MIGRATING APPS TO COLDFUSION 2021
FROM EARLIER VERSIONS

Charlie Arehart, Independent Consultant
CF Server Troubleshooter
charlie@carehart.org
@carehart (Tw, Fb, Li, Slack, Skype, GitHub)

Updated June 17, 2022
TOPICS

- Compat issues introduced in CF2021
- Compat issues due to CF2021 updates
- Compat issues introduced in CF2018
- Compat issues introduced in CF2016
- Compat issues introduced in CF11
- Compat issues introduced in CF10
- Migration tools and resources
How easy will such a migration be?
- Of course, will depend on how many CF versions back you’re jumping from
- Either way, for some the move may be trivial
  - for others it may seem difficult
  - For most, it will be somewhere in the middle

What we won’t cover
- What language features are new/changed in CF2021, and each release
  - See Ben Nadel post on languages changes in recent releases: bit.ly/3BGso02
  - See also my “Hidden Gems” talks for each release, at carehart.org/presentations
    - I cover language changes, as well as install, admin changes, underlying library version changes, OS and DB support, licensing/pricing, differences between Ent & Std, and more
I focus on CF server troubleshooting, as an independent consultant
  - Assist organizations of all sizes, experience levels
  - Work remotely 99% of the time, safe, secure, easy (via shared desktop)
  - **Solve most problems in less than an hour**, teaching you also as we go
  - **Satisfaction guaranteed**
  - More on rates, approach, etc at carehart.org/consulting

But to be clear, I’m not selling anything in this session! 😊
  - Just sharing my experience, and that of others
Before sharing specifics of each version, there are generic matters, such as:

- A new CF version may add **a new keyword, scope, or function**
  - Your existing code using that already may fail, not always in an obvious way
  - Could be code in some framework you use. For instance:
    - CF2018 added keywords like final, abstract, used previously as vars in ColdSpring
    - CF2016 u7/CF2018 u1 added getCanonicalPath, used previously as var in Fusebox
COMPAT ISSUES INTRODUCED IN CF2021
Dateformat mask of capital “D” now honored as meaning “day in year”
  ▶ Previously, it was no different than lower-case “d”: for day in month
  ▶ Huge difference, if you may have code using D or DD, which will now produce very different result

Example: dateformat("11-23-20","MM-DD-YY")
  ▶ In 2018 and before, produces 11-23-2020
  ▶ In 2021, produces instead 11-328-2020

Example: bit.ly/3m3sLdR

bit.ly/3m3sLdR
carehart.org/blog/client/index.cfm/2020/11/24/breaking_change_in_cf2021_dateformat_D_vs_d
...
Available fix made available in late Nov 2020, then folded into CF2021 Update 1 (in Mar 2021)

- Requires you to add a JVM argument to REVERT back to regarding capital “D” same as “d”

Or here’s a regex to search for using an editor or search tool:

- `l?s?\(dateformat \| datetiemformat\)\(.*\)["'\]\(.*\)\(DY\)\{1,10\}\(.*\)["'\]`
- `(\?i)date\(\?i\)\(.*\)(DD\/)\(.*\) \(\?i)date\(\?i\)\(.*\)(DD\/-)\(.*\) \(\?i)date\(\?i\)\(.*\)(/DD\/)\(.*\) \(\?i)date\(\?i\)\(.*\)(D\/)\(.*\) \(\?i)date\(\?i\)\(.*\)(D\/-)\(.*\) \(\?i)date\(\?i\)\(.*\)(/D\/)\(.*\)`

- carehart.org/blog/client/index.cfm/2020/12/7/cf2021_hotfix_datemask_compatibility_issue
OLD FEATURES REMOVED IN CF2021

- Removed:
  - Corba integration
  - LCDS integration
  - Flex integration
  - Flashpaper type for cfdocument, cfpresentation
  - cfschedule onmisfire="fire_now"

- Seemingly nothing deprecated or “retired” (as we will see in others)
“static” is now a reserved word, at least as variable name in a CFC
  ▶ Will get error, “Cannot Assign Value to variable named static”
  ▶ To allow it, add staticSupport="false" to component definition
  ▶ Problem does not apply to use of static as struct in key, nor as a variable in a cfm
  ▶ See adobe.ly/3k4KFOm

Bonus topic (I will add one of these for each version, a tip that may help you)
  ▶ CF2021 has lifted Enterprise requirement for CF Security Code Analyzer
CF2021 update 2 (and CF2018 update 13) came out in Sep 2021
- Both introduce a bug in query of queries
  - tracker.adobe.com/#/view/CF-4212383
- Fixed with a special hotfix jar, available from Adobe (hf202100-4212383.jar)
  - Separate one available for CF2018
  - Fix is due to be included in next major CF update
COMPAT ISSUES INTRODUCED IN CF2018
As part of the new OO support: **abstract, function** and **final**
  - And **null**, if null support enabled

If you have code using those as variables, etc., will have conflicts/errors
  - You may think you don’t use these, but frameworks might
  - ColdSpring does use both abstract and function as variables (as in coldspring\aop\framework\AopProxyUtils.cfc)

You must carefully find and replace those variable names, properties
  - May not want to do global replace, as you may have text that shouldn’t be replaced
  - Could just run through code and fix those which get errors, but that’s risky, too
**CF_SQL... ABBREVIATION CHANGES**

- `cfqueryparam/cfprocparam` fail if undocumented `cfsqltype` abbreviation used
  - `CF_SQL_int` should instead be `CF_SQL_integer`  
  - `CF_SQL_DateTime` should be `CF_SQL_date`, _time, or _timestamp  
  - `CF_SQL_Num` should be `CF_SQL_Numeric`  
  - And so on  
  - This applies also to queryexecute use of queryparams, for instance:
    - `queryexecute ("select * from art where artid=:id",{id="1", cfsqtype="CF_SQL_Integer"},{datasource="cfartgallery"})`

- Before this, such invalid values were ignored, treated as `cf_sql_string`
- These changes were also rolled into CF2016 as of its last update, 17, in Mar ‘21
Some items removed:

- `cfschedule's timeout` attribute (used in creating a new task)
  - can add `cfsetting requesttimeout` to page being called (if under your control)
- `cfcache's cachedirectory` and `timeout` attributes
- `cfcollection's map` and `repair` values for `action` attribute
- `cferror's monitor` and `option` values for `type` attribute

Some features "retired" (not enabled by default)

- Most rarely used (flash remoting, flash-based UI elements, xml forms, blazeds, lcsds, AIR, CF actionscript functions, CORBA, portlets, etc.)
Also “retired” are UI tags based on YUI javascript library, used by tags like:

- `cfmenu`, `cftree`, `cftooltip`, `cfcalendar`, `cfinput` (autosuggest attribute), `cfinput` (sourcefortooltip attribute)

- These **will not work** out of the box with 2018 and above, but **you can re-enable them**
  - was also true as of CF2016 update 3 and per installer starting in Dec 2016

- To re-enable these, you just copy needed YUI library files back into CF
  - See bottom of deprecated features page
  - `helpx.adobe.com/coldfusion/deprecated-features.html`
OTHER COMPAT ISSUES IN CF2018

- Good news: semi-colons now optional in cfscript
- Update 1 added new getCanonicalPath(), used previously as a var in Fusebox
- Update 4 added new fileAppend()
- Bonus: CF2018 introduced PMT (Performance Monitoring Toolset)
COMPAT ISSUES INTRODUCED IN CF2016
- JSON datatype preservation: now inferred at compile time, rather than runtime
  - New jvm arg to disable that: `-Dcoldfusion.literal.preservetype="false/true"
    - Must also clear previously compiled CF files in cfclasses
  - [https://adobe.ly/3w7PjQl](https://adobe.ly/3w7PjQl)
- CFCollection ignores PATH attribute, uses path defined in CF Admin
- MS Access: workarounds from 10/11 to enable it no longer work
  - That said, some find Access working even in CF2021
- Update 3 added new `isValid()`, `queryGetResult()`, `structToSorted()` functions
- Update 7 added new `getCanonicalPath()`, used previously as a var in Fusebox
- Update 11 added new `fileAppend()`

- Bonus: CF2016 introduced option for storing sessions in Redis
COMPAT ISSUES INTRODUCED IN CF11
CF11 Update 3 re-enabled long-dormant CFProcparam DBVARNANE

- Had been ignored since CF6 (except briefly in 7.0.1)
- Problem: some DBs now require prefix to dbvarname value
  - (":" for Oracle and "@" for SQLServer)
- You may have old code that just says dbvarname = "param1" which would now fail
  - For SQL Server (as example), should be dbvarname = "@param1"
- While a jvm arg was added at the time to revert this, that was NOT carried into CF2016 and above
- coldfusion.adobe.com/2015/07/coldfusion-11-and-dbvarname-attribute/
Cfchart changed underlying engine (from greenpoint to zing)
- No longer uses XML for extensibility, but json instead
- Tool provided with CF to assist in that conversion
  - cfusion/bin/cfchart_xmltojson (bat/sh)
New strict number validation: if true (default), results of numeric validations (in tags, functions, return from methods) fail if value is not truly numeric
- Settable in cfapplication or application.cfc: strictNumberValidation
CF11 update 11 broke any use of an /api folder off the root of your site
- CF now uses that (like /rest folder) for REST feature
- CF2018 changed it, however, to /cfapiresources instead
With CF11, Adobe started obsoleting things that were just deprecated before

- Mostly old attributes of some tags
  - More at blog post from that time https://bit.ly/2ZU1yEE
- But also some tags: cfgraph, cf servlet, cftextinput
- And deprecated many more things (meaning could be removed in the future)
- Again, see deprecated features page for details
  - helpx.adobe.com/coldfusion/deprecated-features.html
Good news: Full support for cfscript (can optionally code entirely without tags)

Bonus: CF11 added new CF Admin page to view Secure Profile’s 20 settings

Also, some things formerly Enterprise-only ARE now available in Standard

- CF archive (car) mechanism (to move CF admin settings from server to server)
- Security Sandbox (to lockdown what code in different directories can do)
- HTML5 charts
- Web socket limit lifted

OTHER COMPAT ISSUES IN CF11
COMPAT ISSUES INTRODUCED IN CF10
COMPAT ISSUES IN CF10

- CF 10 removed Verity search engine, replaced with Solr
- Web services changed to support version 2, can change back to v1
  - See Admin-, application-, and code-level options to change wsversion, etc
- Bonus: CF10 added "secure profile", which can enable 20 admin settings
  - These might impact how your app runs, if someone implements it during CF install
MIGRATION TOOLS AND RESOURCES
CF Admin: Code Compatibility Analyzer
  - In Debugging & Logging section
Adobe created CFFiddle, at cffiddle.org
  - Can test code against CF2021, 2018, and 2016 (each at their latest update)
  - See also trycf.com, supporting all 3 CF versions and Lucee 4/5 as well
Docker images available for CF2021, CF2018 can aid in testing
  - Consider also Ortus Commandbox (and images) as another alternative to installing CF
  - ortussolutions.com/products/commandbox
Again, the deprecated features page in docs
- helpx.adobe.com/coldfusion/deprecated-features.html

Note that Adobe also offers a “Migration Guide” PDF…
- adobe.ly/3bDlVbI
  - But most of the doc is about installation, configuration, administration settings

See CF docs for more on new and changed language features
- Both the “CFML Reference” and “Developing ColdFusion Applications”
- Also blog posts from CF Team, at time of each release, at CF Portal: coldfusion.adobe.com

Also, I can help: something of CF “reference librarian”
  - I also help folks understand and resolve problems, whether migration or other

...and one last reason to motivate migration to CF2021...
When does/did public support (providing updates) end, for each recent version?

- For CF2021, in 2025
- For CF2018, in 2023
- For CF2016, ended in Mar 2021
- For CF11, ended in 2019

helpx.adobe.com/support/programs/eol-matrix.html
Beware that sometimes people assert a new CF version “broke something”...

- But perhaps it is also a move to a new OS, or a new box
  - Maybe the ip address changed and some server you call is not allowing access
  - You may even find that running your CURRENT CF version on new box would show same error
- Maybe some configuration of the CF Admin or web server was not “brought over”
  - Key items include max heap size, template cache size, and more
  - See my talk on migrating/comparing admin settings: carehart.org/presentations/#migrating_admin
- Maybe a CF update was applied but an error occurred. See update install log
  - carehart.org/blog/client/index.cfm/2016/9/6/solve_common_problems_with_CF_updates_in_10_and_above
- Maybe the new CF runs on a more recent Java version than you used before
  - Java 11.0.11 (and 1.8.0_291) both block calls via tls to servers not supporting 1.2 or above
  - carehart.org/blog/client/index.cfm/2021/4/26/new_java_updates_for_java_8_and_11_as_of_Apr_2021
- And so on
We've seen how sometimes one update will fix issues in an earlier update
  - On moving to CF2021, generally best to apply latest update
  - Update 2 (Sep 2021), as of this presentation in early Nov 2021
Note that a new CF2021 installer was also offered at the same time
  - It comes with update 2 and Java 11.0.11 pre-configured
Finally, admin changes (in each release) can affect how your CFML app runs
  - Similarly, new features can enable doing more/better things, so consider them also
  - Again, see my “hidden gems” talks for each release to see all that was new in each
Lots of migration issues to beware in any move, especially from older versions
  • A problem you hit may not be “new” to CF2021 but some version you skipped
• Again, for some migration is trivial, for others difficult, most in the middle
  • I hope this talk has opened your eyes, to be better prepared, whatever version
  • I can help in understanding, resolving issues, and more

My contact info for follow-up:
  • Charlie Arehart
  • charlie@carehart.org
  • @carehart (Tw, Fb, Li, Slack, Skype, GitHub)
  • Slides: carehart.org/presentations