Advanced Usage: ElementComparators

ElementComparators allow you to define your own way of verifying that values output from a pipeline and displayed in XNAT are correct. NRG_Selenium comes with support for the following comparators:

ElementComparator	"type"	Extra config required
EqualityElementComparator	"Equals"	None
MinPercentToleranceElementComparator	"minPercentTolerance"	Minimum percent tolerance allowed (e.g. 1.5), specified as "percent" property.
MinPercentToleranceElementComparator	"maxPercentTolerance"	Maximum percent tolerance allowed (e.g. 2.3), specified as "percent" property.

When creating the YAML file for element comparison, the root level should specify "comparators" with a list of objects. Each comparator object must include the "type" property matching one of the above options. Additionally, some comparators may require additional parameters to be used correctly. Each comparator object should also include an "elements" property which points to a list of element objects. An element object consists of two properties, a "value" with the expected value of the element, and a "locator" property. If the "locator" property is simply a text string, it will be interpreted as an xpath string by which the element can be located. Otherwise, it should be another object with a "type" and "value", where the type is either "id" or "xpath", specifying that the text in "value" is either the DOM id for the element, or an xpath selector. A full example can be seen below:

ElementComparators YAML example

```
comparators:
- type: Equals
  elements:
  - locator: //div[@id='whitematter']
    value: 44.4
  - locator:
      - type: xpath
        value: //div[@id='graymatter']
    value: 55.5
  - locator:
      - type: id
        value: errors
    value: No errors.
- type: maxPercentTolerance
  percent: 1.0
  elements:
  - locator: //div[@id='whitematter']
    value: 44.4
```

✅ A quick tip about **MinPercentToleranceElementComparator:** this comparator specifies that the percent error between the values must be **greater** than some specified value. The maximum version is much more useful and logical, but it was easy to include this version on the off-chance of it being useful.