

Custom Tags

Custom tags also reuse blocks of code. The code is stored in a single page that sits on the server and may be called by other pages much like the `<cfinclude>` tag does. There are some key differences, however. Custom tags are used as if they are tags that exist in CFML. They have attributes that can pass in parameters.

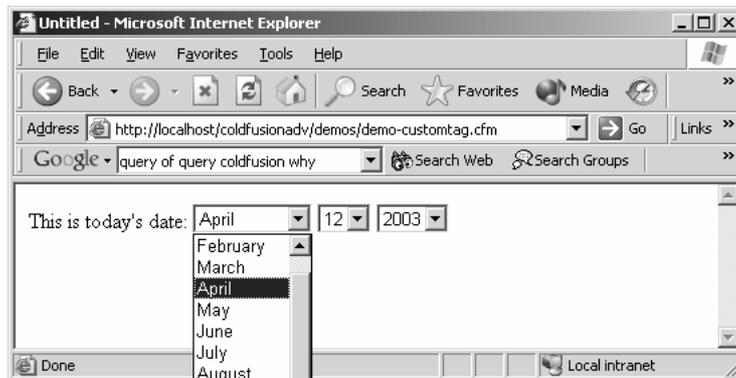
Custom Tags can take complex or tedious code and hide it. It can clean up the main code on a page by storing it elsewhere. Most importantly, this code may be changed once and will immediately affect all pages that call it.

Complicated custom tags may be created by a senior developer and specs can be distributed for everyone to use. Since using a custom tag is no more complicated than using any CFML tag, they are easy to implement.

Demo: custom tags

Perhaps a common desire on your site is to display today's date in the form of a select list. For example, an airline might provide a search feature to check the status of a flight. Perhaps most users are checking the status of today's flights; however, some users might need to select another date. The demo below shows a page that displays today's date in the form of three select lists. The values selected could be passed to the next page which will process that data.

The following demo is saved as **demos/demo-customtag.cfm**:



The code for the page is listed below. Notice the tag `<cf_todaysdate>`. The underscore that follows "cf" tells ColdFusion that this is a custom tag. The rest of the tag name is simply the name of the document without the ".cfm!" This tag must be saved as "todaysdate.cfm."

```
<html>
<head>
  <title>Demo: Custom Tag</title>
```

```

</head>
<body>
This is today's date:

<cf_todaysdate>

</body>
</html>

```

The custom tag is saved as **demos/todaysdate.cfm**. Notice the comment at the top. This is not a requirement, but a suggestion. Providing some information about this tag will be helpful to other developers who need to use it and will serve as a reminder to you when you open it a year later!

```

<!---
  Name: Select List with Current Date

  Attributes: None

  Description: This tag determines the current date
               and builds a select list that will allow the
               user to choose it or change to another day.

               It must be used within an html <form> tag

  --->
<cfset thedate = now()>

<cfset nowmonth=#Month(thedate) #>
<cfset nowdate=#Day(thedate) #>
<cfset nowyear=#Year(thedate) #>

<select name="m">
  <cfif nowmonth IS 1><option selected
value="01">January</option><cfelse><option
value="01">January</option></cfif>
  <cfif nowmonth IS 2><option selected
value="02">February</option><cfelse><option
value="02">February</option></cfif>
  <cfif nowmonth IS 3><option selected
value="03">March</option><cfelse><option value="03">March</option></cfif>
  <cfif nowmonth IS 4><option selected
value="04">April</option><cfelse><option value="04">April</option></cfif>
  <cfif nowmonth IS 5><option selected
value="05">May</option><cfelse><option value="05">May</option></cfif>
  <cfif nowmonth IS 6><option selected
value="06">June</option><cfelse><option value="06">June</option></cfif>
  <cfif nowmonth IS 7><option selected
value="07">July</option><cfelse><option value="07">July</option></cfif>
  <cfif nowmonth IS 8><option selected
value="08">August</option><cfelse><option
value="08">August</option></cfif>

```

```

    <cfif nowmonth IS 9><option selected
value="09">September</option><cfelse><option
value="09">September</option></cfif>
    <cfif nowmonth IS 10><option selected
value="10">October</option><cfelse><option
value="10">October</option></cfif>
    <cfif nowmonth IS 11><option selected
value="11">November</option><cfelse><option
value="11">November</option></cfif>
    <cfif nowmonth IS 12><option selected
value="12">December</option><cfelse><option
value="12">December</option></cfif>
</select>

<select name="d">
    <cfif nowdate IS 01><option
selected>01</option><cfelse><option>01</option></cfif>
    <cfif nowdate IS 02><option
selected>02</option><cfelse><option>02</option></cfif>
    <cfif nowdate IS 03><option
selected>03</option><cfelse><option>03</option></cfif>
    <cfif nowdate IS 04><option
selected>04</option><cfelse><option>04</option></cfif>
    <cfif nowdate IS 05><option
selected>05</option><cfelse><option>05</option></cfif>
    <cfif nowdate IS 06><option
selected>06</option><cfelse><option>06</option></cfif>
    <cfif nowdate IS 07><option
selected>07</option><cfelse><option>07</option></cfif>
    <cfif nowdate IS 08><option
selected>08</option><cfelse><option>08</option></cfif>
    <cfif nowdate IS 09><option
selected>09</option><cfelse><option>09</option></cfif>
    <cfif nowdate IS 10><option
selected>10</option><cfelse><option>10</option></cfif>
    <cfif nowdate IS 11><option
selected>11</option><cfelse><option>11</option></cfif>
    <cfif nowdate IS 12><option
selected>12</option><cfelse><option>12</option></cfif>
    <cfif nowdate IS 13><option
selected>13</option><cfelse><option>13</option></cfif>
    <cfif nowdate IS 14><option
selected>14</option><cfelse><option>14</option></cfif>
    <cfif nowdate IS 15><option
selected>15</option><cfelse><option>15</option></cfif>
    <cfif nowdate IS 16><option
selected>16</option><cfelse><option>16</option></cfif>
    <cfif nowdate IS 17><option
selected>17</option><cfelse><option>17</option></cfif>
    <cfif nowdate IS 18><option
selected>18</option><cfelse><option>18</option></cfif>
    <cfif nowdate IS 19><option
selected>19</option><cfelse><option>19</option></cfif>
    <cfif nowdate IS 20><option
selected>20</option><cfelse><option>20</option></cfif>

```

```

    <cfif nowdate IS 21><option
selected>21</option><cfelse><option>21</option></cfif>
    <cfif nowdate IS 22><option
selected>22</option><cfelse><option>22</option></cfif>
    <cfif nowdate IS 23><option
selected>23</option><cfelse><option>23</option></cfif>
    <cfif nowdate IS 24><option
selected>24</option><cfelse><option>24</option></cfif>
    <cfif nowdate IS 25><option
selected>25</option><cfelse><option>25</option></cfif>
    <cfif nowdate IS 26><option
selected>26</option><cfelse><option>26</option></cfif>
    <cfif nowdate IS 27><option
selected>27</option><cfelse><option>27</option></cfif>
    <cfif nowdate IS 28><option
selected>28</option><cfelse><option>28</option></cfif>
    <cfif nowdate IS 29><option
selected>29</option><cfelse><option>29</option></cfif>
    <cfif nowdate IS 30><option
selected>30</option><cfelse><option>30</option></cfif>
    <cfif nowdate IS 31><option
selected>31</option><cfelse><option>31</option></cfif>
</select>

```

```

<select name="y">
    <cfif nowyear IS 2000><option
selected>2000</option><cfelse><option>2000</option></cfif>
    <cfif nowyear IS 2001><option
selected>2001</option><cfelse><option>2001</option></cfif>
    <cfif nowyear IS 2002><option
selected>2002</option><cfelse><option>2002</option></cfif>
    <cfif nowyear IS 2003><option
selected>2003</option><cfelse><option>2003</option></cfif>
    <cfif nowyear IS 2004><option
selected>2004</option><cfelse><option>2004</option></cfif>
    <cfif nowyear IS 2005><option
selected>2005</option><cfelse><option>2005</option></cfif>
    <cfif nowyear IS 2006><option
selected>2006</option><cfelse><option>2006</option></cfif>
    <cfif nowyear IS 2007><option
selected>2007</option><cfelse><option>2007</option></cfif>
    <cfif nowyear IS 2008><option
selected>2008</option><cfelse><option>2008</option></cfif>
    <cfif nowyear IS 2009><option
selected>2009</option><cfelse><option>2009</option></cfif>
</select>

```

Where to store Custom Tags

Custom Tags may be stored:

1. In the same directory as the calling page

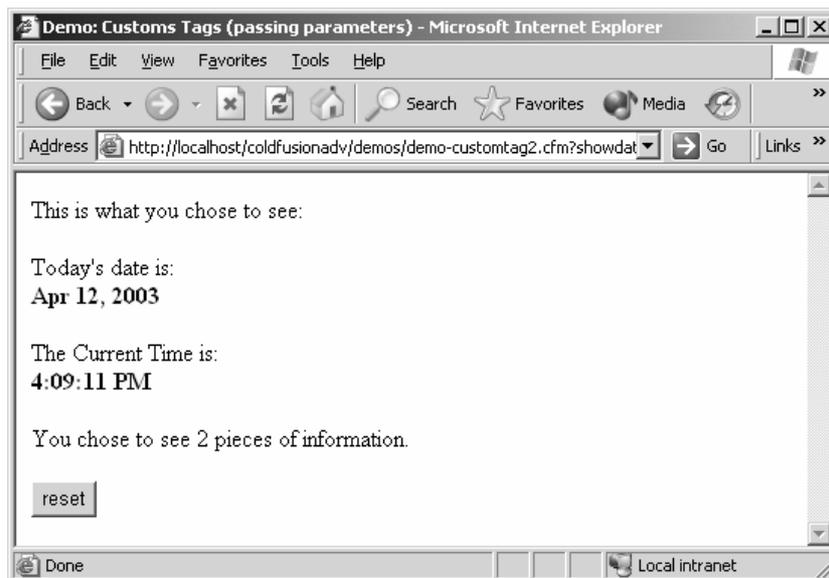
2. In the Custom Tags folder. (On a Windows server, this is typically C:\cfusionmx\CustomTags\)
3. In a subfolder of the Custom Tags Folder.

Demo: Custom Tag with Parameters

This demo uses a custom tag that can accept two parameters. The calling page has two phases. In phase 1, the user is asked to choose from two select lists. This will determine (a) whether the date will be shown and (b) whether the time will be shown.



In phase two, the requested items are shown. Additionally, a note appears above the reset button that shows how many of the items were selected to be shown.



The calling page is saved as **demos/demo-customtag2.cfm**. The most important tag is the custom tag itself. Notice that unlike the previous demo, this one has two attributes.

```
<html>
<head>
  <title>Demo: Customs Tags (passing parameters)</title>
</head>
<body>
<cfif NOT isdefined('beenhere')>
<h2>Please select the date and/or the time</h2>
  <form>
    Show the date:<br>
    <select name="showdate">
      <option>yes</option>
      <option>no</option>
    </select><br>
    Show the time:<br>
    <select name="showtime">
      <option>yes</option>
      <option>no</option>
    </select><br>
    <input type="hidden" name="beenhere" value="true">
    <input type="submit">
  </form>
<cfelse>
  This is what you chose to see:<br>
  <cf_showdate showdate="#showdate#" showtime="#showtime#"><br>
  You chose to see <cfoutput>#number#</cfoutput> pieces of information.
  <form>
    <input type="submit" value="reset">
  </form>
</cfif>
</body>
</html>
```

CALLER and ATTRIBUTE scopes

The custom tag is saved as **demos/showdate.cfm**. The attributes that were passed are available using the prefix “ATTRIBUTES” as in “ATTRIBUTES.showdate”. Notice that a default value is set using the <cfparam> tags in order to make these attributes optional.

The CALLER scope is used when the custom tag accesses a variable in the calling page. In this case, the custom tag page is setting a value for the variable named number. That value is displayed on the calling page above the reset button.

```
<!---
  Name: Select List with Current Date

  Attributes: showdate (optional) - yes/no
```

showtime (optional) - yes/no

Description: This tag determines the current date and displays the date and/or the time
--->

```
<!-- Set defaults for optional attributes --->
<cfparam name="ATTRIBUTES.showdate" default="yes">
<cfparam name="ATTRIBUTES.showtime" default="yes">

<cfset thedate=now()>

<cfif ATTRIBUTES.showdate IS "yes">
  <br>
  Today's date is:<br>
  <cfoutput><b>#DateFormat(thedate, 'medium')#</b></cfoutput><br>
</cfif>

<cfif ATTRIBUTES.showtime IS "yes">
  <br>
  The Current Time is:<br>
  <cfoutput><b>#TimeFormat(thedate, 'medium')#</b></cfoutput><br>
</cfif>

<!-- Set Caller.number variable --->
<cfif ATTRIBUTES.showtime IS "no" AND ATTRIBUTES.showdate IS "no">
  <br>
  YOU CHOSE TO SEE NOTHING!!
  <cfset CALLER.number=0>
<cfelseif ATTRIBUTES.showtime IS "yes" AND ATTRIBUTES.showdate IS "yes">
  <cfset CALLER.number=2>
<cfelse>
  <cfset CALLER.number=1>
</cfif>
```

Using <cfmodule>

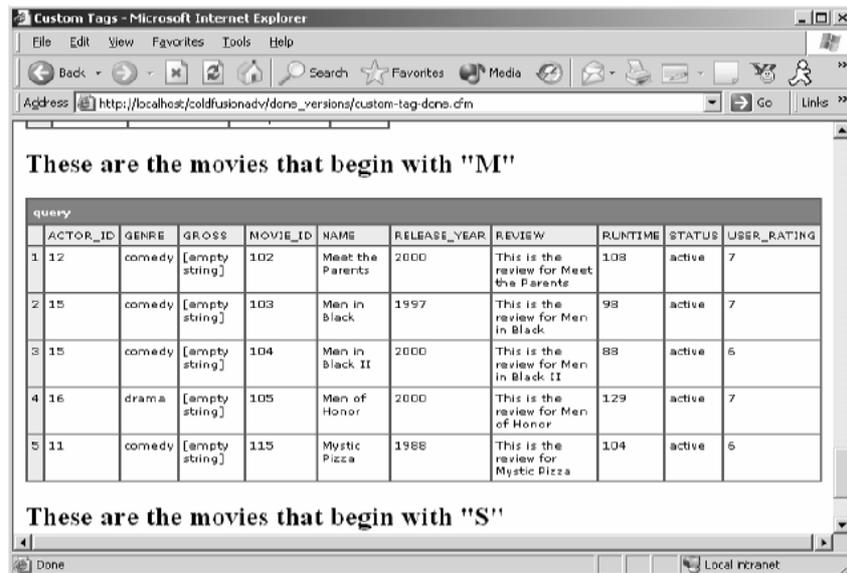
As stated above, custom tag files are stored in one of three places: the current folder, the **C:\cfusionmx\CustomTags** folder or one of its subdirectories. By using the <cfmodule> tag, it is also possible to store your custom tag document in any folder. You merely need to state the path to that document as the value of an attribute called *template*.

For example, if the custom tag is located in the folder above the directory in which the calling page is located the following tag the <cfmodule> tag must be used. Attributes may be set using the identical fashion as standard custom tags.

```
<cfmodule template="../showdate.cfm" showdate="#showdate#"
showtime="#showtime#">
```


Exercise 6: Using Custom Tags

In this exercise you will build two custom tags and will call each twice passing different parameters. When completed, it might look something like this:



1. First we will build the custom tags. Open the document named **pickatable-temp.cfm**. In it, you will need to add a query that will expect to receive an attribute named "tablename." That attribute will be used directly in the query and will select the records for that table. Also, add a `<cfdump>` tag that will display the resulting records. When you open it you will find the following code:

```
<!---
```

```
Add a query that will select all the fields from whichever table is passed through the attribute named 'tablename'
```

```
Also, use the cfdump tag to display the resulting records
```

```
---->
```

2. Open **somemovies-temp.cfm** which will also need a query and a dump tag. This time, the query will use the attribute named "letter" in the WHERE clause. When you open it you will find:

```
<!---
```

```
Add a query that will select all the fields from the movie table that have the same first letter as the one passed in through the attribute names 'letter'
```

```
Also, use the cfdump tag to display the resulting records
```

```
---->
```

3. Finally, open **custom-tag-temp.cfm** and you will find the following code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head>
  <title>Custom Tags</title>
</head>

<body>
<h2>This is the movies table</h2>
<!--
insert a call to the custom tag named pickatable-temp
be sure to pass the name of the table that you would
like to be displayed. In this case "movies"
-->

<h2>This is the actors table</h2>

<!--
insert a call to the custom tag named pickatable-temp
be sure to pass the name of the table that you would
like to be displayed. In this case "actors"
-->

<h2>These are the movies that begin with "M"</h2>
<!--
insert a call to the custom tag named somemovies-temp
be sure to pass the letter M as an attribute
-->

<h2>These are the movies that begin with "S"</h2>
<!--
insert a call to the custom tag named somemovies-temp
be sure to pass the letter S as an attribute
-->

</body>
</html>
```

4. Insert four custom tag calls to:

- a. Call pickatable-temp, passing “movies” as the value for the attribute tablename
- b. Call pickatable-temp, passing “actors” as the value for the attribute tablename
- c. Call somemovies-temp, passing “m” as the value for the attribute letter
- d. Call somemovies-temp, passing “s” as the value for the attribute letter

5. Test your page in a browser!

Possible Solution to Exercise 6

Saved as **done_versions/custom-tag-done.cfm**:

```
<html>
<head>
  <title>Custom Tags</title>
</head>
<body>
<h2>This is the movies table</h2>
<cf_pickatable-done tablename="movies">

<h2>This is the actors table</h2>
<cf_pickatable-done tablename="actors">

<h2>These are the movies that begin with "M"</h2>
<cf_somemovies-done letter="m">

<h2>These are the movies that begin with "S"</h2>
<cf_somemovies-done letter="s">

</body>
</html>
```

Saved as **done_versions/pickatable-done.cfm**:

```
<cfquery name="getstuff" datasource="cf_adv">
SELECT *
FROM #ATTRIBUTES.tablename#
</cfquery>

<cfdump var="#getstuff#">
```

Saved as **done_versions/somemovies-done.cfm**:

```
<cfquery name="getstuff" datasource="cf_adv">
SELECT *
FROM movies
WHERE name LIKE '#ATTRIBUTES.letter#%'
ORDER BY name
</cfquery>

<cfdump var="#getstuff#">
```