

## TYPO3.Flow - Major Feature # 33258

<b>Status:</b>	Accepted	<b>Priority:</b>	Should have
<b>Author:</b>	Sebastian Kurfuerst	<b>Category:</b>	
<b>Created:</b>	2012-01-17	<b>Assigned To:</b>	
<b>Updated:</b>	2014-02-24	<b>Due date:</b>	
<b>PHP Version:</b>			
<b>Has patch:</b>	No		
<b>Complexity:</b>			
<b>Subject:</b>	Implement support for Assetic		
<b>Description</b>			
We still need a clean solution to handle assets; and <a href="https://github.com/kriswallsmith/assetic">https://github.com/kriswallsmith/assetic</a> would be a great fit to integrate.			
First off, we need to create a concept for it.			
<b>Related issues:</b>			
duplicated by TYPO3 Flow Base Distribution - Story # 42407: Asset Management		<b>New</b>	<b>2012-10-26</b>

### History

#### #1 - 2012-02-07 14:11 - Sebastian Kurfuerst

Did anybody start working on a concept yet?

#### #2 - 2012-02-14 16:00 - Adrian Föder

So far, I've created two repositories at Github in order to get a clue and to get a possibility to work with and to point at it.

### Assetic Package

Located at <https://github.com/afoeder/Assetic-Package>, serves as FLOW3 package wrapper for the original repository. Provides a tiny script that re-pulls the necessary files from the original repository.

### TYPO3.Assets Package

Located at <https://github.com/afoeder/TYPO3.Asset>, provides the actual logic for working with Assetic.

For the time being I'll continue at a Wiki: [http://forge.typo3.org/projects/flow3/wiki/Assetic\\_ideas](http://forge.typo3.org/projects/flow3/wiki/Assetic_ideas) page, for any reason I currently think that's the better way.

#### #3 - 2012-02-14 22:55 - Johannes K

Great!

Do you plan to integrate filters and nested assets, too?

It would be nice to be able to set the debug mode ( <https://github.com/kriswallsmith/assetic/blob/master/docs/en/define.md> ) from the configuration file. So we could disable merging/filters etc... in Development Context and have it enabled in Production.

#### #4 - 2012-02-15 09:48 - Adrian Föder

Johannes,

we "plan" anything you like :) We're currently in the absolut early stage of collecting some ideas and implementation possibilities.

Do you have production experience with Assetic? Would be fantastic to have you on board for that!

**#5 - 2012-02-15 09:51 - Sebastian Kurfuerst**

- Status changed from New to Accepted
- Assigned To set to Adrian Föder

**#6 - 2012-02-16 07:56 - Sebastian Kurfuerst**

Hey Adrian,

Thanks for starting with a concept! Some comments from my side:

- I'd suggest to use "Assets.yaml" instead of "Settings.yaml"
- The ViewHelpers IMHO should **directly** output the `<script src>` and `<link rel="stylesheet">` tags, and not work like `<f:for>`. Any particular reason you started with it?
- The bundle names should be namespaced (i.e. **TYPO3.TYPO3:Debug** instead of **Debug**)
- I'd rename "references" to "dependencies" and only allow bundle names there.

I am not yet sure if we actually need individual asset definitions, or if we can **ALWAYS** enforce the user to use Bundles. This would IMHO simplify the configuration.

What do you think?

Greetings, Sebastian

**#7 - 2012-03-05 08:53 - Adrian Föder**

Hi Sebastian,

indeed I thought about it in "idle time" ;-), I'll write down a few things.

Basically, the design of my suggestion was a bit inspired by already-existing implementations for e.g. Symfony or even Smarty.

Another general question I'd have is the level of abstraction we want to have here; how much should the Assetic usage bound to FLOW3? This would mean; how **generic** should the configuration, for example, be. This would be important for the layout of the Settings/Assets.yaml file; or even the naming of it. I, btw, would be fine to make Assetic compulsory per convention; if a user would like to use other stuff he's free to implement it project-based.

So far, to your statements:

1. Assets.yaml is fine for me, as mentioned above.
2. Direct output of `<script>` / `<link rel="stylesheet">`: The drawback is that the user hasn't control over the exact looking of the tags anymore. Maybe he'd like to add additional params or lack the auto-closing for any reason. But I must admit that that would be very rare and that most use-cases can be beaten with using the AbstractTag VH.

Second reason for the nested layout was to make the actual VH less bloated; but that's only important when letting the user define the concrete tag.

3. Namespacing: of course agree.
4. Forcing bundles: I think I'm also fine with it, because as you said: we won't allow such much permutations of configuration possibilities then.

As one of the next steps, I'd love to see a "target" specified, i.e. a use-case with some (JS) files and the appropriate Assets.yaml and VH looking. Because I personally still haven't the whole picture of what's possible with Assetic, that would be a good next step for me.

**#8 - 2013-04-11 08:20 - Adrian Förder**

- *Assigned To deleted (Adrian Förder)*