

How do you decide what to use: UDF or Custom Tag?

Asked 15 years, 9 months ago Modified 12 years, 11 months ago

Viewed 1k times



[WACK](#) says:

3



If you feel you need to have lots of arguments, consider creating a CT instead.

...



CT are significantly more powerful and flexible than custom functions. Try to use UDFs for simple matters... Use CT and Components for more involved processes, especially those you can think of as discrete actions rather than simple "massaging"

Okay, but how are you usually making decisions?
Interesting to know real-life practice and examples.

For me it happens when a function has many not-required arguments, so I have to call them `myFunc(arg1="foo", arg2="bar")`. Sometimes `<cfmodule>` syntax simply becomes more readable, but not always.

Other reason is that I don't like long (say, more than 2 screens of code) UDFs.

But all these thoughts are very subjective, that's why I'm interested in reading other people opinions. Maybe there's better rules for that?

Thanks in advance.

coldfusion

user-defined-functions

custom-tag

Share

edited Jan 14, 2012 at 1:06

Improve this question

Follow

community wiki
3 revs, 2 users 66%
Sergii

@Peter Have no idea. Only Ben F. knows :-)

– [Sergey Galashyn](#) Mar 16, 2009 at 7:31

7 Answers

Sorted by:

Highest score (default)





5



There are probably plenty of people in the community that would disagree with me, but here is how I generally think about it. If what you need to do involves output to the screen, and if it makes sense to "wrap" this around some other code or text, then a custom tag might be in order. In all other cases a UDF works fine and generally better. That being said, in close to 8 years of CF development, I really haven't ever come across a very good reason for a custom tag. I'm not saying the reasons don't exist, but I would say that they are rare.

If you have a very long UDF, is it safe to assume that this is something where you are outputting to the screen by calling this UDF (not really returning a value I mean)? I would consider breaking that UDF into smaller more manageable parts if that could make sense, but like you are alluding to, what matters in the end is what is the most readable, to you, your team and those who may come after you.

Update: Out of curiosity, which CFWACK book are you referring to (version and volume) and which page?

Share Improve this answer

edited Mar 16, 2009 at 12:41

Follow

answered Mar 16, 2009 at 12:29



Ryan Guill

13.9k ● 5 ● 39 ● 48

For a terrific example of the potential power and complexity of Custom Tags, look at the tags included (for encapsulating common content between themes/skins, making them easier to implement) in Mango Blog. They blow my mind every time I look at them! – [Adam Tuttle](#) Mar 16, 2009 at 17:27

Again, I'm not saying that good examples and uses of custom tags don't exist, just that you don't come across too many places where they really are a better way of doing things than UDF's. Templating like this is a good example though.
– [Ryan Guill](#) Mar 16, 2009 at 17:31

@Adam: Thanks for the tip! @Ryan: Can't say page for sure right now, because book in office now, but it is CF8 WACK Book 2 Chapter 24 (UDF). Page # is near 40.
– [Sergey Galashyn](#) Mar 16, 2009 at 19:43



2



As I remember things, custom tags can be called at any time, where as UDFs have to be defined before you can use them. This makes custom tags easier generally. If you had huge libraries of UDFs it would be burdensome to make sure they are all included and potentially hard work for the server to parse them all (in olden days at least).

However UDFs can be used in a more compact way

```
<cfif myUdf(myVariable)>
```

The advantage of custom tags is that they can sit nicely with your markup.

```
<h1>Order Page</h1>
<cf_basket_nav>
  <ul>
    <cfloop ...>
      <li>
        <cf_basket_item item="#items[i]#">
      </li>
    </cfloop>
  </ul>
</cf_basket_nav>
```

Generally nowadays I would have a 'utils' CFC with methods for what were your UDFs.

Share Improve this answer

edited Mar 16, 2009 at 12:56

Follow

answered Mar 16, 2009 at 12:37



Jeremy French

12.1k ● 6 ● 47 ● 72

If you mean that the UDF has to be included or defined before you can use it, this is not true. – [Ryan Guill](#) Mar 16, 2009 at 12:40

Umm, isn't it. I know you can put them in some persistent scope but as far as I know you can't just call a udf and expect CF to know what you mean? – [Jeremy French](#) Mar 17, 2009 at 9:12



one consideration on the use of custom tags over udfs or cfc methods is when you find that a subroutine needs to

1



be passed an array of child items, you can use nested custom tags to associate a child custom tag and its elements to a parent custom tag. this allows you to do very nice clean coding thats easy to read:

```
<cfmenubar ... >
  <cfloop array="menuitems" ...>
    <cfmenubaritem url="#i.url#">
      #i.label#
    </cfmenubaritem>
  </cfloop>
</cfmenubar>
```

yes,yes i know we have nicer dhtml stuff like menus and tabs, this is simply to point out an example. you can use cfassociate in the custom tag to "pass" the attributes to the parent custom tag and then in the executionmode="end" access all the dynamically generated child items in the array of associated attributes. this is where you would loop and output the menu to the screen in this example.

also, as another commented, allows you to do some clever things... one thing in particular i use is that i set prefix="" and then i can essentially force simple html tags (like the <a> tag) to get kicked through a custom tag handler at runtime - so an html tag becomes intelligent at runtime... i do this so i can analyze the href and the target attributes and decide if i want to display a pdf icon (or other mime type icon) next to the link... its pretty slick! this is especially helpful in a content management system or when 7you have html developers using dreamweaver or

contribute and you want to have their tags fire smart coldfusion tags without them doing anything outside of standard html - the editor doesnt know any difference and they dont need to go into "code" view to make some fairly powerful functionality.

finally, in a custom tag you can choose to suppress output (or use a cache), so this can be very useful to wrap around chunk of dynamically generated html... access the `this.tag.generatedContent` variable in the `executionmode EQ "end"` mode

dont throw out the baby with the bathwater on this one ... i agree their usage is much less frequent since we have cfcs, however there is still some powerful functionality in custom tags... i usually have one or 2 in every application (and at least dozens of cfcs)

hth jon

Share Improve this answer

answered Mar 27, 2009 at 21:41

Follow



Jon Briccetti



`cfmodule` has no advantage over functions; `cfinvoke` works just the same.

0



Of course, with `cfimport` you can enable a nice tidy namespaced CT syntax - and this is when custom tags are good: when working with clear open/close construct, or if nested logic is important.



For everything else, especially when returning results, functions are easier to handle.



Share Improve this answer

answered Mar 16, 2009 at 0:09

Follow



[Peter Boughton](#)

112k ● 32 ● 123 ● 177



0

Jeremy pointed to useful option: wrapping HTML by CT. But this feature seems to be so rarely used. Any way, it's a plus for CT.



Also I thought that I prefer function because I love cfscript coding. Few times I had to wrap legacy CT into UDF only to be able to use it at totally cfscript-ed page. It's a plus for UDF.



Share Improve this answer

answered Mar 16, 2009 at 19:51

Follow



[Sergey Galashyn](#)

6,956 ● 2 ● 20 ● 39



0

Simple reusable code can go into a UDF. This might include things like string formatting, data structure manipulation, etc etc.



For anything more complex than that, I would consider using CFCs instead of a custom tag. You'll be able to write much cleaner code in an OO model. You can always wrap the logic in your CFCs with a custom tag if you want



to provide ease of use for someone who is more used to working with tags.

Share Improve this answer

answered Mar 22, 2009 at 0:35

Follow



[cliff.meyers](#)

17.7k ● 6 ● 53 ● 67



0



Not sure why I fell into this pattern, but in general I use Custom Tags (always with cfmodule) for anything that outputs HTML and UDFs for anything that merely returns simple data/objects. I realize that UDFs can do output as well, but I don't like my functions to have any side effects and this feels like one. For similar reasons, I now use Custom Tags anywhere where previously I would have used a cfinclude, since they provide encapsulation of data.

Share Improve this answer

answered Mar 27, 2009 at 22:11

Follow



[Soldarnal](#)

7,768 ● 9 ● 48 ● 65