

# cfUniForm Quick Tip: Struct or String

Posted At : February 27, 2010 6:32PM | Posted By : Matt Quackenbush

Related Categories: ColdFusion, Uni-Form Tag Library

One of the improvements in the [latest cfUniForm release](#) is a change in behavior for the 'pluginSetup' attribute on the field tag. Even though this change does not break any old code, I still wanted to write a brief explanation of what the change is, why I made it, and how to make use of it.

## A Quick Review

NOTE: If you are already familiar with the 'pluginSetup' attribute and know what it is used for, feel free to skip this section. :-)

The 'pluginSetup' attribute is used to pass configuration parameters to cfUniForm, which, in turn, passes those parameters on to the appropriate plugin. So, for example, let's say you have a date field in your form, and you want to set the range of years for that particular field, you would use the 'pluginSetup' attribute.

```

.. <uform:field label= "Start Date"
}. name= "startDate"
}. isRequired= " true "
}. pluginSetup= "#startDateSetupParams#"
}. type= "date" />

```

## How It Changed

In previous releases, the 'pluginSetup' attribute required a string value be passed to it. Technically, in order to avoid a JavaScript error, it required a JSON string. So, using our example above, to provide params for the date range, our code would look something like this:

```

.. <cfset startDateSetupParams = "{yearRange : '2010:2012'}" />
}. <uform:field label= "Start Date"
}. name= "startDate"
}. isRequired= " true "
}. pluginSetup= "#startDateSetupParams#"
}. type= "date" />

```

All we've done here is provide the JSON equivalent to a struct, ultimately telling the date picker plugin to use a year range of 2010-2012. If you'd prefer to not deal with JSON, and just use a regular ol' ColdFusion struct, well, now you can.

```

.. <cfscript>
}. startDateSetupParams = structNew();
}. startDateSetupParams['yearRange'] = "2010:2012" ;
}. </cfscript>
}. <uform:field label= "Start Date"
}. name= "startDate"
}. isRequired= " true "
}. pluginSetup= "#startDateSetupParams#"
}. type= "date" />

```

cfUniForm will automatically convert your struct to JSON and pass it along to the plugin.

Two important things I want to point out about that last code sample:

1. I used array notation to declare the 'yearRange' key. Why? Because CF will UPPERCASE your key names if you don't. Why does that matter? Unlike CF, JavaScript is cAsE-sEnSiTiVe. So, 'YEARRANGE' is not the same as 'yearRange'. Therefore, if you provide 'YEARRANGE' to the plugin, nothing will happen.
- I used single quotes inside of double quotes when I declared the value of the key. Why? Some params in JavaScript require quotes. Some do not. In the case of 'yearRange', they are required. cfUniForm **will not** alter the values you specify. It is up to the developer to know when to use quotes and when not to. As a general rule you will need quotes on any value other than all-numbers or true/false.

### Why It Changed

"Meh, that JSON ain't so bad," you say.

And I agree with you. On that simple little example with one key, it is pretty simple and straight-forward to just write a JSON string. But, what if you are setting lots of params? What if you are setting nested params? The more parameters you have to set, the more cumbersome the JSON string becomes. Miss (or misplace) a comma (,) or a curly brace ({ or }) and you're screwed.

The bottom line is, cfUniForm was created for the specific purpose of making great looking, consistently coded, cross-browser supported forms quick and easy. By enabling structs instead of JSON, they just got a little bit easier!