

Customizing Display with <cfoutput>

In the last section you learned a quick way to display resulting records with <cfdump>. Although it is fast and easy, the lack of control over the layout makes it a poor choice for most applications. However, it is a wonderful tool when building an application or for debugging.

If you have any programming experience, you might expect to learn about loops now. However, ColdFusion has an automatic method of looping that uses a tag called <cfoutput>.

A loop is piece of code that executes a certain number of times. A query will return “X” number of records. Once for each record a table row should be created to display the value of that row. A loop can be used to create the table we desire.

ColdFusion has a tag called <cfloop> that allows for the creation of loops. However, the <cfoutput> tag can be used as an *automatic* loop that loops through each resulting record of the query. It is easier to use than <cfloop>. Anything and everything that appears between the open and close <cfoutput> tags will be repeated once for each resulting record that was returned in the query.

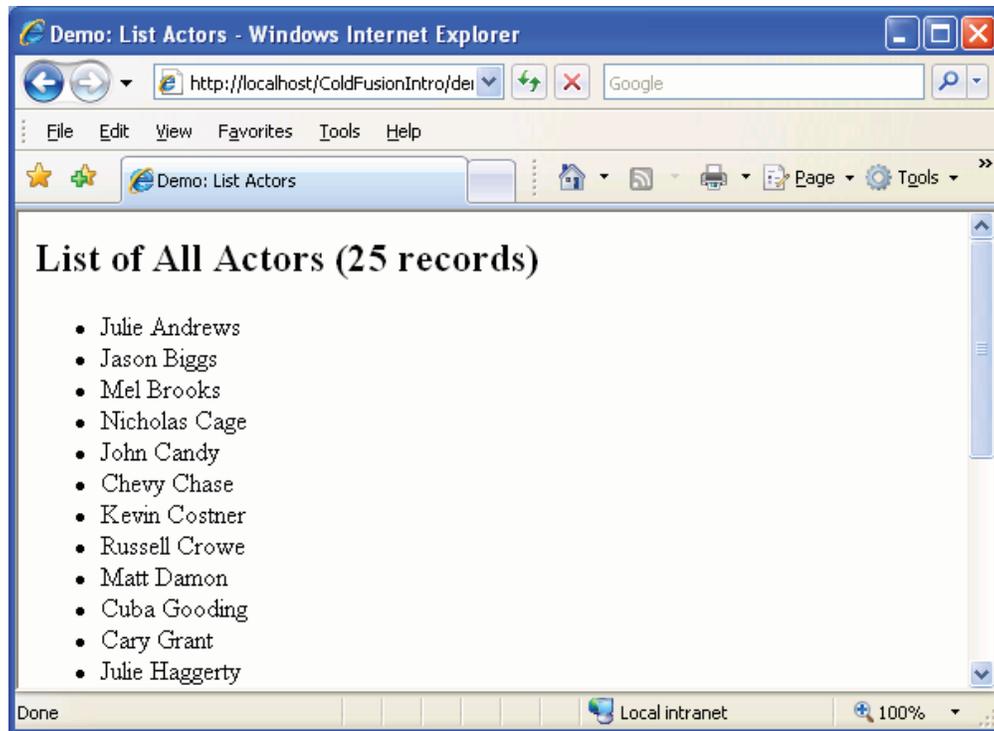
You have already been using <cfoutput> to display the value of various variables. These variables did not come from database queries, however. In the following example you should take note of the additional attribute used in the <cfoutput> tag: The query attribute is what connects this <cfoutput> block to the resulting records of one particular query

```
<cfoutput query="query_name">
```

By connecting an output block to a specific query, you are telling it to repeat once for each resulting record. Everything inside the output block will be repeated.

Demo: Using <cfoutput>

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



The code for **demos/demo-output.cfm** appears below.

```
<html>
<head>
  <title>Demo: List Actors</title>
</head>

<body>
<cfquery name="getActors" datasource="movieList">
  SELECT actor_id, firstname, lastname
  FROM actors
  ORDER BY lastname
</cfquery>

<h2>List of All Actors (<cfoutput>#getActors.recordcount#</cfoutput>
records)</h2>
<ul>
  <cfoutput query="getActors">
    <li>#firstname# #lastname#</li>
  </cfoutput>
</ul>

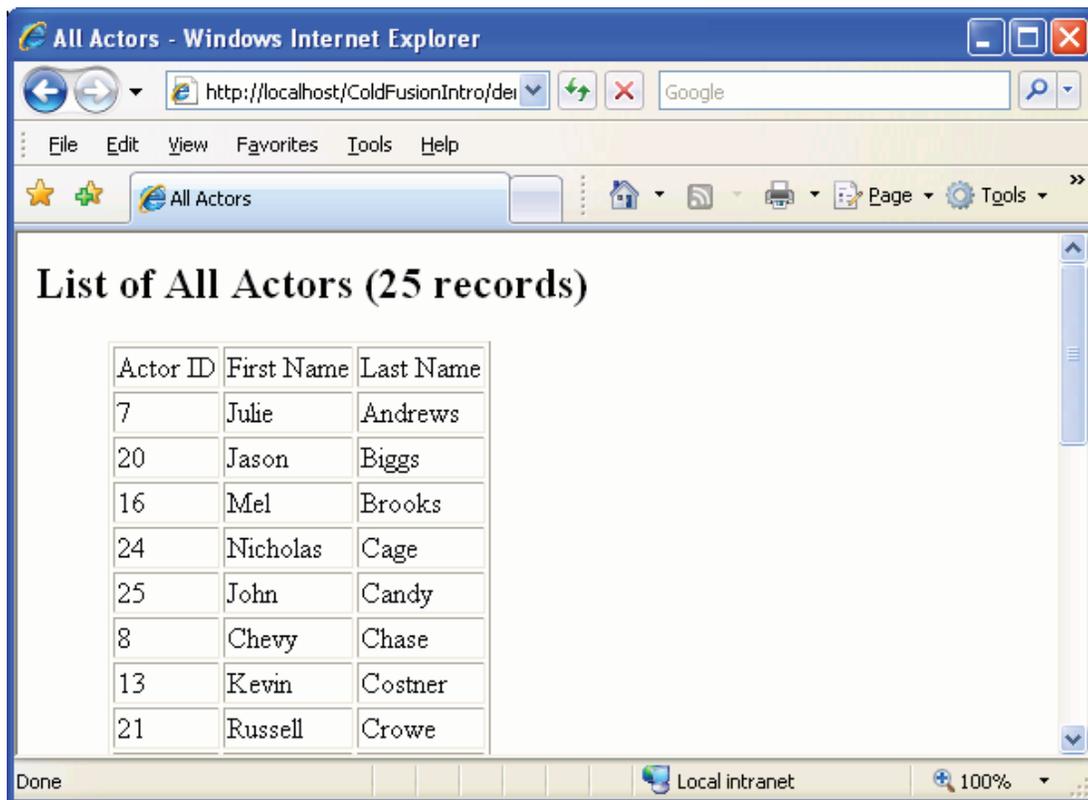
</body>
</html>
```

This demo has two `<cfoutput>` blocks. One has a `query` attribute and the other does not. Can you figure out why?

Building Dynamic Tables

Since the `<cfoutput>` tag will repeat anything that falls inside it, it is possible to build any repeating-data display mechanism desired – including HTML tables. In fact, HTML tables lend themselves quite well to the task. Usually, each record will be displayed inside its own row.

In order to build the HTML displayed below, a table row and all of its contents will be placed within a `<cfoutput>` block. Anything that will appear only once (like the header row, the `<table>` tag and its closing `</table>`) will appear on the outside of the `<cfoutput>` block.

A screenshot of a Windows Internet Explorer browser window. The title bar reads "All Actors - Windows Internet Explorer". The address bar shows "http://localhost/ColdFusionIntro/dei". The browser's menu bar includes "File", "Edit", "View", "Favorites", "Tools", and "Help". Below the menu bar is a toolbar with icons for home, back, forward, print, and page settings. The main content area displays a heading "List of All Actors (25 records)" followed by a table with three columns: "Actor ID", "First Name", and "Last Name". The table contains eight rows of data. The status bar at the bottom shows "Done", "Local intranet", and "100%".

Actor ID	First Name	Last Name
7	Julie	Andrews
20	Jason	Biggs
16	Mel	Brooks
24	Nicholas	Cage
25	John	Candy
8	Chevy	Chase
13	Kevin	Costner
21	Russell	Crowe

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

```
<html>
<head>
  <title>All Actors</title>
</head>

<body>
<cfquery name="getActors" datasource="movieList ">
  SELECT actor_id, firstname, lastname
  FROM actors
  ORDER BY lastname
</cfquery>
```

```

<h2>List of All Actors (<cfoutput>#getActors.recordcount#</cfoutput>
records)</h2>
<table border="1">
  <tr>
    <td>Actor ID</td>
    <td>First Name</td>
    <td>Last Name</td>
  </tr>
  <cfoutput query="getActors">
    <tr>
      <td>#actor_id#</td>
      <td>#firstname#</td>
      <td>#lastname#</td>
    </tr>
  </cfoutput>
</table>
</body>
</html>

```

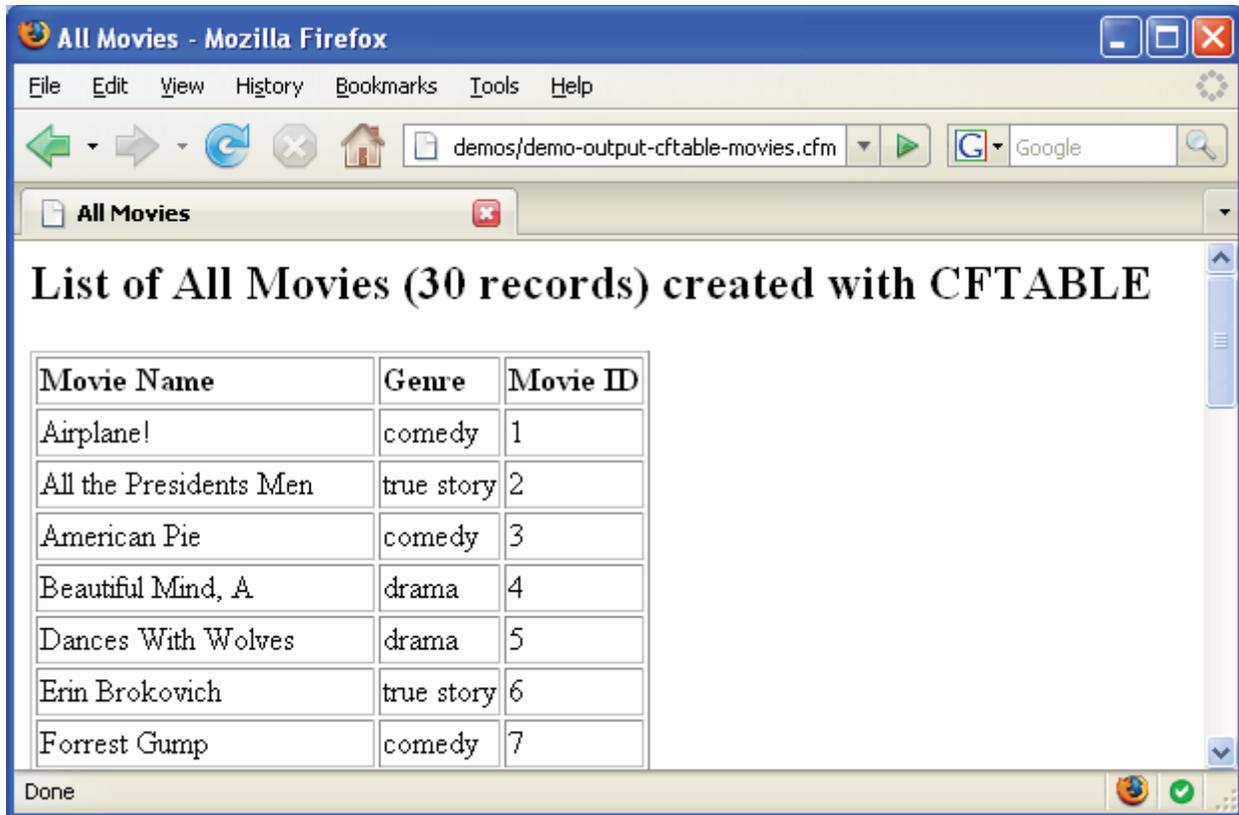
Shortcut: <cftable>

CFML offers several shortcuts to common tasks including <cfselect>, <cfinsert> and <cfupdate>. These tags are quick and require little or no HTML coding skills (or SQL in the case of <cfinsert> and <cfupdate>). Although tempting to use, they do have some drawbacks. We will discuss each of these in later sections.

The first, <cftable>, is fairly limiting. It is fine for a simple table, but since most HTML <table> attributes are not available, it is not possible to adjust cell padding, cell spacing, background color or other table attributes.

Also, it can take slightly longer to process the page since it requires that ColdFusion automatically generate the SQL and HTML code for us.

The following demo is saved as **demos/demo-output-cftable.cfm**. As you can see, in a browser it looks very much like the previous example.



```

<html>
<head>
  <title>All Movies</title>
</head>

<body>
<cfquery name="getMovies" datasource="movieList">
  SELECT movie_id, name, genre, summary
  FROM movies
  ORDER BY name
</cfquery>

<h2>List of All Movies (<cfoutput>#getMovies.recordcount#</cfoutput>
records)
created with CFTABLE</h2>

<cftable query="getMovies" colheaders htmltable border>
  <cfcol text="#name#" header="Movie Name">
  <cfcol text="#genre#" header="Genre">
  <cfcol text="#movie_id#" header="Movie ID">
</cftable>

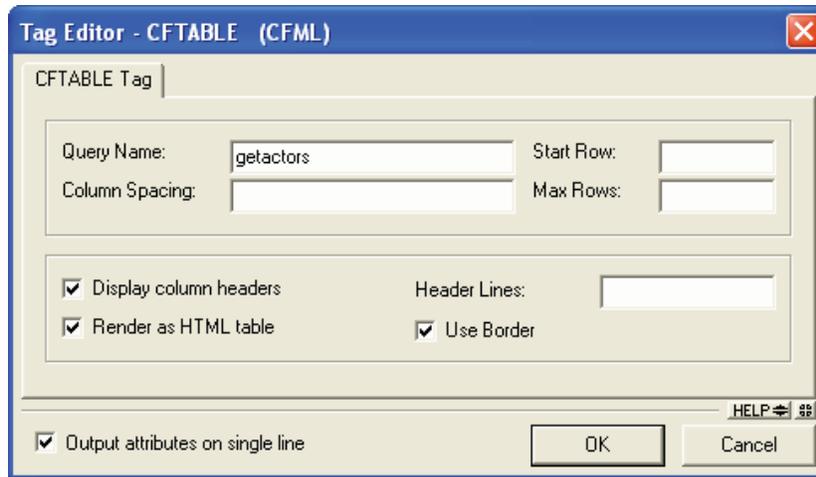
</body>
</html>

```

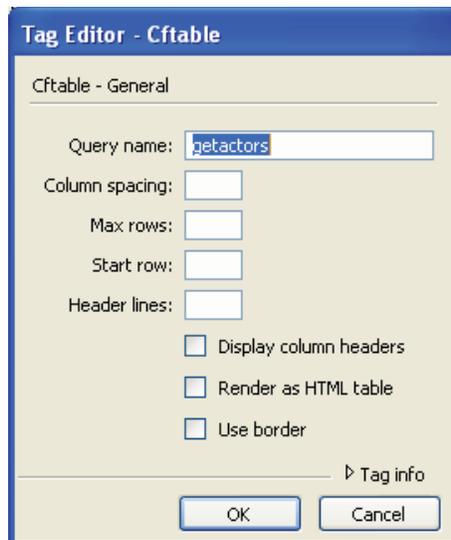
As you can see below, the `<cftable>` tag has a dialog box that can help you with its attributes. If “render as HTML table” is checked, it will generate a table using `<table>`,

<tr> and <td> tags. Otherwise it will use the <pre> tags and will add spaces to line up the columns.

HomeSite+ Dialog Box:



Dreamweaver dialog box:



A <cfcol> tag must be inserted for each column desired. (Its dialog box appears below.) Column Caption holds the text that will be atop the column as a heading. The Column Content will hold the variable and any other static text that will reside in the cell.

Shortcut: Dynamic Table Tool (in Dreamweaver)

Dreamweaver offers a special tool that is used to build a table from the columns selected in a query. On order to use this tool, you must have created a Site. After creating your query, choose Insert → Application Objects → Dynamic Data → Dynamic

Table. After you answer the questions in one small dialog box, Dreamweaver will create all of the HTML tags needed for your table.

Exercise 3: Display Resulting Records manually with <cfoutput>

15 to 20 minutes

In this exercise, you will replace the <cfdump> tag with a more customized display using <cfoutput> block.

1. Open **allmovies.cfm** (from the last exercise)
2. Replace the <cfdump> tag with a <cfoutput> block that displays one <tr> tag (table row) for each movie in the database. Create a <td> tag (table data – or cell) for each name, genre and movie_id. The result should look like this:



Challenge

- Generate a list of actors in an ordered list or a table.
- Can you add the Recordcount as was done in the demo?
- Use the <cftable> tag to create a similar output.
- Replace your table with one created with the Dynamic Table Tool.

Possible Solution to Exercise 3

Saved as **solutions /allmovies-ex-03-done.cfm**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<title>My Movie Reviews - Home</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<link href="mymovies.css" rel="stylesheet" type="text/css">
</head>

<body>
<table width="750" border="1">
  <tr>
    <td colspan="2" valign="top"> <h2 align="center">My Movie
Reviews</h2>
    </td>
  </tr>
  <tr valign="top">
    <td width="175" valign="top">

<!-- Navigation Table -->
<table width="100%" border="0">
  <tr>
    <td><a href="allmovies.cfm">View Movies</a></td>
  </tr>
  <tr>
    <td><a href="allactors.cfm">View Actors</a></td>
  </tr>
  <tr>
    <td><a href="admin.cfm">Admin Section</a></td>
  </tr>
  <tr>
    <td>&nbsp;</td>
  </tr>
</table>

</td>
  <td><p>Movie List:</p>
    <cfquery name="getMovies" datasource="movieList">
      SELECT name, movie_id, genre
      FROM movies
      ORDER BY name
    </cfquery>

    <table width="100%" border="1">
      <tr>
        <td>Movie Name </td>
        <td>Genre</td>
        <td>Movie ID </td>
      </tr>
```

```

        <cfoutput query="getMovies">
        <tr>
            <td>#name#</td>
            <td>#genre#</td>
            <td>#movie_id#</td>
        </tr>
        </cfoutput>
    </table>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p>
    <p>&nbsp;</p></td>
</tr>
<tr>
    <td colspan="2" valign="top">
        <cfinclude template="footer-date.cfm">
    </td>
</tr>
</table>

</body>
</html>

```