

## TYPO3.Neos - Feature # 33612

<b>Status:</b>	Resolved	<b>Priority:</b>	Must have
<b>Author:</b>	Sebastian Kurfuerst	<b>Category:</b>	
<b>Created:</b>	2012-02-02	<b>Assigned To:</b>	
<b>Updated:</b>	2012-07-25	<b>Due date:</b>	
<b>Subject:</b>	Create "Initial Setup" wizard		
<b>Description</b>			
<p>Create a wizard for the initial setup of TYPO3 Phoenix, where one can enter DB credentials etc</p> <p>Ideas for direction:</p> <ol style="list-style-type: none"><li>1. setup.php in the Web folder -&gt; check if the TYPO3.Setup bootstrap file is available<ul style="list-style-type: none"><li>- TYPO3.Setup is a separate package since continues deployment won't need this feature</li><li>- Check if the package TYPO3.Setup is activated?</li></ul></li><li>2. Run the TYPO3.Setup bootstrap<ul style="list-style-type: none"><li>- Check for minimum requirements? Most are duplicated in the FLOW3 core bootstrap (could we combine this somehow)?</li><li>- We should have some kind of authentication (Maybe a file should be CREATED to run the setup wizard)<ul style="list-style-type: none"><li>- A file could be created automatically if it's the first time FLOW3 is executed (also a file somewhere) with the login information</li></ul></li><li>- Check file permissions?</li></ul></li><li>3. If everything is okay (FLOW3 is running):<ul style="list-style-type: none"><li>- Run the FLOW3 core bootstrap in Setup context through the already executed code and then create a request handler which mimics the web request handler, although the request builder doesn't use routing for fetching the correct setup controller<ul style="list-style-type: none"><li>- Authenticate with login field (FLOW3 security)</li></ul></li></ul></li><li>4. A step setup wizard<ul style="list-style-type: none"><li>- Steps configured in yaml</li><li>- Handle a variable number of steps</li><li>- Use form step stuff from the form project</li></ul></li></ol> <ul style="list-style-type: none"><li>- Call commands through the \TYPO3\FLOW3\Core\Booting\Scripts executeCommand method</li><li>- Would be nice if the setup could create the database if provided with user/pass with access for that<ul style="list-style-type: none"><li>- <a href="http://www.doctrine-project.org/documentation/manual/1_0/nl/connections:creating-and-dropping-database">http://www.doctrine-project.org/documentation/manual/1_0/nl/connections:creating-and-dropping-database</a></li></ul></li><li>- Steps (must be able to go back &amp; forth)<ol style="list-style-type: none"><li>1. Language</li><li>2. Pre-installation check (requirement meets)</li><li>3. Database setup (type, name, user, pass)<ul style="list-style-type: none"><li>- Advanced options (host, port)</li><li>- Backup of old db?</li></ul></li><li>4. Title, Administrator (username, password, e-mail), default language/timezone? (TYPO3 specific)</li><li>5. Import site (TYPO3 specific)</li></ol></li></ul> <p>User scenarios:</p> <ul style="list-style-type: none"><li>- Cloud server</li><li>- Shared hosting server</li><li>- No database created (but access to create db's)</li><li>- Full access</li></ul> <p>Open questions/suggestions:</p> <ul style="list-style-type: none"><li>- Create a configuration tool like the v4 install tool using the same request handler for TYPO3 / FLOW3</li><li>- Create a install script suggested by Jacob Floyd, which would fetch the packages through or w/o git (if available)<ul style="list-style-type: none"><li>- Then after this process has finished it should redirect to the setup wizard</li><li>- More about it here: <a href="https://docs.google.com/document/d/136atxn0pNVR04uD5b_I57rwdfxwnUImRFmF0rD07ut4/edit">https://docs.google.com/document/d/136atxn0pNVR04uD5b_I57rwdfxwnUImRFmF0rD07ut4/edit</a></li></ul></li></ul>			
<b>Related issues:</b>			
related to TYPO3.Neos - Feature # 39240: Setup: Final step		<b>Needs Feedback</b>	2012-07-25

related to TYPO3.Neos - Feature # 39239: Install script	<b>New</b>	<b>2012-07-25</b>
related to TYPO3.Neos - Feature # 39238: CLI setup	<b>New</b>	<b>2012-07-25</b>
related to TYPO3.Neos - Feature # 39236: Setup step features	<b>New</b>	<b>2012-07-25</b>

## Associated revisions

---

### Revision 25deba8c - 2012-05-09 23:33 - Aske Ertmann

[FEATURE] Administrator account step for TYPO3.Setup wizard

Change-Id: I3def7d2be0eb89e481d8965f86946c936bc2d14c  
Related: #33612

### Revision 4853097c - 2012-05-09 23:33 - Aske Ertmann

[FEATURE] Site import step for TYPO3.Setup wizard

Change-Id: Ida75e74bf308c977406b7178e9ae095730d1ef21  
Related: #33612

### Revision e4d3787f - 2012-05-16 16:34 - Rens Admiraal

[TASK] Add TYPO3.Setup and requirements to site distribution

Change-Id: I04e71c58b3eee0733c02f1ca5a49509c75412e0b  
Related: #33612

## History

---

### #1 - 2012-02-09 14:57 - Gerrit Code Review

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

It is available at <http://review.typo3.org/8931>

### #2 - 2012-02-09 21:12 - Jacob Floyd

I've added to that google doc about the pre-setup install script (that downloads/initializes a flow3/typo3 phoenix distribution): Make the install script so that it can be automated from the command line, as well as running in the browser. CLI for more cloud environments especially corporate environments. Web install for shared hosts, and mom/pop shops.

In other words: make the installation friendly to enterprise customers as well as small businesses and even personal websites. If people can use it at home, they're more likely to learn to love it and start a ground swell of support from within the corporate environment. That's one reason WP is so popular - lots of people use it at home for personal sites, and then they recommend using WP when it comes time to build a site in their org.

I think an ideal basic installation should take about 5 minutes. That includes the initial setup wizard. So, people should be able to get a basic, mostly extension-less Typo3 Phoenix installation, running with a barebones functional website, in about 5-10 minutes. (Some fudge room could be acceptable for bandwidth considerations or download time - so, that's 5-10 minutes of active effort on the user's part.)

Make the install and setup process *fast*, so that the first impression of Typo3 is, "Wow, I like this", or "Wow, that was fast, so my website will be fast!" (first impressions aren't always logical... But they color your view of the product for the rest of the evaluation process)

### #3 - 2012-03-09 15:17 - Bastian Waidelich

I agree to jacobs ideas!

Some more ideas after talking to Aske:

We need to keep the number of required steps as **low as possible**. So I suggested to have the language defaulting to English and have a language selector on each step. Also the "requirement check" should be skipped as soon as all **vital** checks pass - on the last page (post-setup) you could get a nice overview of possible next steps ("You're done, go to the [website], [advanced setup options], ... and there are [3 warnings & 5 recommendations]") warnings & recommendations could be a separate module (usable for FLOW3, too) that is equal to the System report in v4. For a slick Web interface we should make use of AJAX where possible (e.g. you enter the db credentials and the db selector gets active as soon as the connection was established successfully). The CLI based setup is important too, but could be postponed to a later sprint IMO.

Some technical ideas:

- The steps are configured via YAML and are extensible
- Each step is **one** Form (form factory or YAML based form definition)
- The controller knows the **context** and executes the required step
- Steps can define **requirements**. A requirement could be implemented by a simple PHP class. One example is a "DatabaseConnectionRequirement". If a requirement is not met the controller needs to jump to the previous step until all requirements are met (this needs some more thoughts)

### #4 - 2012-05-21 15:18 - Aske Ertmann

- % Done changed from 0 to 50

### #5 - 2012-05-21 15:18 - Aske Ertmann

- Status changed from Accepted to On Hold

### #6 - 2012-05-21 15:18 - Aske Ertmann

We have a working setup atm. I think we should postpone making it smoother..

### #7 - 2012-05-29 10:11 - Sebastian Kurfuerst

OK I just checked what needs to be done in order to test the setup:

1. checkout f.e. the Flow3Org Distribution
2. place <https://gist.github.com/2006620> in Web/setup.php
3. move TYPO3.Setup package from Packages/Application to Packages/Framework

Then, call setup.php and follow the instructions.

For me it fails with "( ! ) Fatal error: Class 'TYPO3\FLOW3\Utility\Files' not found in /Volumes/data/htdocs/Flow3Org/Packages/Framework/TYPO3.Setup/Classes/Core/Bootstrap.php on line 184"

This is totally reasonable as the autoloader is not yet active at this point....

How has this ever worked? :-)

Greets, Sebastian

---

after fixing the above issue by adding `require_once(__DIR__ . '/../..../TYPO3.FLOW3/Classes/Utility/Files.php');`

I needed to add a setup route:

```
-  
name: 'Setup'  
uriPattern: '<SetupSubroutes>'  
subRoutes:  
  SetupSubroutes:  
    package: TYPO3.Setup
```

- On the first hit of the setup tool, we see a password such as "8lsPbLSf" -- it needs to be copied and safely stored.
- The database setup screen is still a little weird -- after entering username and password there is no way to reload it in order to select the database
- When clicking submit and the migrations are executed etc, it would be great to show an intermediary screen telling the user what is happening. else it just takes long and there is no user feedback.
- error messages in setup should be styled with red background
- Message: "No site packages were available, make sure you have an active site package"
  - can be fixed by adding `<categories><category>Site</category></categories>` to `package.xml`

#### #8 - 2012-05-29 14:07 - Bastian Waidelich

Sebastian Kurfuerst wrote:

Thanks for testing & reporting.

1. checkout f.e. the *Flow3Org Distribution*
2. place <https://gist.github.com/2006620> in `Web/setup.php`
3. move `TYPO3.Setup` package from `Packages/Application` to `Packages/Framework`

I'm still not very happy with the additional "setup.php" because we have to add & maintain it in all distributions and it requires the user to call "/setup.php". Also the setup currently requires a special Route and we decided that a global, always working, fallback route can't be the solution.

I didn't join the Setup efforts lately (sorry for that) and you might have completely different plans, but what do you think of following approach:

- The core bootstrap gets a mechanism for the [very fundamental requirements](#) - those checks are only executed on the first hit (the bootstrap could create a "REQUIREMENTS\_CHECKED" file and check for its existence ,or similar)
- The setup package registers a RequestHandler that takes over when some argument is present (e.g. `?__typo3.setup[step]`). Redirecting to a setup step (for instance when no db is setup) would be as simple as a `UriBuilder::uriFor()` call. The performance hit due to the additional `isset()` check in the RequestHandler is neglectable IMO and also, the `TYPO3.Setup` package could deactivate itself after the last step.

Do you agree?

- *The database setup screen is still a little weird -- after entering username and password there is no way to reload it in order to select the database*

I'll work on a custom FormElement for this one asap!

#### #9 - 2012-05-29 14:45 - Bastian Waidelich

Bastian Waidelich wrote:

- *The core bootstrap gets a mechanism for the "very fundamental requirements" [...]*

BTW: This mechanism could also try to determine the "phpBinaryPathAndFilename" as Aske implemented it once: <https://gist.github.com/2358452>

**#10 - 2012-07-13 16:31 - Karsten Dambekalns**

- *Status changed from On Hold to Needs Feedback*

This has been on hold, but IMHO the setup feature is done. So, shouldn't this be split and/or some (sub-)tasks created? This is just too epic...

**#11 - 2012-07-13 16:43 - Aske Ertmann**

Agreed. This is simply too epic.

**#12 - 2012-07-25 11:59 - Aske Ertmann**

- *Status changed from Needs Feedback to Resolved*

- *Assigned To deleted (Aske Ertmann)*

- *% Done changed from 50 to 100*

Created the following tickets: #39240, #39239, #39238, #39236