TYPO3.TypoScript - Bug # 39865

Work Package # 48275 (New): TypoScript consistency

Status:	Resolved	**Priority:**	Must have
Author:	Sebastian Kurfuerst	**Category:**	
Created:	2012-08-16	**Assigned To:**	Sebastian Kurfuerst
Updated:	2013-05-24	**Due date:**	
Subject:	create proper prototype inheritance		

Description

Imagine:

```
prototype(Matcher)
prototype(SpecializedMatcher) < prototype(Matcher)

prototype(SomeCaseStatement) < prototype(Case) {
  # Question: What to do if "matcher.x" is an extension of Matcher? how do we change the prototype for ALL matchers then?
  # - Idea: remember "inherited" types
  prototype(Matcher).@override {
    identifier = ${identifierPrefix + '.' + name}
  }

  foo = SpecializedMatcher
  # now, as foo is a subtype of Matcher, the @override from above should also apply for here :-)
  # however, this does not happen right now, and implementing it would probably be difficult...
```

Associated revisions

Revision a9717553 - 2013-05-22 21:17 - Sebastian Kurfuerst

[!!!][FEATURE] Implement real prototype inheritance

Before this change, the following behavior happened::

```
prototype(A).test = 'val1'
```

1. this was the "copy" operator

```
prototype(B) < prototype(A)
```

1. thus, the property 'test2' is NOT set on prototype(B), but
2. just on prototype(A)

```
prototype(A).test2 = 'val2'
```

With this change, prototype(B) also has property "test2" set; making TypoScript more ordering independent. The "<" operator on prototypes now always sets up the prototype inheritance chain, while on simple properties it copies as before.

This also works with context-dependent prototypes such as prototype(Foo).prototype(Bar) - this also takes the inheritance into

account, if Foo e.g. has a parent type.

We completely removed the functionality to **copy** TypoScript prototypes.

Currently, setting up the inheritance chain is only allowed on top level, such as "prototype(Foo) < prototype(Bar)", but NOT inside a nested path such as "foo.prototype(Foo) < prototype(Bar)". While this could be theoretically possible, it makes reasoning about the behavior of TypoScript a lot more complicated -- that's why we disallow this behavior.

This change also removes the non-used "=<" operator.

In the longer run, e.g. after profiling TypoScript performance, we can optimize performance by calculating the inheritance chains during compilation, and not during runtime.

Resolves: #39865
Change-Id: Icfba5063e51948f065e8e315240b59ae67f89c98

History

#1 - 2013-01-17 12:05 - Gerrit Code Review

Patch set 1 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/17573

#2 - 2013-01-17 12:07 - Sebastian Kurfuerst

- Status changed from New to Under Review
- Assigned To set to Sebastian Kurfuerst
- Priority changed from Should have to Must have

#3 - 2013-05-17 00:47 - Sebastian Kurfuerst

- Parent task set to #48275

#4 - 2013-05-22 21:04 - Gerrit Code Review

Patch set 2 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/17573

#5 - 2013-05-22 21:17 - Gerrit Code Review

Patch set 3 for branch **master** has been pushed to the review server.
It is available at https://review.typo3.org/17573

#6 - 2013-05-24 16:36 - Sebastian Kurfuerst

- Status changed from Under Review to Resolved
- % Done changed from 0 to 100

Applied in changeset commit:a9717553d380e8bf0495c8688c6e71dcb4a4e20e.