

Uni-Form XHTML Forms Custom Tags v2

Posted At : June 2, 2008 12:39AM | Posted By : Matt Quackenbush

Related Categories: ColdFusion, Uni-Form Tag Library, jQuery

It's been almost six months now since I [released v1](#) of the library, so I figured it was high time I make time to add some enhancements. Here's a quick overview of the new features:

- Integrated dependencies
- Integrated support for the following jQuery plugins
 - Date Picker
 - Time Entry
 - Input Masking
 - Validation
- Path Configuration

Let's look at some of the new goodies in more detail...

Integrated Dependencies

After a few months of using the library myself, one of the things that started to annoy me was the fact that I had to add the following into all of my applications in order to use the library:

```

.. <link href= "/cfMgmt/common/UniForm/css/uni-form.css" type= "text/css" rel= "stylesheet" media= "all" />
?. <link href= "/cfMgmt/common/UniForm/css/uni-form-styles.css" type= "text/css" rel= "stylesheet" media= "all" />
?. <!--[ if lte ie 6]>
?. <style type= "text/css" media= "screen" > .uniForm,.uniForm fieldset,.uniForm .ctrlHolder,.uniForm .formHint
?. { /* Trigger hasLayout*/ zoom:1;} </style>
?. <![endif]-->
?. <InvalidTag src= "/cfMgmt/common/UniForm/js/uni-form.jquery.js" type= "text/javascript" > </script>

```

I originally "fixed" this by using a `<cfinclude />`. While this worked okay, it was still kinda annoying to me, as I did not want to include the requirements on every page load. With this release, you no longer have to worry about that nonsense, because the dependencies are loaded automagically when you call the library.

Integrated jQuery plugins: Date Picker

v2 has a brand new 'loadDateUI' argument on the form tag, which accepts a boolean. If your form needs a [date picker](#) in it, just provide this new argument and set it to true.

```

.. <uform:form action= "action.cfm"
?. id= "myForm"
?. loadDateUI= " true " >

```

Integrated jQuery plugins: Time Entry

Need a [time entry](#) field? Just use the new 'loadTimeUI' argument...

```

.. <uform:form action= "action.cfm"
?. id= "myForm"

```

```
}. loadTimeUI= " true " >
```

Integrated jQuery plugins: Input Masking

"What about [input masking](#)?" you ask. You guessed it; just supply the 'loadMaskUI' argument...

```
.. <uform:form action= "action.cfm"
?. id= "myForm"
?. loadMaskUI= " true " >
```

Integrated jQuery plugins: Validation

What good is a form without validation, right? Not much, to be sure. v2 utilizes [jQuery](#)'s [form validation](#) plugin, which is integrated by setting the 'loadValidation' argument to true...

```
.. <uform:form action= "action.cfm"
?. id= "myForm"
?. loadValidation= " true " >
```

... along with the new 'validation' argument for the field tag.

```
.. <uform:field label= "First Name"
?. name= "contactFirstName"
?. isRequired= " true "
?. type= "text"
?. validation= "required" />
```

One of the really bad-ass things about this plugin is that you can do basic client-side validation by simply adding a class to the field, such as 'required'. The plugin will check to make sure that field was filled out, and displays an error message if it isn't. Just supply validation="required" as shown above, and you're done. No JavaScript to write. Nothing. Just add it to your CF and stick a fork in it.

For those of you who like to take advantage of the more advanced features of the validation plugin, you *should* be able to continue to do so by utilizing this same argument. However, I must admit that I have not done any advanced testing with it as of yet, so if you run into any difficulties with that, please feel free to contact me and hopefully we can work together to iron it out.

Path Configuration

To load all of the various dependencies, the library uses convention to locate the required files. Should your application's directory structure require you to use paths other than the defaults (same directory structure as the [download](#)), you can utilize the form tag's new 'config' argument to supply the correct paths.

```
.. <uform:form action= "action.cfm"
?. id= "myForm"
?. config= "#myConfigStruct#" >
```

As you can see, the config argument accepts a structure. Valid keys are:

- renderer: path to the error message renderer
- uniformCSS: path to the uni-form.css file
- uniformJS: path to the jquery.uni-form.js file

- validationJS: path to the jquery.validate.js file
- dateCSS: path to the ui.datepicker.css file
- dateJS: path to the ui.datepicker.js file
- timeCSS: path to the jquery.timeentry.css file
- timeJS: path to the jquery.timeentry.js file
- maskJS: path to the jquery.maskedinput.js file

IMPORTANT NOTE #1!

Some of the JavaScript files have had modifications made to them in order to integrate with the Uni-Form markup. Changing the files will break things, so I would recommend changing only the *path* to them!

IMPORTANT NOTE #2!

For page load speed, the [packed](#) versions of the various JavaScript files are used. If you get really, really bored and decide to dig into them to see what changes I made, good luck. ;-)

IMPORTANT NOTE #3!

ALL of the new 'load{Dependency}' arguments default to FALSE. If you're not using a particular plugin for your form, no need to supply the argument. The scripts/CSS will **not** be loaded.

IMPORTANT NOTE #4!

It is assumed that you use jQuery for more than just your forms, and therefore already have it loaded by default. If, however, you don't, I decided at the last minute to include an argument to load it automatically with the form...

```
.. <uform:form action= "action.cfm"
}. id= "myForm"
}. loadjQuery= " true " >
```

If you're not using jQuery elsewhere in your application, what the hell is wrong with you??!!??!!

:-)

Oh yeah, and if you decide to alter the path to jQuery itself, you can supply the 'jQuery' key in the config struct.

Summing it Up

Wanna [see it in action](#) ?

Thanks to [Dan Wilson](#) for his contributions, and to everyone else out there that has had kind words for the library. You guys (and gals) all make it fun to release code. :-)

Update

Please see the [2.1 release](#) for the download.