

# Copy of Default actions for dialogs - possible solutions summary

## A definition with default actions

- Was implemented as optional class and reverter because:
  - one still have to know the class name a add it to the config...although it's still better then define the two actions, we said that this is not enough
- We don't want to include default actions into the default dialog definition:
  - break compatibility
  - it's kind of magic (we provide a default for simple properties like Strings, numbers but can be confusing for maps or collections)

## Use decoration

It's a technical solution but how it's suppose to be configured?

Currently we have:

- dialog definitions
- their decorators (to be found under exact location)
- dialog ← decorators

For default actions we would need:

- ❌ dialog ← defaultActions ← decorators (this would not change the principle but the default actions would override the dialog)
- defaultActions ← dialog ← decorators (this is obviously changing the principle since the dialog itself would become a decorator)

➖ goes (at least partially) against the current impl

➖ YAML only (on the other hand...we need it for light dev mail + we have extends in JCR)

➕ decoration is trendy, let's use it for everything

[Quick POC Impl](#)

## Intelligent dialog builder

### Configuration

```
DialogDefinition.Builder()  
...  
    .withDefaultActions()
```

How it's suppose to be configured?

### Provide an implementation in the module descriptor

➕ different implementation per apps/subapps

➖ java developer required

### Default configurable implementation

Where to configure it? Do we need a new configuration class? Wouldn't it be the same as the dialog definition?

Probably yes. So we can use the dialog definition as the configuration for the builder. Then we can merge it. So why we would need a builder in the end?

➕ no java developer required

➖ more complexity

## Site prototype like

Templates are merged with the site prototype before being rendered.

+ users know this from the site module

+ no new classes needed

- Where to do the merge? Dialog registry → Dialog factory → Dialog presenter
  - `info.magnolia.config.registry.AbstractRegistry#getProvider(java.lang.String)`
  - `info.magnolia.ui.dialog.formdialog.FormDialogPresenterImpl#start()`
- How to name the prototype config
  - `prototype(.yaml)`
  - ?
- Where it's possible to define the prototype?
  - same as dialog definition
    - one prototype only (`ui-dialog/dialogs/prototype`), btw. `ui-dialog` doesn't exist as a module yet
    - per module `module/dialog/prototype`
  - anywhere as `prototype(.yaml) dialog`, the closest one is used

GIT