

J2-page-manager-profiling

J2 Page Manager Profiling

The Page Manager Profiling facilities are built upon the standard Profiler and Profiler Locator components. The Page Manager implementation outlined here resides in the Page Manager component and is file system based. The profiling capabilities of the Page Manager can be turned off using the appropriate component options in the Spring Framework configuration. Doing so will turn off the features documented here and reduces the Page Manager to simply returning various content directly accessed by portal request paths. Using the Profiler to select pages and navigational elements is the default configuration for the Page Manager.

The Profiler is configured for within the Page Manager by setting up profiling rules associated with a user principal. By default, J2 comes with several predefined rules including “j1” and “role-group” used in examples below. Custom rules can easily be defined and assigned to users using the appropriate administration portlets. All Profiler Rules are assigned to users by name; these mapping names are used to select which rules are used while profiling portal pages.

Page Manager Profiling Rules and Profiler Locators

Pages and the navigation context in which they belong are selected by the Page Manager per portal request. The Page Manager is passed the complete set of Profile Locators generated from each Profiling Rule for the user and the portal request by the Profiler. The Profile Locators are passed to the Page Manager by the Profiler Valve in the J2 request processing pipeline. This set of Profile Locators completely describes the request for the Page Manager and the set is used to generate keys for cached page contexts. The Page Manager does not interact with the Profiler directly: it only consumes the Profile Locators passed to it.

The Page Manager requires that at least one named Profile Locator is constructed from the user Profiler Rules; this locator must be mapped to the name “page”. As one would expect, this is the locator rule used to select the page returned by the Page Manager and to populate the Profiled Page Context filled with page navigation elements.

Page Manager Page Search Paths

Each Profile Locator is mapped by the Page Manager into a series of search paths that assume a directory tree structure unique to each Profiling Rule. The mapping is made by converting the name value properties within the locators to directory names. All search paths are rooted at the WEB-INF pages directory. For example, here is a typical “j1” Profiling Rule locator:

```
path:/:user:fernand:mediatype:html:language:fr:country:FR
```

The Page Manager maps these properties to directories by prefixing the Profile Locator Property name with a '_' character and folding values to lower case. The special property names “page” and “path” modify or replace the request path and do not otherwise appear as part of the search path. Assuming a request path “/search/search-engine.psml” is specified or generated by the “page” or “path” properties, the following ordered search paths are used to select the profiled page using the previous example:

```
WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/_country/fr/search/search-engine.psml
```

```
WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/search/search-engine.psml
```

```
WEB-INF/pages/_user/fernand/_mediatype/html/search/search-engine.psml
```

```
WEB-INF/pages/_user/fernand/search/search-engine.psml
```

```
WEB-INF/pages/search/search-engine.psml
```

The first page that is found to match is selected and the folder that it resides in is used to construct the Profiled Page Context that specifies all site navigation elements.

In the event the page is not found using these search paths, a fall back search is made assuming a “/” request path using the same Profile Locator. For our example, this would result in searching for default pages for the following folder paths:

```
WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/_country/fr/
```

```
WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/
```

```
WEB-INF/pages/_user/fernand/_mediatype/html/
```

```
WEB-INF/pages/_user/fernand/
```

J2-page-manager-profiling

WEB-INF/pages/

Locating Folder Default Pages

Fall back and explicit folder request path pages are selected by the Page Manager by looking for default pages along the Profiler Locator generated page search paths. A folder request path is defined as any path not ending in the page extension, (i.e. “.psml”), or a request for any page named “default-page.psml”, (see explanation below). Using our example from above, let's assume that there were no matches for the “/search/search-engine.psml” request path. This would invoke the “/” Page Manager fall back search. If the following page existed:

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/default-page.psml

it would be returned, (assuming that there were no default pages specified or available before this path in search order). In addition to creating pages named “default-page.psml”, default pages can also be specified in folder meta data. The Page Manager does not distinguish between either of these techniques of specifying a folder default page.

As a special case, aggressive default page resolution is used to determine a matching page when only one search path directory actually exists and/or when searching in the “/” directory as a last resort. In these cases, the first page found in the folder is selected if there is no page specified as the default and the default-page.psml file does not exist.

Finally, explicit portal requests for any page named default-page.psml is assumed to be a folder request and not a request for a particular page. This allows the folder/default page logic to be run for these requests. The “default-page.psml” page specification is always stripped by the Page Manager for page requests.

Alternate Profiler Locator Page and Folder Selection

If there is more than one Profiler Locator passed to the Page Manager and the fall back search for a page match using the “/” path is successful, additional page profiling will be performed to extend page and folder selection. Only the page or folder default page is searched for: the successful fall back profiling result is still used to populate the page context navigation elements.

Using our previous example, the “/search/search-engine.psml” request path could be profiled against other specified Profiler Locators assuming the fall back search successfully matched

a default page. Assume the following alternate locator derived from the “role-group” Profiling Rule was supplied to the Page Manager:

role:user:group:user:group:admin

Again an ordered search paths are generated by the Page Manager, (note here the multi-valued group profiler property locator and how it is interpreted by the Page Manager):

WEB-INF/pages/_role/user/_group/user/search/search-engine.psml

WEB-INF/pages/_role/user/_group/admin/search/search-engine.psml

WEB-INF/pages/_role/user/search/search-engine.psml

WEB-INF/pages/search/search-engine.psml

If the request path is exactly matched using any alternate profiler locators, the matched page or folder becomes the selected page for the portal request. If multiple alternate locators are supplied to the Page Manager, all are searched for a match. No control is given over the sequencing of the alternate locators: folder and/page name paths should be unique to avoid unpredictable content selection.

Profiled Meta Data and Navigation Elements

The Profiled Page Context is populated by the Page Manager based on the profiled page and its folder selected using the various Profiler Locators. This context is made up of the following:

- page - the selected page,
- folder - the folder of the selected page,
- sibling pages - aggregated pages that belong to the same request path folder,
- parent folder - the parent of the selected page folder,
- sibling folders - aggregated folders that belong to the same request path folder,
- root links - aggregated links defined in the root request path, and
- document sets - aggregated set of named document lists relative to the request path.

Without delving into too much detail, the default “jetspeed” layout decorator renders these into the following navigational controls found on each portal page: sibling pages are mapped to tabs, the parent folder is the target of the “back” link, sibling folders are manifest as folder links, and both root links and each document set are represented as collections of links in the

J2-page-manager-profiling

navigational menu area. The Page Manager itself makes no assumption on how these elements are rendered, but does guarantee that the urls associated with each navigation element are stripped of and locator search path artifacts.

In addition to these navigational elements represented by folders and content files, document order is also profiled by the Page Manager to provide control over profiled content and resulting navigation elements. It selects the most specific document order defined in the folder meta data and applies it to all collections returned in the Profiled Page Context.

General Profiling Aggregation Rules

The Page Manager defines a set of rules it uses to aggregate the various navigation elements using the Profiler Locator determined search paths. As mentioned above, navigation elements are always determined using the “page” Profiler Locator with either the specified request or fall back paths.

The “page” ordered search paths are processed in order as is done to locate the profiled page and folder, but any page specification is stripped from the path. In addition, some navigation elements “inherit” setting from parent request folders. This extended aggregation is limited and does not extend into or through locator defined directories. Here are examples of the page and inherited ordered search paths used to aggregate elements for the examples above:

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/_country/fr/search

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/search

WEB-INF/pages/_user/fernand/_mediatype/html/search

WEB-INF/pages/_user/fernand/search

WEB-INF/pages/search

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/_country/fr/search

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/_country/fr

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr/search

WEB-INF/pages/_user/fernand/_mediatype/html/_language/fr

WEB-INF/pages/_user/fernand/_mediatype/html/search

WEB-INF/pages/_user/fernand/_mediatype/html

WEB-INF/pages/_user/fernand/search

WEB-INF/pages/_user/fernand

WEB-INF/pages/search

WEB-INF/pages

Aggregation also filters duplicate elements with the same file name as it proceeds through the search paths. This implies that the first occurrence of any element with a given file name is selected while subsequent versions of the same file are ignored. This same rule is used to match folders and pages and is what gives the search path order its significance.

Profiling Aggregation Rules Table

Here is a list of the profiling aggregation rules used to determine the Profiled Page Context members and document order meta data:

Aggregated Information	Search Path Type	Notes
sibling pages	page	Sibling pages are aggregated using the portal page request path folder.
sibling folders	page	Sibling folders are aggregated using the portal page request path folder.
root links	page	Root links are always aggregated using the root request path, "/"; the portal page request path is ignored.
document sets	inherited	Document sets are aggregated using the portal page request path folder and are filtered if the profiling rule name specified in the document set is not available as a corresponding profiler locator.
document order	inherited	Document order for pages, folders, and document sets is determined by the first folder

		with meta data specifying document ordering. The folder is located using the portal page request path folder search paths. Note that the order of root links and document set members is determined by the document order found for the root request path, "/", using the appropriate profiling rule for document sets, (see below).
--	--	--

Table 1: Table1

Controlling Navigation Element Ordering

Specifying the order that decorator elements should render navigational controls in the page is done using the document order specification in the folder meta data files. The order of the sibling folder links, sibling page tabs, and decorator set menu groups is determined by the Page Manager using the first folder found with ordering information using the inherited ordered search paths, (see above). Documents are ordered by name and any documents that are not explicitly ordered will be collated by default and appear after the explicitly ordered set. Because document ordering within a single folder meta data file is applied to all aggregated pages, folders, and document sets, it will commonly specify the ordering of documents that are not children of the folder itself.

The ordering of root links and document set members is similarly determined by locating folder meta data orderings. However, the portal page request path is ignored and the ordering meta data is generally located using only the root, "/", request path. As a special case, document set members can be ordered using the member's folder meta data file if only one wild card document path is specified. These approaches limit the amount of ordering information to be maintained throughout the various folder meta data documents. Using the root request path by default is also a natural simplification since root links and document set members are always specified relative to the root request path.

Profiling Document Set Members

Document sets support the specification of an alternate Profiling Rule to be used by the Page Manager when binding their document paths to concrete document set members for the user. By default, the special "docset" or standard "page" Profiler Locators are used. However, any named Profile Locator supplied to the Page Manager can be specified. A document set that

specifies a profiling rule name that does not match an available profiler locator name is ignored.

As mentioned above, ordering of document set members is determined by folder meta data located within the alternate locator profiled root request or single wild card document path search directories.

When explicit or folder default page references are generated using alternate Profiler Locators, they are made available only as pages to the portal with normal urls. Since no special state or Page Manager directives are encoded in the urls, the usual request path processing is done on the document set member page requests. Page requests paths generated by an alternate Profile Locator can be overridden by content in the “page” locator search paths, (see above for “page” vs. alternate locator profiling details). Consequently, care must be taken to use unique request page names and paths when defining content referenced by document sets that should not be customized or replaced.

Alternate Profile Locators are not used to determine navigation elements passed to the decorators. The fall back root request path is used for navigations by default when an alternate Profile Locator is used to select a portal page. While this insures a consistent navigation context, it also implies that document set referenced pages and folders may not appear as siblings within the Profiled Page Context.

Profiled Page Context Caching

Page Manager profiling results are currently cached using the portal request path and all Profiler Locators in a composite key. The goal of this cache is to provide rapid J2 pipeline execution for repeated requests. The maximum cache size defaults to 100 Profiled Page Context entries, but can be specified in the Page Manager Spring Framework configuration. Caching intermediate profiling results could be more effective by taking advantage of frequently used Profiler Locators shared by multiple users, but is currently not leveraged.

All entries in the Profiled Page Context cache are cleared if any entries in the underlying File Cache are refreshed or evicted. Due to the relative complexity of composing these contexts, file system to context dependencies are not tracked. Furthermore, access times are not currently maintained for file system folders and documents in the File Cache, so cache behavior is not ideal. To prevent excessive Profiled Page Context cache resets, the size of the Page File Cache size must be set sufficiently large in the Spring Framework Configuration to limit cache evictions.