

TYPO3.TypoScript - Feature # 43752

Status:	Resolved	Priority:	Should have
Author:	Marc Neuhaus	Category:	
Created:	2012-12-08	Assigned To:	Marc Neuhaus
Updated:	2013-10-11	Due date:	
Subject:	Add support to unset paths and processors		
Description	<p>Currently it's not possible to unset Paths or Processors.</p> <p>To unset paths i'd suggest to reuse the pattern that's known from typoscript1:</p> <pre>prototype(TYPO3.Expose:RecordList) { controllerLinks > }</pre> <p>To unset processors we should do something similar to this:</p> <pre>prototype(TYPO3.Expose:RecordList) 100 >></pre>		

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 TypeScript 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 - 2012-12-08 15:31 - Marc Neuhaus

i dug a bit around an realised unsetting paths with ">" is already implemented.

The Processing types on the other hand don't.

#2 - 2012-12-12 11:32 - Marc Neuhaus

- *Assigned To set to Marc Neuhaus*

#3 - 2013-10-10 11:29 - Gerrit Code Review

- *Status changed from New to Under Review*

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

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

#4 - 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>

#5 - 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>

#6 - 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>

#7 - 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.