Expansion of Component Options

Expanders

Infusion component options, as written in `fluid.defaults` blocks, go through a process called **_expansion_** when they are used to instantiate a component. Two kinds of expansion happen during this process -

- Expansion of IoC references, written as strings in the form `"{context}.path"` as a result of the Value Resolution process, and
- Expansion of **expanders**, which are blocks of JSON occurring in the options with the key `expander`

The standard use of an expander is to designate a function to be called when instantiating the component options, which produces a value based on processing the expander arguments. This can be useful when static definition of a default option is not possible.

Expanders are specified using the keyword `"expander"` in the component defaults:

```
fluid.defaults("component.name", {
    optionName: {
        expander: {
            ...
        }
    }
});
```

The basic form of an expander record is very similar to that of an Invoker - it contains entries **func/funcName** together with **args** to designate the function call which will produce the required options values.

Name	Description
`func/funcName`	Either an IoC reference to a function (an invoker or other function member) or else a global function name holding the function to be invoked
`args`	An array of arguments (or single argument) to be passed to the user-provided function specified in `func/funcName`.

Examples

This example locates the global function named `"cspace.search.modelFilter"` and calls it with the arguments given by resolving the context `"{searchView}"` - in this case, most likely the top-level component defined in defaults itself. The return value from this function is then placed in the options of the instantiated component (the `"fluid.pager"`) at the path `modelFilter`:

```
fluid.defaults("cspace.search.searchView", {
    components: {
        resultsPager: {
            type: "fluid.pager",
            options: {
                modelFilter: {
                    expander: {
                        func: "cspace.search.
makeModelFilter",
                        args: ["{searchView}"]
                    }
                }
            }
        }
    }
});
```

Tthe `resultsPager` is specified as an instance of the Infusion Pager component. When this subcomponent is created, the expander will call the function `cspace.search.makeModelFilter`, passing it the parent `searchView` component as an argument. The return value will be used as the default `modelFilter` option to the Pager.

The `fluid.noexpand` expander

The `fluid.noexpand` expander is a very specialised expander that normal users of the framework should not require to use. It has been retained in the framework for completeness, but its effects should normally be obtained using a [mergePolicy](#) of `"noexpand"`. This expander simply dumps its literal argument (held at a path named `value` or `tree`) into the component's options without expansion.

Name	Description
type	`"fluid.noexpand"` (the type field must hold this literal value)
value /tree	This property holds some literal component configuration (either a primitive value or larger tree of JSON values) which will be inserted without expansion into the component options

Example:

```
fluid.defaults("cspace.specBuilder", {
    components: {
        specBuilderImpl: {
            type: "cspace.specBuilderImpl",
            options: {
                unexpanded: {
                    expander: {
                        type: "fluid.noexpand",
                        value: "{specBuilder}.
urlExpander"
                    }
                }
            }
        }
    }
});
```

In this example, the function name `{specBuilder}.urlExpander` will NOT be resolved as an IoC reference. The value {specBuilder}.urlExpander will be assigned to the option named `unexpanded`.