Core - Bug # 28120

Status:	Closed	**Priority:**	Should have
Author:	Chris topher	**Category:**	TypoScript
Created:	2011-07-12	**Assigned To:**	
Updated:	2013-06-01	**Due date:**	
TYPO3 Version:	4.6		
PHP Version:	5.3		
Complexity:			
Is Regression:			
Sprint Focus:			
Subject:	stdWrap.hash: Return nothing for non-existing algorithm		

Description

#28095 introduced the new property "hash" for stdWrap.
Here is the diff: commit:d2e68695fd4e55d888544d7b7b8b699544ab16ed

The following could still be improved:

Add a check if $algorithm contains a value, which is present as a hashing algorithm on the system. If it is not, use a default.
Currently you will get a PHP error, if there is no hashing algo with the name specified in $algorithm...

Idea from Stefan Neufeind:
Please don't hardcode the default. Maybe fetch as singleton via hash_algos()?

Related issues:

follows Core - Feature # 28095: Add hashing algorithms to stdWrap	**Closed**	**2011-07-11**

Associated revisions

Revision 9c255a38 - 2011-07-14 21:18 - Stefan Neufeind

[BUGFIX] stdWrap.hash: Return nothing for non-existing algorithm

Return an empty string to prevent disclosing information unhashed.

Change-Id: Id25b85de039797aa7e39225fb0e2b1c75207a505
Resolves: #28120
Reviewed-on: http://review.typo3.org/3324
Reviewed-by: Jo Hasenau
Tested-by: Jo Hasenau
Reviewed-by: Xavier Perseguers
Tested-by: Xavier Perseguers

History

#1 - 2011-07-12 16:21 - Xavier Perseguers

- Target version changed from 1281 to 4.6.0-beta1

#2 - 2011-07-14 11:17 - Ernesto Baschny

I am not sure what the usecase would be to have a "default" hashing algorithm that can be configured. That would mean that upon change of configuration all already created hashes are invalid.

I would prefer to leave the decision on the hash to the one that is actually using the "hash" property in stdWrap, so that it's more explicit.

What do you think? What are the use-cases of having it configurable?

#3 - 2011-07-14 11:23 - Stefan Neufeind

I agree that encryption should just "fail" if a certain crypto is not available. Neither is leaving it unencrypted nor having it encrypted differently (just for the sake of enrypting :-)) much useful.
So I'd say return FALSE would be a good option. Together with stdWrap I guess that would result in an "empty" value then.

#4 - 2011-07-14 11:26 - Xavier Perseguers

How can we have a PHP error? I covered this case with a unit test and we have an explicit test to see if $algorithm is available.

The current behavior is to return the text as-this if it could not be hashed because of an invalid, empty or non-existing hashing algorithm. I'm not sure it really makes sense to automatically fall back to another hashing algorithm because in the end, you certainly have to know the hashing algorithm being used and automatic fallback, even if explicitly defined by the administrator really sounds odd to me...

#5 - 2011-07-14 11:28 - Xavier Perseguers

Stefan Neufeind wrote:

> *I agree that encryption should just "fail" if a certain crypto is not available. Neither is leaving it unencrypted nor having it encrypted differently (just for the sake of enrypting :-)) much useful.*
> *So I'd say return FALSE would be a good option. Together with stdWrap I guess that would result in an "empty" value then.*

I like the idea of not returning it unhashed, would help much in stdWrap conditions

#6 - 2011-07-14 11:33 - Stefan Neufeind

Using an unknown algo gives a PHP-warning. I guess that's what was meant to be suppressed by checking first. Imho fetching the list of available algos (maybe storing it via a singleton - not sure if that's really needed) would be all it takes.

Singleton for the list? Or just uncached access to the hash_algos()?

```
$ cat hash.php
  echo hash('myalgo', 'abc123');
```

```
$ php hash.php
PHP Warning:  hash(): Unknown hashing algorithm: myalgo in /home/sn/asd.php on line 2
```

#7 - 2011-07-14 11:49 - Xavier Perseguers

Stefan Neufeind wrote:

> *Using an unknown algo gives a PHP-warning. I guess that's what was meant to be suppressed by checking first. Imho fetching the list of available algos (maybe storing it via a singleton - not sure if that's really needed) would be all it takes.*

> *Singleton for the list? Or just uncached access to the hash_algos()?*

Before coming up with a cache for hash_algos() I'd like to see numbers showing that it is slow call worth some caching mechanism.

> *$ cat hash.php*
> *echo hash('myalgo', 'abc123');*
>
> *$ php hash.php*
> *PHP Warning: hash(): Unknown hashing algorithm: myalgo in /home/sn/asd.php on line 2*

Sorry but I still don't understand. We are explicitly checking that both the 'hash' function and the requested algorithm are available so are you really sure we get a PHP warning anyway? Does not make much sense.

Of course your example has this pitfall but our stdWrap implementation should not.

#8 - 2011-07-14 11:53 - Stefan Neufeind

Hmm, by the subject "add check" I assumed there was no check yet. I just reviewed the diff - of course you are right. And I think it's fine as is. Maybe we should make it return nothing/false if the algo is unavailable (avoiding disclosing unhashed information). But the rest is fine imho.

#9 - 2011-07-14 11:57 - Xavier Perseguers

I'd prefer "nothing" instead of FALSE because then we would have either a boolean or a string as return value, which I don't like. So OK for a follow-up to return '' but make sure to fix the unit test as well.

#10 - 2011-07-14 12:12 - Mr. Hudson

Patch set 1 of change Id25b85de039797aa7e39225fb0e2b1c75207a505 has been pushed to the review server.
It is available at http://review.typo3.org/3324

#11 - 2011-07-14 12:15 - Stefan Neufeind

Patch submitted for not disclosing (potentially sensitive) information. Somebody might want to update this issues headline, please.

#12 - 2011-07-14 12:21 - Xavier Perseguers

- Subject changed from Add check for valid hashing algorithms in stdWrap to stdWrap.hash: Return nothing for non-existing algorithm

#13 - 2011-07-14 12:22 - Mr. Hudson

Patch set 2 of change Id25b85de039797aa7e39225fb0e2b1c75207a505 has been pushed to the review server.
It is available at http://review.typo3.org/3324

#14 - 2011-07-14 21:30 - Stefan Neufeind

- Status changed from New to Resolved

- % Done changed from 0 to 100

Applied in changeset commit:9c255a3841db196e4a3585426306142eedd2ff64.

#15 - 2012-03-07 13:34 - Xavier Perseguers

- Status changed from Resolved to Closed

#16 - 2013-06-01 14:45 - Ernesto Baschny

- Target version deleted (4.6.0-beta1)