

XPath is (almost) Perfection

Posted At : August 29, 2008 12:00 PM | Posted By : Steve

Related Categories: ColdFusion

A good friend of mine has recently been educating me on the advantages of **Functional Programming**. As part of that, he has demonstrated some pretty nifty examples of **LINQ** in **C#**. When I first heard about LINQ, I wasn't impressed. Having seen his examples of using LINQ for Functional Programming, however, I now see the advantage.

The consequence of this is that I now find it ugly to have to do loops just to get a subset of data. I want to just be able to describe the data that I want.

I can do this in SQL, but it is a bit harder to do when dealing with most complex data types in CFML. If you are like me, your instinct is to argue that I only believe that because I haven't really looked into all of the fun things you can do with structures in ColdFusion.

That could be the case, but I don't mean the ability to get a structure based on the name or value of a key. I mean the ability to get a structure based on some combination of the name or value of the key and the name or value of one of the descendants of the that key.

One of the things that I really like about structures, however, is that for simple criteria I don't have to do any looping to get a value. If I want to get the key from the "sData" structure where the key is equal to the value of the variable named "mykey", it is easy:

```
<cfset result = sData[mykey]>
```

Of course, if I care about the order then a structure does me no good. An array is better for that. Arrays are nice because I can do an array of structures and get the best of both worlds. Unless, of course, I want to get a the structure with the name of "bob", in which case I am off to cfloop.

XML, however, offers the best of both worlds:

Consider this example:

```
<cfsavecontent variable="result">
<table name="chkExtensions">
<field ColumnName="ExtensionID" CF_DataType="CF_SQL_INTEGER" PrimaryKey="true" Increment="true" />
<field ColumnName="Extension" CF_DataType="CF_SQL_VARCHAR" Length="8" />
<field ColumnName="Rules">
<relation table="chkRules" type="list" field="RuleID" onDelete="Error"/>
</field>
<field ColumnName="NumRules">
<relation table="chkRules" type="count" field="RuleID" join-field="ExtensionID"/>
</field>
</table>
</cfsavecontent>
```

My fields are in order, so I can loop through them and be certain to get them in the order that they are written, but I can also find all of the fields that have onDelete="Error" in their relation.

```
<cfset xData = XmlParse(result,false)>
<cfset aFields = XmlSearch(xData,"//field/relation[@onDelete='Error']/..")>
```

The XPath could probably be written better, but that isn't the point. The point is that with no looping, I could get a key based on the value of its child.

This all seems great, except that it is case sensitive. So, if the XML has onDelete="Error", it doesn't work. Now, I could toss LCase() around it, but in my case, I want to preserve the case on the attribute values, and search for attributes without regard to case - a bit of a problem. It doesn't make for a good

API, if the case of the elements and attributes matters.

Even so, I really like XPath so far. If you want to learn more about XmlSearch in ColdFusion, [Ben Nadel](#) has several great entries on the topic.