

## TYPO3.TypoScript - Task # 48361

Work Package # 48275 (New): TypoScript consistency

<b>Status:</b>	Resolved	<b>Priority:</b>	Should have
<b>Author:</b>	Sebastian Kurfuerst	<b>Category:</b>	
<b>Created:</b>	2013-05-17	<b>Assigned To:</b>	Sebastian Kurfuerst
<b>Updated:</b>	2013-10-11	<b>Due date:</b>	
<b>Subject:</b>	Kill processor classes; change processor syntax		
<b>Description</b>	<p>the whole team feels that processors are somewhat not fitting into the TS concept; especially because many things can be done as well using Eel; and we want to provide one consistent way of doing things.</p> <p>This means:</p> <ul style="list-style-type: none"><li>- we want to <b>remove processor classes</b>; instead the functionality should be made available in the Eel standard library</li><li>- we want the following syntax for processors:</li></ul> <pre>page.title = "foo"  // short syntax page.title.@process.wrapInTitle = \${ "before" + value + "after" }  // long position syntax page.title.@process.wrapInTitle {   expression = \${ "before" + value + "after" }   @position = "end 50" }</pre> <ul style="list-style-type: none"><li>- positional arguments as known from TypoScript should just work here.</li><li>- we are not yet sure if the "current value" variable should be called "value" or differently. Same, it is not yet clear whether the long form should be called "expression".</li></ul> <p>Together with #48359 this means one could also write custom TypoScript objects and use them as "processors".</p> <p>We all think this is more powerful and consistent than the old processor implementation.</p>		
<b>Related issues:</b>	related to TYPO3.TypoScript - Task # 48359: Simple Value / Eel Expression / T... <b>Resolved</b> <b>2013-05-17</b>		

### Associated revisions

#### Revision db4ae808 - 2013-10-11 21:32 - Rens Admiraal

[[!]]**[FEATURE]** re-implement Processors based on TypoScript Objects and Eel

This is a major overhaul of the TypoScript processors feature.

Before this change, processors were written such as::

```
myProperty = "some text"
myProperty << 1.wrap(after: '...')
```

However, then, we introduced Eel objects, and it became more and more clear that processors, while they are very important from a users standpoint, lead to very much duplicated code.

That's why we are dropping the above syntax completely, replacing it with the following:

New Processors=====

::

```
myProperty = "some text"
  myProperty.@process.1 = ${value + '...'}

```

- We use the @process meta-property to run processors.
- The index afterwards describes the ordering on which processors are applied.
- "value" is a special context variable available in processors which contains the to-be-processed content.

Extended Syntax

-----

Furthermore, there exists an extended syntax as follows:

```
myProperty = "some text"
  myProperty.@process.foo {
    expression = ${value + '...'}
    @position = 1
  }

```

- This allows to use **named processors**.
- For @position, every positional argument is valid, such as "start", "end", [any number], "before [otherkey]", "after [otherkey]". That's extremely powerful!

Using TypeScript Objects or Eel Expressions

-----

Instead of using Eel expressions, you can use arbitrary TypeScript objects for processing stuff, as in the (contrived) example below::

```
myProperty.@process.1 = Some.Namespace:SomeTypeScriptObject {
  value = ${value + '...'}
}

```

Unsetting of Processors

-----

Unsetting processors was not possible before; now that's easily done using the ">" operator::

```
myProperty.@process.1 >
```

That also works correctly when being combined with prototype inheritance.

Using TypeScript Objects in @override

-----  
Furthermore, this change also adjusts the @override functionality such that a context variable's value is the return value of a TypoScript object::

```
// this worked already before this change:  
myObject.@override.myVariable = ${myOtherVariable + '...'}  
  
// this works now as well:  
myObject.@override.myVariable = Value {  
    value = ${myOtherVariable + '...'}  
}
```

#### Code Cleanup

-----

This feature has been developed in a very test-driven manner, every functionality is well-covered with tests. Additionally, the tests and the code of the runtime and parser are cleaned up. The following features were removed as they are obsolete:

- "old" processor syntax and implementations
- relicts of TypoScript variables (were non-functional before)
- relicts of "<<" operator (was non-functional before)

Additionally, the internal representation of TypoScript objects has been changed a little, to make sure you can also set processors on e.g. simple strings.

Furthermore, it is now made sure that if e.g. an Eel expression is defined, this expression **overrides** a TypoScript object or a simple value which has existed beforehand on a given TypoScript path. The same has been done as well for simple values and TypoScript objects. This is to make sure that at any given TypoScript path, there can only be **either** a TypoScript object, **or** an Eel expression, **or** a simple type. That improves general stability.

When testing this feature, make sure to check out the related changesets with the same topic as well.

Resolves: #48361

Resolves: #43752

Change-Id: lee6b6a839a56416c6d1cc72cd92ece049a3fba1f

Reviewed-on: <https://review.typo3.org/24423>

Reviewed-by: Sebastian Kurfuerst

Tested-by: Sebastian Kurfuerst

Reviewed-by: Rens Admiraal

Tested-by: Rens Admiraal

## History

---

#1 - 2013-10-06 13:06 - Gerrit Code Review

- Status changed from Accepted to Under Review

Patch set 1 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #2 - 2013-10-08 08:19 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #3 - 2013-10-09 14:27 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #4 - 2013-10-09 14:39 - Bastian Waidelich

Sebastian Kurfuerst wrote:

Great, looking forward to that ;)

| - we are not yet sure if the "current value" variable should be called "value" or differently.

What about "subject" - we use that a lot in format ViewHelpers and it is more generic than "value".

| Same, it is not yet clear whether the long form should be called "expression".

Would it make sense to stress that it's an Eel expression or should that be clear?

Also I'm wondering if the process key ("wrapInTitle" in this example) is used for sorting if @position is not set, or - to put it differently - can I write:

```
page.title.@process.20 = ${ "a" + value + "b" }
```

```
page.title.@process.10 = ${ "c" + value + "d" }
```

and I would get "ac<value>db" ?

#### #5 - 2013-10-09 16:01 - Gerrit Code Review

Patch set 4 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #6 - 2013-10-09 16:07 - Sebastian Kurfuerst

| - we are not yet sure if the "current value" variable should be called "value" or differently.

| What about "subject" - we use that a lot in format ViewHelpers and it is more generic than "value".

OK, we could also do that. I'll open a doodle for that.

| Same, it is not yet clear whether the long form should be called "expression".

| Would it make sense to stress that it's an Eel expression or should that be clear?

It does not have to be an eel expression, it **can** as well be a TypoScript object which re-uses "value" (or "subject" if we decide to rename it)

| Also I'm wondering if the process key ("wrapInTitle" in this example) is used for sorting if @position is not set, or - to put it differently - can I write:

| `page.title.@process.20 = ${ "a" + value + "b" }`

| `page.title.@process.10 = ${ "c" + value + "d" }`

| and I would get "ac<value>db" ?

yes that's the way it works :)

Greetings, Sebastian

#### #7 - 2013-10-10 11:29 - Gerrit Code Review

Patch set 5 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #8 - 2013-10-10 22:01 - Gerrit Code Review

Patch set 6 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #9 - 2013-10-11 21:06 - Gerrit Code Review

Patch set 7 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #10 - 2013-10-11 21:28 - Gerrit Code Review

Patch set 8 for branch **master** has been pushed to the review server.

It is available at <https://review.typo3.org/24423>

#### #11 - 2013-10-11 21:36 - Rens Admiraal

- Status changed from Under Review to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:db4ae808075218de7eab09cfed178c2f1ace25ad.