is NoValue when not set. It allows incoming and outgoing callbacks. It is locked when a result is calculated.

`RESULT` Result calculated by the element using parameters and data. The calculation of a result locks the element. It is NoValue when not calculated. It allows incoming callbacks.

`QUERY` Query result calculated by the element. The calculation of a query does NOT lock the element. It is NoValue when not calculated. It does not allow callbacks.

`EVANESCENT` Evanescent variable calculated at runtime (typically employed for handles of GUI components). It is NoValue when not calculated. It does not allow callbacks.

`FIGURE` Parameter used to plot the results in a figure. It allows incoming and outgoing callbacks. It is not locked when a result is calculated.

`GUI` Parameter used by the graphical user interface (GUI). It allows incoming and outgoing callbacks. It is not locked when a result is calculated.

Box Format of Properties

Property lifecycle: get, set, memorize, lock

Box all tokens

Box special tokens

Overview of Elements

Explain element structure + FIG

Implementation of an Element

Concrete Element extension

basic header and basic props

show how it works, set, get, memorize, lock

Results, Data, Parameters

locking, seeded randomness

Query

Evanescent, Gui, Figure