

# How to use CSS and D2W Rules Instead of Custom Components

Typically in a D2W application in order to display a boolean field/attribute as an image one would need to use a special component, `D2WDisplayBoolean`, or another similar but custom component. In Diva look, that's no longer the case.

In *ERDivaLookExample*, the property *isDirector* is displayed with CSS + rules:

## List

462 Talents	First Name	Last Name	Is Director
	Harrison	Ford	<input type="checkbox"/>
	George	Lucas	<input checked="" type="checkbox"/>

Standard CSS image replacement techniques are used here to display an image for **checked** and **unchecked** values.

In order to do this another new D2W key was introduced: **class**.

You use a **dynamic** D2W assignment for the *class*, so that *checked* values may be styled differently from *unchecked* values.

LHS	assignment	key	RHS
entity.name = 'Talent' and task = 'list' and propertyKey = 'isDirector'	ERDDelayedKeyValueAssignment	class	object.isDirector

Note: the component being used for *isDirector* is *ERD2WDisplayString*.

This wraps the string value in a `<span>` with a class defined by `d2wContext.class`.

Having done that, they may be styled accordingly:

```
span.true, span.false {
    text-indent: -5000px;
    display: block;
    width: 16px;
    height: 16px;
}

span.true {
    background: url(/WebObjects/Frameworks/JavaDirectToWeb.framework/WebServerResources/CheckboxOn.gif);
}

span.false {
    background: url(/WebObjects/Frameworks/JavaDirectToWeb.framework/WebServerResources/CheckboxOff.gif);
}
```

Similarly, any multi-valued property may be displayed just as easily - without the use of custom components.

Tip



Development of a Diva look app is mostly customization of the **CSS** and **D2W** (rules).