Structures and Pointers

Posted At : November 10, 2008 3:48 PM | Posted By : Charlie Griefer
Related Categories: ColdFusion

This is one of those entries that's kind of embarrassing to write, because I'd like to think that after doing this for as long as I have, it wouldn't have been an issue that stumped me for an hour. But it did. And I'd like to think that it may potentially stump somebody else, so let's get it out in the open and try to save somebody else an hour.

I had data that related to 3 distinct entities. Let's say those entities were representations of an employee (a nice textbook-ish example). Employees have names, phone numbers, and departments. I want to store that information in a structure for each employee, and put each structure into an array. Here's how I went about doing that:

```
<cfscript>
 employees  = arrayNew(1);
 employeeStruct = structNew();
 // populate employeeStruct, append to array
 employeeStruct.name  = "Bob Smith";
 employeeStruct.phone = "800.555.1212";
 employeeStruct.dept  = "Marketing";
 arrayAppend(employees, employeeStruct);
 // populate employeeStruct, append to array
 employeeStruct.name  = "John Jones";
 employeeStruct.phone = "800.555.1313";
 employeeStruct.dept  = "HR";
 arrayAppend(employees, employeeStruct);
 // populate employeeStruct, append to array
 employeeStruct.name  = "Frank Wilson";
 employeeStruct.phone = "800.555.1414";
 employeeStruct.dept  = "IT";
 arrayAppend(employees, employeeStruct);
</cfscript>
```

What I expected to see from dumping the employee array was a 3 element array, with each element populated by a unique structure. For each employee (for each employee structure, specifically), I thought I was overwriting the previous values for 'name', 'phone', and 'dept' and appending that struct to the array. But here's what the dump actually displayed:

Notice that the most recent structure (Frank Wilson) was being repeated for all 3 array elements. Why? Wasn't i effectively overwriting the values in the struct for each employee, and subsequently appending that struct (with the new values) to the array?

Well, yes. I was overwriting the values in the struct for each employee. BUT, because structures are shallow copies (e.g. new instances point to the same memory space), for each new employee I wasn't only overwriting the values for that instance of the employeeStruct, but for the previous instances of the employeeStruct that had already been appended to the employee array.

For the first employee (Bob Smith), I populated the employeeStruct with Bob's name, phone number, and department. The struct gets appended to the array. For the second employee (John Jones), I populated the employeeStruct with John's name, phone number, and department. In doing so, I also populated the employeeStruct that was appended to the array with John's name, phone number, and department. Same for the third employee (Frank Wilson). The reference to employeeStruct was shared. Updating one updated them all.

There are a number of ways to resolve this... using duplicate() to create new deep copies of the struct... using structNew() for each employee... the list goes on. But what's important is to understand how structures work and the concept of a shallow copy and pointers.

I thought I did... I've even blogged about it in the past. Yes, I am duly embarrassed. But hopefully this example helps somebody else understand and avoid the same issue.