WHAT’S NEW IN CF10
FOR CFSCRIPT CODERS?
(UPDATED NOVEMBER 12, 2012)

Charlie Arehart
Independent Consultant

charlie@carehart.org / @carehart
• Rate your frequency of use of CFSCRIPT
  • Never
  • Rarely
  • Occasionally
  • Most of the time
  • Whenever possible
  • Exclusively

• Why do you feel as you do, if on either end of the scale?
• We’ll talk about what’s new in CF10
• We’ll preface that with some intro to evolution of CFSCRIPT
• Also some features new in 10, of interest to CFSCRIPTers
• We’ll end with resources for learning more
• Presentation is online, carehart.org/presentations/
ABOUT CHARLIE AREHART

Independent Consultant
- 15 yrs CF experience (30 in Enterprise IT)
- Certified Adv CF Developer, Instructor
- Adobe Forum MVP, CF CAB member
- Frequent speaker to conf’s worldwide
- Organizer, Online ColdFusion Meetup (coldfusionmeetup.com), 2800+ members
- Living in Alpharetta, Georgia (Atlanta)

Web home at www.carehart.org
- 100+ presentations, 80+ articles, 400+ blog entries
- UGTV: recordings of 600+ presos by 300+ speakers
- CF411.com: 1800+ tools/resources, 150+ categories
- CF911.com: CF server troubleshooting resources
- Hosting courtesy of EdgeWeb Hosting
- Consulting: available for CF troubleshooting, tuning
  - Remote or on-site; on-demand, single instance is ok
EVOLVING SCRIPT-BASED DEVELOPMENT IN CFML

- CFSCRIPT added in CF4
  - has evolved each release since then
- Recent releases: focus on parity with other script languages
  - new operators (like ++)
  - more flexibility in conditionals (> vs GT, for instance)
  - literally the new keyword (for creating CFC instances)
  - Etc.
• Also about creating more script-equivalents to traditional tags
  • Some as statements (used only in CFSCRIPT)
  • Some as functions (used in both script and tags)
  • Some as CFCs (several related methods for a feature)
• Grumbling at 3 approaches? More later
WHAT’S NEW IN 10?

• Support for colon separators in structures (key:value pairs)
• Enhanced script looping (for-in) over queries and Java arrays
• Support for new tag equivalents in script statements
  • (setting, cookie)
• Support for new tag equivalents as functions
  • (invoke)
• Support for new tag equivalents as CFCs
  • (collection, index, search)
• And more
SUPPORT FOR COLON SEPARATORS IN STRUCTURES (KEY:VALUE PAIRS)

- Creating a structure named student, with 3 keys, one being an array:
  <cfscript>
  student = {firstName:"Jane", lastName:"Doe", grades:[77, 84, 91]};
  writeDump(student);
  </cfscript>

- Such implicit structure and array definition added in CF 8
  - Curly braces for structure, square braces for arrays

- Approach similar to other scripting languages

- Also supported:
  - <cfset student = {firstName:"Jane", lastName:"Doe", grades:[77, 84, 91]}>
  - Not supported: <cfset name:”Jane”> or name:”Jane”;
ENHANCED SCRIPT LOOPING (FOR-IN) OVER QUERIES AND JAVA ARRAYS

- ColdFusion has long been able to use the for-in statement to loop over structures
  - And ColdFusion 9.0.1 added the ability to loop over CFML arrays
  - Now can loop over query results
- Example:

```cfml
<cfquery name="arts" datasource="cfartgallery">
    select artName, price from art
</cfquery>
<cfscript>
    for(row in arts) {
        writeoutput(row.artName & ' ' & row.price & '<br>');
    }
</cfscript>
```
ENHANCED SCRIPT LOOPING (FOR-IN) OVER QUERIES AND JAVA ARRAYS (CONT.)

• Closer look:

  for(row in arts) {
      writeoutput(row.artName & ' ' & row.price & "<br>");
  }

• *row* declares a variable holding reference to each row
  • as statements within braces ({}) loop over each record in query

• If done within function, could/should use *var* keyword
  • as in *for (var row in arts)*

• Column reference is *row.col*, not *query.column*
  • *row[columnname]* is supported, of course

• Currentrow accessed for query, not row: *arts.currentrow*
ENHANCED SCRIPT LOOPING (FOR-IN) OVER QUERIES AND JAVA ARRAYS (CONT.)

• What about script-based query? When will we ever do that?
  • Can do since CF 9!

• Example:
  
  <cfscript>
  arts = new query(datasource="cfartgalle ry",sql="SELECT artName, price FROM Art" ).execute().getResult();
  </cfscript>

• More on CFC-based tag equivalents later
FOR-IN FOR JAVA ARRAYS

• Various Java methods return java arrays
  • Can now loop over those in CFSCRIPT

• Example:

```cfs
<cfscript>
cookiearray = getpagecontext().getrequest().getcookies();
for (acookie in cookiearray) {
    WriteOutput(acookie.getname() & '=' & acookie.getvalue() & '<br>');
}
Writedump(cookiearray);
</cfscript>
```
SUPPORT FOR NEW TAG EQUIVALENTS IN SCRIPT STATEMENTS

- Two new statements supported in CF10
  - *setting* statement (for CFSETTING)
  - *cookie* statement (for CFCOOKIE)
SUPPORT FOR NEW TAG EQUIVALENTS IN SCRIPT STATEMENTS: SETTING

- CFSETTING can be used to set requesttimeout, enablecfoutputonly, showdebugoutput
  - Any can be set in CFSCRIPT now
- Example:
  
  ```
  <cfscript>
  setting requesttimeout=20 enablecfoutputonly=true;
  </cfscript>
  ```
- Especially valuable for use in cfscript-based application.cfc
  - Which is also new in CF9
- Hidden Gem in CF10: can set requesttimeout=0
  - Turns off any requesttimeout set in Admin or previously in request
SUPPORT FOR NEW TAG EQUIVALENTS IN SCRIPT STATEMENTS: COOKIE

- CFCOOKIE can be used to set cookies
  - Can now set in CFSCRIPT
  - Could always set as cookie.name="value", but now can set all properties

- Example:
  
  ```
  <cfscript>
  cookie.firstname = {value="Charlie",expires="10");
  writedump(cookie.firstname);
  </cfscript>
  ```

- Note approach of setting props in a structure
  - Could also do as cookie.firstname = {value:"Charlie",expires:"10");

- Properties that can be set:
  - value, expires, domain, path, secure, and httponly
  - CF10 provides two new props (attributes for CFCOOKIE): preservecase and encodevalue
SUPPORT FOR NEW TAG EQUIVALENTS AS FUNCTIONS : INVOKE

- Can now invoke CFC/web service methods using invoke()

- Assume CFC called demo.cfc:

```cfc
component {
  function helloWorld() {
    return "Hello World";
  }
}
```

- Can then call it using the `invoke` function:

```cfs
<cfscript>
retval=invoke("demo","helloWorld");
writeOutput(retval);
</cfscript>
```
SUPPORT FOR NEW TAG EQUIVALENTS AS FUNCTIONS : INVOKE (CONT.)

• Passing in arguments: must do as structure in 3rd arg

  • Assume new method in CFC:
    
    function helloFriend(string name) {
      return "Hello " & arguments.name;
    }
  
  • Can call as:

    <cfscript>
    retval=invoke("demo","helloFriend",{name="Charlie"});
    writeOutput(retval);
    
    </cfscript>

• That 3rd argument structure could of course hold within it arrays, structs, etc.
SUPPORT FOR NEW TAG EQUIVALENTS AS FUNCTIONS: INVOKE (CONT.)

• Reusing a saved CFC instance is often more appropriate

• Example:

```javascript
helloInstance=createobject("component","demo");
retval=helloInstance.helloWorld();
```

• Can also use a middle ground:

```javascript
helloInstance=createobject("component","demo");
retval=invoke(helloInstance,"helloWorld");
writeOutput(retval);
```

• Or more simply:

```javascript
writeOutput(helloInstance.helloWorld());
```
SUPPORT FOR NEW TAG EQUIVALENTS AS FUNCTIONS : INVOKE (CONT.)

• When invoking a web service, you MUST use that approach of reusing a saved instance

• Example:

  <cfscript>
  retval=invoke(tempSvc,"GetCityWeatherByZIP",{zip="30005"});
  writeOutput("Temp=' & retval.GetTemperature());
  </cfscript>

• Or could replace middle line with following (3rd arg then NOT struct):

  retval=tempSvc.GetCityWeatherByZIP(zip="30005");
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S

- 3rd and final form of tag equivalent is CFCs
  - Saw use of query CFC earlier

- CF10 adds CFC-equivalents for CFCOLLECTION,CFINDEX, and CFSEARCH

- CF9.0 had added ftp, http, mail, pdf, query, and storedproc

- CF9.0.1 added dbinfo, feed, imap, ldap, and pop
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S: CFCOLLECTION

• CFCOLLECTION is used to create/manage Solr collections

• Simple example (list collections), in tags:

  <cfcollection action="list" name="myCollections">
  <cfdump var="#myCollections#">

• And with script (using new keyword to create CFC instance):

  <cfscript>
  myCollections=new collection().list();
  writeDump(myCollections.getresult().name)
  </cfscript>

• Method name equates to ACTION value from tag
• Note also chaining the list method in the first statement
  • Note also that the object returned has a getresult method & name property
• Be sure to have started “CF10 Jetty Service” which runs Solr
• Can create a Solr collection
  • Location can be a directory anywhere on server

• Simple example (list collections), in tags:

  `<cfscript>
  newCollection=new collection();
  newCollection.create(collection="merchandise",path="c:\coldfusion10\Collections\");
  `<cfscript>

  • Or could do more simply on one line, of course:

    `new collection().create(collection="merchandise",path="c:\coldfusion10\Collections\");`

• What’s a negative of either approach, though?
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S : CFCOLLECTION (CONT.)

• We might want to issue many requests against the CFC instance
  • Could choose to save instance, as discussed before

• And note that we were passing args into the method
  • Could also pass args into the instantiation of the CFC instead

• Example producing same result:
  • `<cfscript>
    • newCollection=new
collection(collection="merchandise",path="c:\coldfusion10\cfusion\Verity\Collections\");
  • newCollection.create();
  • `<cfscript>

• In both cases, result is a new folder in the named location
  • Solr “index” is a collection of folders and files in that location, ready to populate
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S : CFCOLLECTION (CONT.)

• Other available methods in collection.cfc

• Delete:

  ```
  new Collection.delete(collection="merchandise");
  ```

• Also:

  • map, optimize, and categoryList (last one new for Solr)

• See docs for CFCOLLECTION to learn more on these

  • Will also discuss other available resources for learning more about these
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S: UNDERSTANDING ERRORS

• Errors from these CFC-based tag equivalents not always clear

• Consider this:

```cfscript
new Collection.delete("merchandise");
</cfscript>

• Error is:

  Attribute validation error for tag CFCOLLECTION.
  It does not allow the attribute(s) 1. The valid attribute(s) are ACTION, COLLECTION, LANGUAGE, PATH.

• Problem is what?
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S: CFINDEX

- Now we have a collection, we can populate it, with:
  - documents (text files, PDFs, MS Office-format documents, static web pages, and more)
  - query results (from CFQUERY, etc.)
- Then we can search it with CFSEARCH tag or search.cfc
- This talk is not really “about Solr”, so will not delve deeply
  - See docs for more on when/why/how to use Solr in various ways
- Let’s first grab some data to index:(using OWS dsn, from forta.com)
  <cfscript>
  q = new query(datasource="ows");
  merch=q.execute(sql="select merchid, merchname, merchdescription from merchandise").getResult();
  </cfscript>
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S: CFINDEX (CONT.)

• Could now index that with CFINDEX, as in:

```xml
<cfindex action="update"
    collection="merchandise"
    query="merch"
    key="merchid"
    type="custom"
    title="merchname"
    body="merchdescription">
```

• Or in script (with one gotcha to beware, and one other difference):

```javascript
m=new Index();
m.update(collection="merchandise",query=merch,key="merchid",type="custom",title="merchname",body="merchdescription");
```
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S: CFINDEX (CONT.)

- Other methods:
  - delete (to remove one or more items from the index)
  - purge (to remove all items from the index)
- ColdFusion 10 adds other CFINDEX actions
  - abort, deltaimport, fullimport, and status
  - Currently no corresponding methods for these in the index.cfc
    - Will discuss later how you can solve that easily if you want
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S : CFSEARCH

• CFSEARCH is used to search Solr collections
  • Much we could discuss: defining search criteria, search keywords, what parts of an index record to search, and much more
  • See the ColdFusion documentation for more

• Simple example
  • search the titles of records (title was set to a query column above)
  • To do with CFSEARCH, could do:
    <cfsearch collection="Merchandise"
      name="merch"
      criteria="title:ColdFusion"
      maxrows="10">
    • Note use of MaxRows to limit results
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S : CFSEARCH (CONT)

• Could dump the result:

```cfdump var="#merch#" show="score,title,summary">
• Note use of SHOW attribute, added in CF8
  • one of many CFDUMP enhancements, including also TOP, and more
• This would appear as follows:

<table>
<thead>
<tr>
<th>query [Filtered - 3 of 15 columns shown]</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCORE</td>
</tr>
<tr>
<td>1 1.2331685</td>
</tr>
<tr>
<td>2 1.2331685</td>
</tr>
</tbody>
</table>
SUPPORT FOR NEW TAG EQUIVALENTS AS CFC’S: CFSEARCH (CONT)

• Script equivalent:

  <cfscript>
  merch=new search().search(collection="merchandise",criteria="title:ColdFusion",maxRows=10);
  writeDump(var=merch.getresult().name,show="score,title,summary");
  </cfscript>

• Though note again need to use merch.getresult().name

• Note also, new search().search(), may look awkward
  • there’s no ACTION=”search” on the CFSEARCH tag
  • so no action value to use as method name. Adobe just picked search
  • Same in CF9 CFCs, such as query().execute(), mail().send(), http().send()

• Note also use of name= on writedump, since we have add’l args to the dump
SOME FINAL COMMENTS ON SOLR/CF

• For more in general, see
  • See Developing ColdFusion 10 Applications, Chapter 7: Accessing and Using Data”, section “Solr search support”

• For problems, see special logs related to Solr
  • [ColdFusion10]\cfusion\jetty\logs

• May also want to do more with Solr than CF tags/CFCs allow
  • Can call via CFHTTP (as Solr runs as separate service from CF)
  • See tutorial by David Faber (among resources from others)
    • http://www.thefaberfamily.org/search-smith/coldfusion-solr-tutorial/

• What about crawling a site?
  • See http://nutch.apache.org/
  • Could define a collection in ColdFusion pointing to collection created and populated by Nutch
  • Then use the Solr searching capability to query it
ABOUT THESE ADOBE-PROVIDED CFCs

• Docs
  • *ColdFusion 10 CFML Reference*, “Script Functions Implemented as CFCs”
  • *Developing ColdFusion 10 Applications*, “Service tags with bodies”
    • Yes, docs call these “Service tags with bodies”, as well as “Script Functions Implemented as CFCs”

• Viewing the CFML source for these CFCs
  • [/ColdFusion10]/cfusion/CustomTags/com/adobe/coldfusion
  • See also [/ColdFusion10]/cfusion/wwwroot/WEB-INF/cftags/META-INF/taglib.cftId

• Viewing the component metadata, methods, and more
  • http://[/servername]/CFIDE/componentutils/cfexplorer.cfc?method=getcfcinhtml&name=com.adobe.coldfusion.collection
ABOUT THESE ADOBE-PROVIDED CFCS (CONT)

• Community-contributed CFC-based tag equivalents
  • https://github.com/CFCommunity/CFScript-Community-Components
  • 3 in CF10 came from here, written by Ray Camden (project encourages that)

• Others there of interest
  • content, execute, flush, header, htmlhead, loginuser, logout, registry, schedule, spreadsheet, wddx, and zip

• And if you want more, or want to change, go for it!
ABOUT THE THREE APPROACHES TO TAG EQUIVALENTS

• Do you wonder: Why as *setting* done as a statement? And *invoke*, as a function? And *collection*, as a CFC?
  • Is it purely arbitrary, and illogical?

• I’d say no
  • Statement does one thing (setting, cookie, etc.)
  • Function does different things based on arguments (invoke, etc.)
  • CFC is a collection of related functions, with properties (solr cfcs, etc.)

• Some might with same was done for all the image* or spreadsheet* functions, etc.
OTHER CF10 FEATURES OF INTEREST

• Many other CF10 changes, not specific to CFSCRIPT
  • But of interest to script-based developers

• Method chaining for ColdFusion components (CFCs)
  • as well as implicit constructors/notation for CFCs

• Enhancements to queryNew and queryAddRow functions

• New functions like structeach, arrayeach, arrayslice, etc.
  • (related to closures, but useful separately)

• New imageCreateCaptcha function

• and still more

• See docs, my Hidden Gems in CF10 talk, blogs, for more
LEARNING MORE

• Many assume there is little documentation of script coding
  • Don’t rely ONLY on the CFML Reference!

• See also *Developing 10 ColdFusion Applications*
  • 32-page Chapter, “Extending ColdFusion Pages with CFML Scripting”

• But CFML reference also has chapters on
  • CFSCRIPT
  • “Script Functions Implemented as CFCs”
SUMMARY

• So sure, only “4 new things about CFSCRIPT”
  • But lots to know and consider
• And also might give you motivation to study similar script-related things changed in CF9
  • Enjoy!
• Remember: the presentation is online
• And let me know if I can ever help with CF server troubleshooting, tuning, configuration, etc.
  • Or considering how to use CF, CFBuilder features