

Define defaults in a consistent order

The core definition of a component is its `'defaults block'`, defined using a call to `fluid.defaults()`. The defaults block defines the type of component, the events it fires, its subcomponents, and other integrator-configurable options.

Since the defaults block is the main definition of what a component is, consistency in its form makes modifying code and debugging much easier. The Fluid community has developed a preferred ordering of items in the component defaults block. Maintaining this ordering will make it easier for anyone looking at the code to understand the component.

The following code snippet includes all possible contents of a defaults block in the recommended order. (Note that any given component will likely not include *all* of these properties.)

```
fluid.defaults("my.component.name", {

    // gradeNames should always go first, so we know what "type" of component is being defined.
    gradeNames: ["fluid.rendererComponent", "autoInit"],

    // standard options for model-bearing components
    model: {},
    modelListeners: {},
    modelRelay: {},

    // standard options for evented components
    events: {},
    listeners: {},

    // standard options for view components
    selectors: {},
    strings: {},
    markup: {}, // HTML-formatted configuration values
    styles: {},

    // standard options for renderer components
    renderOnInit: true,
    selectorsToIgnore: [],
    repeatingSelectors: [],
    protoTree: {}, // mutually exclusive with produceTree
    produceTree: "",
    rendererFnOptions: {},
    rendererOptions: {},

    // templates, usually used with renderer components
    resources: {} // template

    // component specific options
    compOpt1: "",
    compOpt2: {},
    compOptETC: [],

    // general component members (usually static data)
    members: {},

    // component methods
    invokers: {},

    // child components
    components: {}
});
```