COLDFUSION: MORE MODERN THAN MOST REALIZE

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

Updated Jul 19, 2022
- Briefly: the most recent evolution, CF2021 (its modernizing features)
- Modernizing of CF and CFML over time (even many versions ago)
  - Developer features (updates to language, including cfscript and more)
  - Other CF aspects/features (deployment, security, monitoring, IDE support, more)
- Evolution in the CF community (tool-makers, learning and help resources, etc.)
- Slides available online for you now or later:
  - carehart.org/presentations
- Videos of these Dev Week sessions should be online soon
  - coldfusion.adobe.com/videos/
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! 😊

And in fact, even this talk isn’t “all about me” 😊 ....
“Modern Functional & Fluent CFML REST APIs”
“Building apps with VueJS x ColdFusion APIs”
“Learn DevOps: CI/CD and Pipelines with ColdFusion 2021”
“A Sneak Peek into ColdFusion Builder on VSCode”
“Enhance Your Web Pages with Alpine.js”
“Improving Legacy Code while Retaining Your Sanity”
“You handle the CRUD, let [cloud db’s] handle the rest”
“I Didn’t Know S3 Could Do That!”
And others

…and yet still…
"I haven't heard of CF since the millennium. Didn't know it's still used."

- Just so sad!

"I had to google what is ColdFusion to understand why someone would try to bring attention to it. So what I found is a proprietary system with an XML language."

- XML? What?!

"The problem with CF is not the language..., the problem is with the server. It just one gigantic Java monolith where, when one wants to use a couple of if statements, one gets 1GB of legacy Java dependencies loaded."

- Not with CF2021!
"There's a huge "fake news" style of negative press surrounding CF that needs to be handled before this will turn around. Remember the adage - if you tell a lie enough times, it becomes the truth."

"Over the last five years it's matured a lot - if it wasn't for the Coldbox framework, CommandBox package manager, and Lucee on Docker I wouldn't be using it. But modern CF is pretty awesome, I think."

...and one more...
“When most folks think of ColdFusion, they think of the version they used in '05 with `<cfdiv>` and `<cftable>` and all that good stuff. Be serious. That legacy junk has been frowned upon for a long time. The real ColdFusion is a **JS-like syntax with full object-oriented functionality** coming straight from Java - **final and static functions, accessors, etc** - and **functional programming** features such as **threading, async and locking**. To me, that's what ColdFusion really is.”

These quotes are from blog posts by Lucee fan Michael Born, and comments:

- dev.to/mikeborn/why-cf-3jal
- dev.to/mikeborn/yes-coldfusion-is-unpopular-no-i-don-t-care-5f8c
KEY CF2021 MODERNIZATION FEATURES

No time to demo all these, and really they’re not the main point of this talk
- Cloud message queuing (AWS SQS, Azure Service Bus)
- AWS SNS (notifications) and SES (email)
- Cloud noSQL support (Amazon DynamoDB, Azure Cosmos DB)
- Multi-cloud RDS support
- Cloud Storage Svcs (AWS S3, Azure Blob)
- And more...

- And yes, all new features in 2021 are supported in Standard as well as Enterprise
- Devops-oriented features
  - Optional smaller zip/Express installer (under 200 meg)
  - Available **package manager** (cfpm, letting you choose engine components)
  - Available **JSON-based configuration** (cfsetup)
  - CF2021 Docker containers (more in a bit) can leverage all 3 of the above

- Other features
  - **SAML** (single sign-on/ logout) support (configuring an IDP or SP)
  - MongoDB (noSQL) support
  - Addition of Azure Redis cache as CF session storage option
  - Addition of AWS Elasticache, Azure Redis, as CF distributed caching options
    - CF Enterprise only (only thing “new” in 2021 that is Ent-only)
Support for parallel programming in arrays, structs, and queries
Immediately invoked function expressions (IIFE)
Iterable objects, via Spread and Rest operators
Iterator support for CFCs
Support for static fields and methods for CFCs
Identity operator, "==="
Case-sensitive and ordered structs
"Destructuring assignment" feature for structs, arrays
Dynamic switch case
...and much more!
Each of these new features is well-documented!

Can google the topics, or see links offered in these posts at time of release:

- carehart.org/blog/client/index.cfm/2020/11/10/the_many_new_and_improved_features_in_CF2021
- community.adobe.com/t5/coldfusion/introducing-adobe-coldfusion-2021-release/m-p/11585468

See Adobe conference recordings of past year

- “Modern Semantics Of ColdFusion 2021” (video.tv.adobe.com/v/334883)

But maybe you’re not yet on CF2021, or don’t have plans to be

- Am I saying you must move to CF2021 to benefit from “modern CFML”?...
- Definitely not!
- CF and CFML have evolved with each release
  - Will chronicle some key aspects of modernizing CF/CFML in various releases
- Reminder of timeline of releases (the past 20 years)
  - CF2021, 2018, 2016
  - CF5 (2001) and before
DEVELOPER-ORIENTED EVOLUTION

Several categories by which CFML has evolved!
CF4: initial release of cfscript
CF8: increased support of cfscript
CF9: CFCs could be written entirely in cfscript (no need for cfscript tag)
CF11: complete support of cfscript, for all tags
CF2018: semi-colons no longer required (still supported)
Don’t miss cfscript.me (from Pete Frietag)
  ▶ Converts CF tags to cfscript, vice-versa, free and on a single page
- CF9: ternary operator (??)
- CF11: elvis operator (??:)
- CF2016: safe navigation operator (?.)
- CF2018: lambdas/fat arrow functions (and in tags), null support
- CF2021: items listed earlier
- CF10: closures; Array Each/Filter, Struct Each/Filter, List Filter and more
- CF11:
  - Map/Reduce for arrays, structs, lists
  - First-class functions
- CF2016: map/reduce/each/filter for queries
- CF2018:
  - closures in tags
  - array some/every/splice; struct some/every; query some/every/slice/append/prepend & more
- CF2021: items listed earlier
- CF7: `cfthread`
- CF2018: `runAsync`, `futures` (like `promises`)
- CF2021: `parallelism` option in various features

- Again, I realize demos would be awesome, but we’re not half-done
  - We’d be here all morning!
  - Again, there are docs (often with demo code) for all these
MEMBER FUNCTION EVOLUTION

- **CF11**: member functions introduced
- **CF2016**: many new member functions
- **CF2018**:  
  - More for arrays, queries, xml, numeric, date, async, more  
  - **Chaining** of member functions  
- **CF2021**: array push, pop, shift, unshift
- Forever we had structs (via `structnew`)
- CF8: *implicit notation* `{…}` syntax for creating structs (and arrays `[…]`)
- CF10: use of `colon` for key/value *separator*
- CF2016 declaring *ordered/sorted* structs
- CF2021: declaring *casesensitive* and *ordered-casesensitive* structs
- **CF8:**
  - Added `serializeJson`, `deserializeJson`, `isJson` functions
  - added support of `returnformat` of CF methods as `json` (or plain or `wddx`)
  - As well as support of `returning json` from various tags, functions
- **CF11:** added support to serialize `queries` to JSON; and optional `useCustomSerializer`
- **CF2016:**
  - `serialization.structmetadata` property in application.cfc to declare `datatypes` for serialization
  - update 2 added `structSetMetadata()`
- **CF2018:** preservation of quoted `Booleans` and numbers as strings
- **CF2021**
  - Returntype of `json` for `cfquery` and `queryExecute` (`,json/array`), as well as `array`, `query`
  - And also `json/struct` and `struct` (not documented originally)
CF6: CFCs introduced

CF8:
- support for interfaces
- added onmissingmethod (option to handle calls to a CFC for an undefined method)

CF9:
- Again, CFCs could be coded entirely in script
- implicit getters and setters, via accessors="true"
- support for new keyword to implicitly invoke CFCs
- New oncfcrequest method in application.cfc (like onrequest, for CFC calls via http)

CF10: Implicit constructors for CFCs, method chaining

CF11: invokeImplicitAccessor property, for CFCs, including application.cfc
CF2018

- **Abstract** CFCs & Methods
- **Subtyping/covariant** method return types
- **Final** variables, methods, components
- **Default** functions in interface

CF2021

- Support for **static fields and methods** for CFCs
- **Iterator** support for CFCs
- Property **shorthand** (creates keys with same name as those passed in)
- Forever we had application.cfm
- CF6: introduced application.cfc
- CF8:
  - application-level mappings
  - new onmissingtemplate handler
- CF10:
  - this.sessioncookie; this.authcookie
  - application-level java class loading
  - onabort method
  - applicationGetMetadata function
- CF11: app-level datasource definition
- ...
CF2016: this.searchImplicitScopes, this.passArrayByReference options added
CF2018: this.enableNullSupport option added
CF2021: this.timeZone option added
  along with many things related to new features discussed initially today
To name just a few!
OTHER FEATURE EVOLUTION

A few categories beyond CFML the language
Forever we’ve had installer
- CF6: silent install; optional WAR file deployment
- CF8: CAR file admin config export/import
  - (2014: Ortus CommandBox released)
- Since CF2016: Docker images (more in a moment); Amazon AMI
  - (2018: CommandBox Cfconfig module released)
- CF2021: zip installer and cfinstall; modular deployment; json-based config (cfsetup)
In 2018, **CF2016** and **CF2018** first offered as images, then **CF2021** in 2020
- (Ortus also offers Commandbox Docker images for CF and Lucee)
- Adobe images originally offered via bintray.com
  - Now offered via [dockerhub.com](http://dockerhub.com) and [Amazon ECR](http://aws.amazon.com/ecr/)
- Adobe images offered admin config via CAR file, CFML script/Admin API

2018: InLeague published “CF Swarm” guide to CFML docker deployment

2021
- Adobe images offer admin config via json/cfsetup; support modules/cfpm
- BTW: CF images can of course be deployed to Kubernetes, Cloud services, etc
  - Adobe images for each new update to supported CF versions

**DOCKER SUPPORT EVOLUTION**
• CF8:
  • **Developer Security Guidelines** (last updated in **CF11**)  
  • initial **ColdFusion Lockdown Guide** (still updated in 2021)
• 2009: **Fuseguard, Hackmycf** released (from Foundeo)
• CF10
  • Many security features introduced into CFML and CF, for **XSS, CSRF, CRLF**, more
  • **Secure Profile**:
    • Option to make 20 admin settings most secure
• CF11: Secure profile causes cfhttp addtoken to default to “no”
  
  ...
• CF11
  • Anti-samy support bundled, isSafeHTML/getSafeHTML

• CF2016
  • Security Code Analyzer
    • Introduced in 2016 originally for CF Enterprise only; now also for Standard/Dev in CF2021
  • Samesite cookie support (per update 15; CF2018 update 9)
  • CF Admin no longer accessible via external web server, by default

• CF2018
  ▶ Auto Lockdown tool
  ▶ 2019: Fixinator released (from Foundeo, at fixinator.app)
- 2006: **FusionReactor, SeeFusion** released
- CF8: **ColdFusion Server Monitor** (for CF Enterprise-only)
- CF2018: ColdFusion **Performance Monitor Toolkit** (PMT, for CF Ent, Std, Dev)
  - Replaced CF Server Monitor
- CF5 had rudimentary command-line support
- CF2016: re-introduced more capable CLI, naming file to execute
  - CF Docker images also support CLI for execution of a single named file
- CF2018: introduced a true REPL
- (Commandbox has long offered a still more capable CLI and REPL)
  - for CF and Lucee

CLI EVOLUTION
OTHER FEATURE EVOLUTION

- Deprecation/obsoletion/retirement of features:
  - helpx.adobe.com/coldfusion/deprecated-features.html
- Underlying libraries updated each release
- Supported OS’s, db servers, web servers updated each release, or in CF updates
- CF11 opened Enterprise features to Standard
  - Security Sandbox, CAR export/import, Multiple Admin users, HTML5 charts
- CF2016 added powerful API Manager (for CF Enterprise-only)
2010: **CFBuilder** released, replacing Dreamweaver, CF Studio, HomeSite
   - Based on Eclipse

CFML **extensions** existed also for Sublime, NotePad++, IntelliJ, Atom, others

2015: VisualStudio Code (**VSCode**) released, gained traction quickly
   - Multiple CFML extensions created in years since

2021
   - Adobe announced plans to change CFBuilder from Eclipse to VSCode
   - Finally coming out this summer
- Enterprise integrations (databases, exchange)
- PDF generation and processing
- Websockets
- ORM/hibernate
- Caching (including support for Redis, elasticcache, memacached, ehcache, JCS)
- And much, much more

ALL THIS ON TOP OTHER CF FEATURES
EVOLUTION IN THE CF COMMUNITY
• Many have come and gone, **these still stand** and are ever-evolving:
  • Ortus: **CommandBox, ForgeBox, TestBox, WireBox, CacheBox**, and more
  • Foundeo (**Fixinator, Fuseguard, HackmyCF**, etc.)
  • Intergral (**FusionReactor, NerdVision**, etc.)
  • **Lucee** (open source CFML engine)
  • And more

• Also, a search for “coldfusion” at github returns over 2200 projects
  • With more than 300 showing update date of 2021 or greater
Many have come and gone, but these still stand and are evolving:
- ColdBox
- CFWheels
- Fw/1
- And others

And still other “kinds” of CFML frameworks
- Taffy (REST), DI/1 (injection), etc.

Several CMSs (CommonSpot, CONTENS, Contentbox, FarCry, Preside, and more)
CF docs (more than just the CFML Reference)
Cfdocs.org
Learn Modern CFML in 100 Minutes (modern-cfml.ortusbooks.com)
Learn cf in a week (learn cf in a week.com)
Modernize or Die podcast (cfmlnews.modernizeordie.io)
CF Alive podcast (teratech.com/podcast/)
cfcasts.com
CF Conferences: DevWeek, CFSummit, IntoTheBox, etc.
...

LEARNING RESOURCES
See also cffiddle.org, trycf.com (online code testing tools for different CFML engines/versions)

akbarsait.com, who lists the blog posts/resources about new features of each CF release

Online ColdFusion Meetup (coldfusionmeetup.com)

My “Hidden Gems” presentations on each release, at carehart.org/presentations

And still others, indeed much more, at my site: cf411.com
GETTING HELP

- Cf forums (community.adobe.com)
- CF portal (coldfusion.adobe.com)
- Cf slack (ctml.slack.com)
- Facebook CF Programmers group

- Finally, what about Adobe and the future of CF?
ADOBE’S COMMITMENT TO CF

- 5 releases in past 10 years
  - About 15-20 updates per release
  - Hundreds of bugs fixed over each release
  - Support cycle of 5 years per version
  - Next release already in planning state

- Resources they support:
  - CF Devweek, CF Summit conferences
  - Ongoing webinars (from Brian Sappey, Mark Takata, and others)
  - CF Portal, community forums
  - Certification program
  - Free install support, free trial, free developer edition

- Phew...just about done
As for **finding developers**, help them see these options for using modern CFML

- They should find modern CFML more familiar than they’d likely expect!

As for **management buy-in**, help them see that CF need not be a dead-end

- Refactoring does not have to presume “move off of CF”
- Nor does cloud deployment/integration, as we have seen amply!

One last reminder/plea/quiz…
ASK A CF SKEPTIC:
DOES CF SUPPORT...?

- Closures
- Promises/futures
- Lambdas/fat arrow functions
- Functional programming
- Higher order functions
- Destructuring assignment
- Docker/container deployment

- Cloud service integration
- noSQL support
- SAML/Single-signon
- Json-based configuration
- Package management
- 140mb core engine
- CLI/REPL
- And more
I welcome feedback, corrections, questions, challenges

Again, my contact info for follow-up:
- Charlie Arehart, charlie@carehart.org
- @carehart (Tw, Fb, Li, Slack, Skype, GitHub)
- Slides: carehart.org/presentations
- Adobe’s DevWeek Gift card giveaway

THAT’S ALL, FOLKS!