

Journeyman ColdFusion

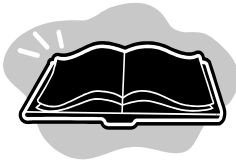
A New Section of "Working Class"
CFUG Topics

Charles Arehart
carehart@systemanage.com

February, 1999

"Journeyman" Defined

- From Merriam-Webster



an experienced reliable
worker or performer,

especially as
distinguished from one
who is brilliant

Purpose of Journeyman Topics

- News you can use, today, in any application
- Less advanced topics than normal CFUG
"bleeding edge" topics
- Still offering doctorate-level topics in second
half of meeting

- Will be offered as new section at DC CFUG

This Month's Topics

- Getting the Beta of 4.0.1 (it's open to all)
- Deleting a Session Variable in 4.0
- Mailing to More Than 40 People in 4.0
- Top Ten Studio Customizations
- Using Access "Stored Procedures" in 4.0

Getting the Beta of 4.0.1

(it's open to all)

About the Beta

- New versions of App Server, Studio, and HomeSite
 - for server, includes both Enterprise and Professional editions
- Primarily a maintenance release to fix bugs
 - but several new features, discussed changes last month
 - see release notes at beta site (also installed with product)
 - http://beta.allaire.com/productinfo/cfasnt401b2/allaire_support/cf4newfeatures.htm
- Currently in Beta (just released Beta 2)
- Beta is open to all current users
 - must be licensed for product being updated
- Beta 1 was released on 12/17

Obtaining the Beta

- Beta at <http://beta.allaire.com>
 - username: cf401
 - password: content
 - (these are offered right from a public link at the top of the developer forums page, so they're not a secret)
- Need your product serial number in order to download
 - is on the "Version Info" page of the your ColdFusion Server Administrator for Server
 - in the Help About box for Studio
- Report problems with beta through beta channels
 - beta@allaire.com or beta.allaire.com

Some Beta Observations

- Besides new features, also some new language changes to be aware of that may cause code to fail:
- Erroneously using an "=" rather than an "EQ" in a boolean statement in a CFIF tag
 - was not being caught in 4.0 as it was in 3.1.1.
 - this now works as it did in 3.1.1.
 - Mention this because the "reindex" feature of the beta's online docs has this mistake and fails ([CFDOCS/searchmain.cfm](#))

Some Beta Observations (cont)

- The NAME attribute of the CFAPPLICATION tag is now required
 - so that you can have two separate applications with different sessions for the same user
 - if you don't use the NAME attribute you'll get an error in your template that the name attribute was not supplied
 - Mention this because the example app on "dynamic content publishing" has a failure in its admin screen due to this.
- Watch for these in your own apps as well

Beta 2 Changes

- Previous topics are from Beta 1. Beta 2 just came out
 - New TYPE attribute for CFLOCK
 - readonly/exclusive
 - Enhanced scheduling engine
 - New performance monitor configuration file
 - ...and more

Studio Beta 2 Enhancements

- Query Builder
 - Significant performance enhancements were made to the Query Builder, especially when working in databases with a large number of tables. Support was added for Oracle synonyms, as well.
- New set of HTML and VTML tag editors
 - A new set of HTML and VTML tag editors and tag definitions from Marjolein Katsma was integrated into the product. As a result of this, forty-five HTML tag dialogs were improved and 44 new tag editors were added to the product. The functionality of the Tag Inspector and Tag Insight was also improved as a result of this change.

Deleting a Session Variable in 4.0

New Capability in 4.0

- Can now delete session (as well as application) variables
 - These can now be manipulated as "structures" in 4.0
 - There is a structdelete function, whose format is:
 - `<cfset foo = structdelete(session,"test")>`
 - note: scope is not quoted, variable name is
 - This will do more than set the variable to null, but will delete it entirely
 - `isdefined(session.test)` will return false
 - In case of session variable, deletes variable for current user

Example

- `<cfapplication sessionmanagement="yes" name="testapp">`
`<cfoutput>#isdefined("session.test")#</cfoutput>`
`<cfset session.test = 'x'>`
`<cfoutput>#isdefined("session.test")#</cfoutput>`
`<cfset x = structdelete(session,"test")>`
`<cfoutput>#isdefined("session.test")#</cfoutput>`
- Yields:
 - False, True, False indicating the variable is created and then deleted

Additional Uses

- This approach also works for
 - application variables
- But interestingly not for
 - server or client variables
 - nor local ("variables"), query, or form variables
- Attempt to reference these as structures, as in:
 - `<cfset foo = isstruct(client)>`
 - yields a CF error saying the variable "client" is undefined

Mailing to More Than 40 People in 4.0

Not Quite the Limitation it Seems

- CF 4's Release Notes indicates:
"The CFMAIL tag currently supports a maximum of 40 recipients per mail message. An error message is generated if this number is exceeded. This limitation will be addressed in a future release."
- This can be misinterpreted
 - The limit is 40 recipients listed as the recipients of a single message, as in addresses listed in its TO, CC, and BCC fields
 - It DOES NOT refer to using CFMAIL QUERY=... and listing a table column to identify the recipient. In that case, there is no limit

Example

- Following is OK
 - `<CFMAIL FROM="carehart@figleaf.com" QUERY="mylist" TO="#email#" ...`
 - this will send a copy of the email to as many records as were found in the query referenced as "mylist", using the value of the "email" column in that query
- Following will be limited to 40 recipients
 - `<CFMAIL FROM="carehart@figleaf.com" TO="joeblow@figleaf.com, monica@figleaf.com, oj@figleaf.com, deon@figleaf.com, mj@figleaf.com..."`
 - it seems that the sum of the number in the to, cc, and bcc fields is what must be less than 40

Top Ten Studio Customizations

Molding Studio to Fit Like a Glove

Studio is Very Flexible

- Many features available to customize
 - Options>Settings
 - 14 tabs within this command
 - Options>Customize
 - 3 tabs here
 - Toolbars, keyboard shortcuts can all be moved, modified, and added to
 - Code reuse supported via snippets, shared snippets, file templates, code templates
- Customizing Studio can make significant difference in productivity
- Following are the customizations I make on any new Studio install

Top 10 Customizations...

- Turn on "Allow undo after save"
 - allows undo's even when a file is saved (as long as not closed in Studio)
 - *in options>settings>edit*
- Turn on a "development mapping" for the webroot
 - this will be useful both for debugger and for using browse button (Studio's internal browser)
 - *in options>settings>browse>development mappings*
 - set at least one for localhost if cf server installed on your workstation
 - click file find icon to right of "studio path", browse to your webroot
 - Studio will fill in next two lines for you, select OK

Top 10 Customizations... (cont)

- Turn off "Lowercase all inserted tags"
 - will cause Studio to upper case all inserted tags (that it inserts for you)
 - *in options>settings>html*

Top 10 Customizations... (cont)

- Turn on "show line numbers in gutter" (as well as "show gutter")
 - will cause Studio to show line numbers, to help debug reported errors
 - *on Studio's editing interface, in the "editor toolbar"*
 - between the resource tab and the text entry area
 - click the icons with a pound sign as well as the one above it
- Turn on "Show keyboard shortcuts in toolbar hints"
 - will display keyboard shortcuts when mouse is placed over various toolbars
 - *options>customize>keyboard shortcuts (checkbox on screen)*

Top 10 Customizations... (cont)

- Change keyboard shortcut for comments (ctrl+shift+M) to be CFML comments
 - make the default a CFML comment rather than an HTML one
 - *options>customize>keyboard shortcuts, find "cold fusion comment" in list of shortcuts*
 - place cursor in or next to "apply" button, press the keys ctrl+shift+f8
 - notice that the string "ctrl+shift+f8" appears in the area, press apply

Top 10 Customizations... (cont)

- Change keyboard shortcut for F4
 - change F4 to not call up the Tag Inspector (last tab in "resources tab")
 - This can be frustrating if hit by mistake (and easily done if meant to hit F3 for "find next")
 - It's easy enough to call it up via mouse clicks
 - Consider just turning off any keystrokes for it
 - *options>customize>keyboard shortcuts, find "tag inspector" in list*
 - and hit backspace in the "apply" input area mentioned above

Top 10 Customizations... (cont)

- Add "Script toolbar" to access javascript tools
 - displays useful document object model tree
 - *right click on toolbar, choose "script" if not checked*
 - drag and drop onto toolbar area (upper right of screen)
- Add "javascript tree" to script toolbar
 - shows document object model attributes, methods, properties, etc.
 - *select "script" toolbar, right click on toolbar, choose "customize" (at bottom of list)*
 - ensure "script" toolbar appears as the toolbar being customized, or select it in the toolbars list at right
 - choose "javascript tree" from "toolbuttons"
 - drag onto toolbar shown in dialogue

Top 10 Customizations... (cont)

- Selecting a preferred toolbar to be "always on"
 - the tool buttons at top left of screen (cut/copy/paste/undo, etc.) are actually the "edit" toolbar
 - if you don't use those often but do more often use a toolbar you're always having to select from the "quick bar" (the tabbed list of tables/frames/lists/cfml basic, etc. toolbars), you can swap the preferred toolbar for the edit one
 - *drag left edge of edit toolbar (just left of scissors icon) onto the quickbar.*
 - This will dock it as another tabbed toolbar (as described above). If it doesn't dock there, double click it.
 - select the tab for your preferred toolbar, then select ITS left edge and drag it onto the area under the menus (where the edit bar had been). It will dock itself there.
 - Note that you can actually drag another one and place it below that one

Might Also Consider

- If you like “tag insight”, set its response speed to 0 seconds
- If they get in your way, turn off “tag completion” and “tag automation”
- Turn on “word wrap”
- Unless you work solely with html (not CFML), turn on “Disable (hide) design tab”
- Turn on “Dynamically refresh file list when changes occur in current directory”
- Turn off “Show Splash Screen at startup”

These Should Be Enabled

- Ensure the following are on
 - These should be on when Studio is installed
 - You'll want to be sure no one has turned them off on your system
 - Auto indent
 - causes new line in studio to be indented at same level as current line
 - *in options>settings>Edit*
 - Use MS IE as the internal browser
 - allows use of browse command (F12) in studio to call up IE within studio
 - *in options>settings>browse*

Rolling Your Own

- Keyboard Shortcuts
 - Assigning your own
 - Overriding existing ones
 - Reviewing current ones
 - *options>customize>keyboard shortcuts*
- Code Snippets
 - Custom text to drop in, selected from list
 - can setup to wrap around selected text
 - *click “snippets” button at bottom right of resource toolbar*
 - opens blank window (at first)
 - right click, select “create folder”
 - right click on folder, select “add snippet”
 - to use, double click
 - can assign keyboard shortcut
 - *options>customize>snippets shortcuts*

Rolling Your Own (cont)

- Code Templates
 - similar, but not restricted to keyboard shortcuts
 - used by typing in an abbreviation, and pressing CTRL-J
 - several are predefined, can define your own
 - *use options>settings>code templates*
 - can review existing ones, add a new one
 - note that inserting the pipe (|) character specifies where to place cursor when text is inserted

Studio Summary

- Very customizable
- Various settings to control environment
 - indents, undo after save, development mappings, and more
- Modifiable toolbars
 - can add, change, rearrange, and even add new things (javascript tree)
- Tools to facilitate use of frequently used things
 - keyboard shortcuts, snippets, code templates

Using Access "Stored Procedures" in 4.0

Using Access "Stored Procedures" in 4.0

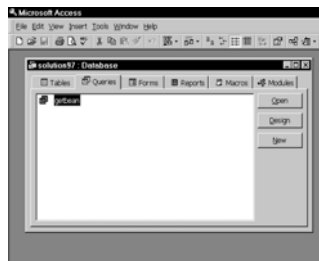
- New stored procedure features in 4.0
- Not intended for use with Access, really
 - but we have found that they do work, and add functionality previously missing
- Opens door to using "stored procedures" approach with Access databases
 - Several benefits both in performance and productivity

Background

- Access doesn't support stored procedures, per se
 - but does support notion of "queries"
- Queries can hold any normal SQL query you'd put in CFOQUERY
 - can even define input parms that are filled at run-time
- Pre-4, could call a "query" using ODBC {Call} statement
 - but could not pass in values for input parms
 - severely limited usefulness of these for CF applications

About Access Queries

- Queries are defined similarly to tables
- Can create with "new" command
 - edit with "design"
 - run with "open"
- Can create using visual tools similar to Studio query builder
- We'll instead show creating one in SQL



Defining an Access Query (cont)

- Use New>Design View, close the table selector
- Use View>SQL View to enter/modify SQL
- Here, we can enter SQL directly (copying what we had in a CFCQUERY)
- Surround input variables in square brackets

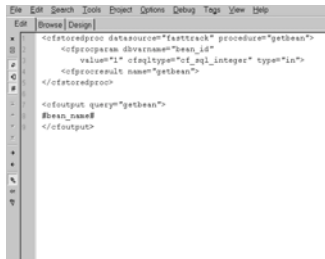


Calling Access Queries in CF

- 3 new stored procedure (SP) tags in 4.0
- CFSTOREDPROC
 - names the sp to be executed
 - or query in our example
- CFPROCPARAM
 - identifies parms to be passed in/out, their type, etc.
 - can list several of these
- CFPROCRESULT
 - identifies result set to be passed out
 - some db's support multiple result sets

CF Code Sample: Calling Access Query as an SP

- Note that CFPROCPARAM and CFPROCRESULT are subtags to CFSTOREDPROC
- Manual says "type" is not required for CFPROCPARAM
 - should default to "in"
 - my testing shows it to be required even so
 - in both 4.0 and 4.01]

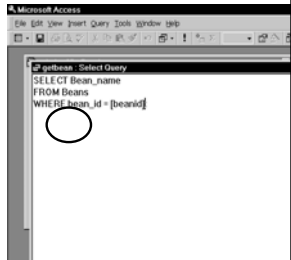


Things to Keep in Mind

- Does not open up full range of Transact SQL as typically used in Stored Procedures
 - Can't use conditional logic within query, or create temp tables
 - Can simply enter the same SQL you would in a CFQUERY
 - May be possibilities we've not explored
- Be sure to enable use of SP's on the datasource definition (off by default, in "CF Settings" for datasource in CF administrator)
 - Difficult to debug errors, as SQL is not displayed in CF debug output
 - Cannot make table name variable, nor it seems, column names
- Might work for calling Access "modules". Haven't tried it yet.

Naming Input Variables in Query


- Name of input variable should not be same as name of column
 - Access will interpret as a column name
 - In our example, if we did not change it, would result in a WHERE of:
 - bean_id = bean_id
 - Changing to beanid solved problem
- Can now run, and will prompt for value of beanid



Benefits of Using SP's in Access

- Cross-platform development
 - Can develop/test against Access, run in prod against SQL Server
- Modularization of code
 - Can store common queries in database rather than repeating in several templates
- Improved processing due to reduced ODBC overhead
 - have seen up to 8 times reduction in query processing time
 - for very minor query, can have slight overhead penalty





Recap of Topics

- Getting the Beta of 4.0.1 (it's open to all)
- Deleting a Session Variable in 4.0
- Mailing to More Than 40 People in 4.0
- Top Ten Studio Customizations
- Using Access "Stored Procedures" in 4.0
