Dispatcher quickstart

Table of contents

1 Intended Audience

People who are helping to develop the Dispatcher.

> Warning:
>
> The "Dispatcher" (previously known as "Views") is new functionality which is still in development phase. That is why it is in the "whiteboard" section of Forrest. This document will also need to change to keep pace. We are working at the moment on moving this plugin from the whiteboard into the core plugins. See Status of Themes: Skins and Dispatcher.

2 Purpose

This document will get you started. We will Dispatcher-enable an existing site and show how to add/remove ready-made contacts. Then we will discuss how add your own new contracts.

This document encourages developers to get involved with fine-tuning and testing the dispatcher. Please help to enhance the current core contracts so that people do not need to re-invent the wheel.

3 Prerequisites

- Using Forrest trunk of SVN (i.e. head of development).
- Followed the installation instructions below.
- You have an existing forrest site and want to try the new Dispatcher. Otherwise create a 'forrest seed-sample' site.

4 Steps

> Note:
>
> When developing with the dispatcher we assume you are using 'forrest run' and the following workflow "change files -> refresh browser".
> Installing a mozilla browser and the Forrestbar helps a lot with developing, but is not necessary.

4.1 Get ready

- Do 'svn update' on forrest/trunk
- Do the 'cd main; build clean; build' (necessary because the Dispatcher is changing rapidly and uses some Java)

4.2 Dispatcher-enable the existing site

- Add the new plugins to forrest.properties ... ,org.apache.forrest.plugin.internal.dispatcher,org.apache.forrest.themes.core
- localhost:8888/index.html ... fantastic. See the pelt view.

4.3 Use another theme

- Add <property name="dispatcher.theme" value="pelt"/> to your forrest.properties.xml
- Re-start 'forrest run'
- localhost:8888/index.html ... See the new view.

4.4 Create our own structurer by copy-and-customise

Copy the default structurer for the pelt theme and make local changes.

> Note:
>
> At this stage of rapid development of the Dispatcher, be sure to keep your copy synchronised. Use 'diff' against the known svn version of the core pelt.fv to track your local changes. Also please consider contributing new contracts and changes to the default structurers to the Forrest projects. That eases your local version management and everyone will benefit.

> Note:
>
> We use `${themer.project.dir}` for PROJECT_HOME/src/documentation/resources/themes (create the new directory first). You can change this location by adding <match pattern="themer.project.dir"> <location src="{properties:resources}/themes" /> </match> to your locationmap and point to another directory.

> Note:
>
> By default THEMER_PLUGIN can be located at FORREST_HOME/whiteboard/plugins/org.apache.forrest.themes.core

- Copy THEMER_PLUGIN/themes/pelt.fv into your project at ${themer.project.dir}/pelt.fv
- Copy THEMER_PLUGIN/themes/pelt/panels/pelt-html.header.panel.xml into your project at ${themer.project.dir}/pelt/panels/pelt-html.header.panel.xml
- Re-start 'forrest run'
- localhost:8888/index.html ... See the same view, now structured by us.
- In localhost:8888/resolve.structurer.index you can find a copy of your current structurer. Any change made to your pelt.fv file will be added here.

From here on there is no need to re-start 'forrest run'. Just edit the structurer and see the effect.

Warning:

Right now the dispatcher is heavely cached, so maybe in the current version it should be necessary to reboot the system in order to see the changes. To avoid temporally this problem, turn off the cache system changing: <forrest:views xmlns:forrest="http://apache.org/forrest/templates/1.0" xmlns:jx="http://apache.org/cocoon/templates/jx/1.0" jx:cache-key="#{$cocoon/parameters/getRequest}" jx:cache-validity="${Packages.org.apache.excalibur.source.impl.validity.NOPValidity()}"> to <forrest:views xmlns:forrest="http://apache.org/forrest/templates/1.0" xmlns:jx="http://apache.org/cocoon/templates/jx/1.0"> in your THEMER_PLUGIN/themes/pelt.fv file.

4.5 Remove a default contract

- Remove the top breadcrumb trail. Edit your pelt-html.header.panel.xml, find the "branding-breadcrumbs" contract and comment it out.

4.6 Add a default contract

Forrest provides many default contracts. If you are using the Forrestbar then choose "dispatcher-devs => ls.contracts". Otherwise visit localhost:8888/ls.contracts.html

- See the usage detail for your chosen contract via the above mentioned list. e.g. "siteinfo-current-time"
- Insert it at the appropriate point in your structurer. Note that it will need to go inside the appropriate CSS hook, e.g.

```
...
<forrest:hook name="footer">
  <forrest:contract name="siteinfo-current-time"/>
  ...
```

Notice that you did not need to copy any other code to your project space. Forrest finds the default contract in its core (the themes.core plugin).

You will find some commonly used contracts in place but commented out in both the common and the pelt structurer files. Feel free to uncomment them to activate them, move them around into other hooks or create new hooks for them.

Note:

Work is currently under way for implementing Panels (was tiles). Currently we now have a 'panels' sub-dir for each 'theme'. This panels sub-dir currently contains a main html panel '%themename%-html.panel.xml' and a CSS panel '%themename%-css.panel.xml'. So you can change a collection of HTML and CSS outputs from these two XML files.

4.7 Add a new project contract

> Warning:
>
> Carefully consider the purpose of your contracts. If they are useful in a wider context, then they belong in plugins or the core of Forrest. See below for further discussion on this important topic.

Project-based contracts are defined in theme-specific and output-format-specific directory structure, e.g. ${themer.project.dir}/THEME_NAME/OUTPUT_FORMAT/

Project-based contracts common to all themes go in ${themer.project.dir}/common/OUTPUT_FORMAT/

To get started quickly (it is not a project-based contract but a demo), copy one of the default contracts. e.g. copy siteinfo-current-time.ft to become siteinfo-doodad.ft ...

- Copy THEMER_PLUGIN/themes/common/html/siteinfo-current-time.ft into your project at ${themer.project.dir}/common/html/siteinfo-doodad.ft (create the new directory first).
- Edit it to suit. Replace all occurrences of "siteinfo-current-time" with "siteinfo-doodad" and make your other changes (e.g. the text and the javascript function).
- Declare your new contract in your structurer as done above for adding a default contract.

See a list of your project-based contracts and their usage notes via Forrestbar "dispatcher-devs => ls.contracts.project". Otherwise visit localhost:8888/ls.contracts.project.html

4.8 Decide how to manage your contracts

Depending on the use of a new contract you can place it in different locations. General use contracts should be placed in the THEME_PLUGIN directory. Contracts only suitable for one theme should be stored in the specific theme directory, that is, `THEMER_PLUGIN/resources/themes/THEME_NAME/OUTPUT_FORMAT`. Otherwise, common contracts should be place in the common folder: `THEMER_PLUGIN/resources/themes/common/OUTPUT_FORMAT`.

If the contract is specific to a particular purpose, that is a particular plugin, it should be included with the plugin itself. For example, the "employment history" contract is specific to the resume plugin so it should be placed there. The correct location for contrats with a particular purpose is `PLUGIN_NAME/resources/themes/THEME_NAME/OUTPUT_FORMAT`.

There is another category for contracts, those that are both specific to a particular purpose and defined for a specific site. For example, a "process order" contract that integrates with an in-house order management system. In this case, another location is more suitable. However, before proceed think again about the uniqueness of the contract. Most contracts can be generalised to be useful in more than one environment and so one of the above locations can be used. If you are still sure that this is a site-specific contract then place it in `PROJECT_HOME/src/documentation/resources/themes/common`.

If you develop a new contract, please provide a patch via our issue tracker so that we can include it in future releases of Forrest.

5 Further Reading

Congratulations you are now able to work with the Dispatcher. From here we recommend to read the following How-Tos:

- How to use the structurer
- Create your own contract implementation

6 Feedback

Please provide feedback about this document via the mailing lists.