

Site Building Guide

Including

- Introduction: Content at a glance.
- Translating Reports: Create reports in multiple languages.
- Feeds Importing: Create nodes using data blocks and/or reports.
- Bean Integration: Create Drupal blocks out of reports.
- Views Row Style: Create more complicated layouts than can typically be created using views.
- WYSIWYG Editors: Enhance the report editing experience for users not familiar with (x)HTML.

Introduction

This guide documents how to further enhance a Forena implementation by taking advantage of various integrations of Forena with other Drupal modules that further extend Forena's capabilities, such as:

- Use the locale module included in Drupal to enable translation of reports in multiple languages.
- Use the Feeds module to create nodes using Forena data blocks and/or reports.
- Use the Bean module to create drupal blocks out of Forena reports.
- Use the Views module with Forena to create additional more complicated layouts than can typically be created using views. Forena provides a views rowstyle plugin that lets you use a forena report to format the output of a view.
- Use the Masquerade module to review (experiment with) all sorts of Forena related authorizations.
- Use a **WYSIWYG editor** to edit the body of a report, to avoid editing the specifications of the report body using XHTML tags.

Translating Reports

Forena includes support for report translations in multiple languages. To enable this support simply enable module locale which is included in Drupal core. Unlike other approaches at localization, because Forena allows full HTML markup, each report must be translated into the target language.

Feeds Importing

Using the Forena integration with the Feeds module it is possible to import data blocks and reports as Drupal nodes or Drupal users).

For a tutorial about the available features check out the video about [How to import data and reports based on data as Drupal nodes](#).

Bean Integration

For anyone not familiar yet with the Bean module, here is a quote from the bean project page: **Bean** is an acronym that stands for **B**lock **E**ntities **A**ren't **N**odes. Drupal blocks placed in regions are currently not entities, but the BEAN module creates block entities that can be placed in regions of your theme.

By using the Forena integration with the Bean module, it is possible to create Drupal blocks that display Forena reports. An instance of a particular report with a particular set of parameters can be created as a block using the bean module.

Notes:

- This integration was implemented without creating a Bean module dependency in Forena. If the Bean module is installed and enabled, the integration can be used.
- Don't confuse Drupal blocks with Forena's data blocks. There are different concepts.

Example

As an illustration of how to use this feature, let's create a block showing a condensed version of the Documentation Table Of Contents report:

Enter these options on this Block / Add / Forena page:

Label	Forena Docs
Title	Reporting Documentation

Select a report (within Settings) help/toc

URL Parameters (within Settings) show_chapters=0

Forena report was the only option available block types (i.e. no other Drupal module is in use that also integrates with the Bean module), then the Add Block link will take you straight to the Block / Add / Forena page Otherwise refine your selection until you get to this page.

save button.

- 2.
- 3.
- 4.
- 5.

Views Integration

Using the Forena integration with the Views module, it is possible to use Forena reports as a views row style, so that in a view you can use the Forena templating engine to do the formatting. Although this can be done through the views templating engine Forena users may find they have more control with this engine.

WYSIWYG Editors

Forena's WYSIWYG reporting user interface facilitates the editing of the **body** of a report using a WYSIWYG editor (so that you are not assumed to be familiar with HTML tags to understand or edit the specifications of the report body). But because of the custom (FRX) attributes, some challenges need to be addressed:

- Forena will (a) save away custom attributes in the report body before presenting it for editing via a WYSIWYG editor and (b) merge them back in after editing ended.
- Forena adds back the special tags that were stripped based on id attributes, but it also makes sure that there is an id on every HTML tag that has an FRX attribute.

Some recommendations to prevent issues when attempting to enable a WYSIWYG editor for editing the body of a report:

- Complete your homework first regarding Text formats configuration! Doing so will prevent a lot of trouble with just any WYSIWYG editor.
- Be careful in selecting a WYSIWYG editor so that it doesn't conflict with the FRX attributes. At a minimum id attributes cannot be stripped by the editor. Forena makes an attempt to read any custom attributes based on html id attributes, but if these are stripped, all custom frx:attributes can cause problems.

Some more details about available WYSIWYG editors for editing the body of a report:

- **TinyMCE** editor works provided when configured with these options:

- Verify HTML.
- Remove Line Breaks.
- Convert Font Tags to Styles.

Warning: TincyMCE will strip out SVG graphs from any report it is used to edit.

- It should work with other WYSIWYG HTML editors, however CKEditor seems to cause problems as it strips id attributes by default. You will need to set the allowed Content option on CKEditor to make sure it does not strip them.

If you have experience with other editors that you'd recommend to include here, or you noticed anything that's incomplete, please report about it in the Forena Issue Queue.