

COLDFUSION AT 25: MORE MODERN THAN MOST REALIZE



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

Updated Apr 25, 2022

- ▶ Briefly: the most recent evolution, CF2021 (its modernizing features)
- ▶ Evolution over time in CF and CFML (across versions as far back as CF8!)
 - ▶ Developer features (cfscript, language, json support, etc.)
 - ▶ Other features (deployment, security, monitoring, IDEs, etc.)
- ▶ Evolution in the CF community (tool-makers, learning and help resources, etc.)

TOPICS

2

- ▶ Slides available online for you now or later:
 - ▶ carehart.org/presentations
 - ▶ Original subtitle was “Not the kid that most remember”
- ▶ Other talks this week also discuss aspects of CF’s evolution, including:
 - ▶ Modern Semantics Of ColdFusion 2021
 - ▶ Now in Track 1 (each group can watch the other’s recording)
 - ▶ ColdFusion Modernization Challenges
 - ▶ Thurs at 2p ET in Track 2
 - ▶ And other sessions discuss recent, modern concepts of course
- ▶ 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! 😊

ABOUT ME

4

- ▶ *"I haven't heard of CF since the millennium. Didn't know it's still used."*
- ▶ *"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."*
- ▶ *"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."*

SOME TYPICAL PERSPECTIVES ON CF...

5

- ▶ *“There is 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

AND SOME RESPONSES...

- ▶ “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:
 - ▶ dev.to/mikeborn/why-cf-3jal
 - ▶ dev.to/mikeborn/yes-coldfusion-is-unpopular-no-i-don-t-care-5f8c

SOME RESPONSES (CONT)

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

RICH CLOUD SUPPORT

- ▶ Devops-oriented features
 - ▶ Optional smaller zip/Express installer
 - ▶ Available **package manager** (cfpm)
 - ▶ Available **JSON-based configuration** (cfsetup)
 - ▶ CF2021 Docker containers 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” that is Ent-only)

OTHER FEATURES

- ▶ 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
- ▶ Create and execute Java within CFML
- ▶ Support for implementing/extending Java Interfaces
- ▶ ...and many more!

LANGUAGE MODERNIZATION IN 2021

11

- ▶ See links for each of these offered here:
 - ▶ 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
- ▶ Again, see the recording of the related Devweek talk
 - ▶ “Modern Semantics Of ColdFusion 2021”
- ▶ But maybe you’re not on CF2021, or don’t have plans for it
 - ▶ Am I saying you need to move to CF2021 to benefit from “modern CFML”?...

FINDING OUT MORE ABOUT CF2021

12

- ▶ Definitely not! CF and CFML have evolved since their beginning
 - ▶ Will chronicle key aspects of evolution in various respects
- ▶ Reminder of timeline of releases (the past 20 years)
 - ▶ CF2021, 2018, 2016
 - ▶ CF11 (2014), CF10 (2012), CF9 (2009), CF8 (2007), CF7 (2005), CF6 (2002), CF5 (2001)

CF EVOLUTION OVER THE RELEASES

DEVELOPER-ORIENTED EVOLUTION

14

Charlie Arehart
CArehart.org
@carehart

- ▶ CF4: initial release of cfscript
- ▶ CF8: increased support of cfscript
- ▶ CF9: CFCs can be written entirely in cfscript
- ▶ CF11: complete support of cfscript
- ▶ CF2018: semi-colons no longer required (still supported)
- ▶ **Don't miss cfscript.me** (from Pete Freitag)

CFSCRIPT EVOLUTION

15

- ▶ CF9: **ternary** operator (?)
- ▶ CF11: **elvis** operator (?:)
- ▶ CF2016: **safe navigation** operator (?.)
- ▶ CF2018: **lambdas/fat arrow** functions (and in tags), **null** support
- ▶ CF2021: items listed earlier

LANGUAGE EVOLUTION (JUST SOME)

16

- ▶ CF10: **closures**; ArrayEach/Filter, StructEach/Filter, ListFilter and more
- ▶ CF11:
 - ▶ **Map/Reduce** for arrays, structs, lists
 - ▶ **First-class functions**
- ▶ CF2016: map/reduce/each/filter for queries
- ▶ CF2018:
 - ▶ closures in tags
 - ▶ arraysome/every/splice; structsome/every; queriesome/every/slice/append/prepend&more

CLOSURES, HIGHER-ORDER FUNCTIONS, FUNCTIONAL PROGRAMMING

- ▶ CF7: cfthread
- ▶ CF2018: runAsync, **futures** (like **promises**)
- ▶ CF2021: **parallelism** option in various features

THREADED/PARALLEL PROCESSING

18

- ▶ CF11: **member functions** introduced
- ▶ CF2016: many new member functions
- ▶ CF2018:
 - ▶ for arrays, queries, xml, numeric, date, async, more
 - ▶ Chaining of member functions
- ▶ CF2021: array push, pop, shift, unshift

MEMBER FUNCTION EVOLUTION

- ▶ CF8: implicit notation/struct-literal syntax for creating structs (and arrays)
- ▶ CF10: use of colon for separator
- ▶ CF2016 ordered/sorted structs
- ▶ CF2021: casesensitive and ordered-casesensitive

STRUCT EVOLUTION

20

- ▶ CF8:
 - ▶ added support of returnformat of CF methods as json (or plain or wddx)
 - ▶ As well as support of returning json from various tags, functions
- ▶ CF2016: update 2 added setmetadata, to declare datatypes for serialization
- ▶ CF2018: preservation of quoted Booleans as strings
- ▶ CF2021
 - ▶ Return type of json for cfquery and queryExecute (json/array, as well as array, query)
 - ▶ And also (not documented originally): json/struct and struct

- ▶ CF6: Introduced
- ▶ CF8: support for interfaces; *onmissingmethod* added
- ▶ CF9:
 - ▶ CFCs could be coded entirely in script
 - ▶ implicit getters and setters, via `accessors="true"`
 - ▶ support for *new* keyword to invoke CFCs
 - ▶ New *oncfcrequest* method
- ▶ CF10: Implicit constructors for CFCs, implicit getters/setters; method chaining

CFC EVOLUTION

- ▶ 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

CFC EVOLUTION (CONT.)

- ▶ 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
- ▶ To name just a few
- ▶ More: helpx.adobe.com/coldfusion/cfml-reference/application-cfc-reference/application-variables.html

APPLICATION.CFC/CFM EVOLUTION

OTHER FEATURE EVOLUTION

25

Charlie Arehart
CArehart.org
@carehart

- ▶ Since beginning: installer
- ▶ Since CF6: silent install; WAR file deployment
- ▶ Since CF8: CAR file admin config
- ▶ 2014: Ortus CommandBox released
- ▶ Since CF2016: Docker images (more in a moment); Amazon AMI
- ▶ 2018: CommandBox Cfconfig module released
- ▶ Since CF2021: zip installer/cfinstall; modular deployment; json-based config/cfsetup

DEPLOYMENT OPTIONS EVOLUTION

26

- ▶ CF2016 and CF2018 first offered as images in 2018, then 2021 in 2020
 - ▶ Ortus also offers Commandbox Docker images for CF (and Lucee)
 - ▶ Images were originally offered via bintray.com, with image for each new CF update
 - ▶ Now offered via dockerhub.com and AWS ECR
 - ▶ Images offered admin config via CAR file, CFML script/Admin API
- ▶ 2018: InLeague published “CF Swarm” guide to docker deployment
- ▶ 2021
 - ▶ Images offer admin config via json/cfsetup; support modules/cfpm
- ▶ BTW: CF images can of course be deployed to Kubernetes, Cloud services, etc

DOCKER SUPPORT EVOLUTION

27

- 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
 - Causes cfhttp addtoken to default to no (as of CF11)
- ...

SECURITY EVOLUTION (CONT.)

- CF11
 - Anti-samy support bundled/isSafeHTML/getSafeHTML
- CF2016
 - Security Code Analyzer
 - Introduced in 2016 originally for CF Enterprise only, now also 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)

SECURITY EVOLUTION

- ▶ 2006: FusionReactor, SeeFusion released
- ▶ CF8: ColdFusion Server Monitor (Enterprise-only)
- ▶ CF2018: ColdFusion Performance Monitor Toolkit (PMT, for Std and Ent)
 - ▶ Replacing CF Server Monitor

MONITORING EVOLUTION

30

- ▶ 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 offered a still more capable CLI and REPL
 - ▶ for CF and Lucee since 2018

CLI EVOLUTION

- ▶ Deprecation/obsolescence/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
- ▶ Adobe announced this month plans to support OpenJVM Azul Zulu in July
 - ▶ With their licensing agreement with Oracle for Oracle JVM ending in Dec 2022

OTHER FEATURE EVOLUTION

- ▶ 2010: CFBuilder released, replacing Dreamweaver, CF Studio, HomeSite
 - ▶ Based on Eclipse
- ▶ CFML extensions created for Sublime, NotePad++, IntelliJ, Atom, others
- ▶ 2015: VisualStudio Code (VSCode) released, gained traction quickly
 - ▶ Multiple CFML extensions created in years since
- ▶ 2021
 - ▶ Adobe plans change of CFBuilder from Eclipse to VSCode

IDE/EDITOR EVOLUTION

- ▶ Enterprise integrations (databases, exchange)
- ▶ PDF generation and processing
- ▶ Websockets
- ▶ ORM/hibernate
- ▶ Caching (including support for Redis, elasticcache, memcached, ehcache, JCS)
- ▶ And much, much more

ALL THIS ON TOP OTHER CF FEATURES

34

EVOLUTION IN THE CF COMMUNITY

35

Charlie Arehart
CArehart.org
@carehart

- 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.)
 - Webapper (SeeFusion)
 - Lucee (open source CFML engine)
 - And more in a moment
- Also, a search for “coldfusion” at github returns over 2400 projects
 - With more than 270 showing update date of 2020 or greater

TOOL MAKERS

36

- ▶ 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, CONTENTS, Contentbox, FarCry, Preside, and more)

FRAMEWORKS

- ▶ CF docs (more than just the CFML Reference)
 - ▶ coldfusion.adobe.com/2017/11/did-you-know-theres-far-more-to-the-cf-docs-than-just-the-cfml-reference
- ▶ Learn Modern CFML in 100 Minutes (modern-cfml.ortusbooks.com)
- ▶ Learn cf in a week (learncfinaweek.com)
- ▶ Modernize or Die podcast (cfmlnews.modernizeordie.io)
- ▶ CF Alive podcast (teratech.com/podcast/)
- ▶ cfcasts.com
- ▶ Online ColdFusion Meetup (coldfusionmeetup.com)
- ▶ cf411.com
- ▶ akbarsait.com, who lists the blog posts/resources about new features of **each CF release**
- ▶ My "Hidden Gems" presentations on each release, at carehart.org/presentations
- ▶ CF Conferences: DevWeek, CFSummit, IntoTheBox, CFCamp
- ▶ See also cffiddle.org, trycf.com (online code testing tools for different CFML engines/versions)
- ▶ And others

LEARNING RESOURCES

- ▶ Cf forums (community.adobe.com)
- ▶ CF portal (coldfusion.adobe.com)
- ▶ Cf slack (cfml.slack.com)
- ▶ Facebook CF Programmers group
- ▶ Cfdocs.org

GETTING HELP

39

- ▶ 5 releases in 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
 - ▶ Devweek, CF Summit conferences
 - ▶ Ongoing webinars (recent multi-part series by Brian Sappey on API Manager)
 - ▶ CF Portal, community forums
 - ▶ Certification program
 - ▶ Free install support, free trial, free developer edition
- ▶ Phew...let's wrap up...

ADOBE'S COMMITMENT TO CF

40

- ▶ 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

ASK A SKEPTIC: DOES CF SUPPORT...?

41

- ▶ As for finding developers, help them see these options for using modern CFML
 - ▶ They should find modern CFML more familiar than they'd expect!
- ▶ As for management, help them see that CF need not be a dead-end
 - ▶ Refactoring does not have to presume “move off of CF”
 - ▶ Cloud deployment/integration does NOT have to presume “move off of CF”
- ▶ Whether you treat your CF instances as “pet” or “cattle”, CF serves you well
 - ▶ Not everyone moves at same pace, is at same level
 - ▶ Let's all be patient with one another, working together as a community

WRAP-UP: MODERNIZING FOR YOUR ORG

42

- ▶ 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

THAT'S ALL, FOLKS!