What is Adobe ColdFusion? Why it Still remains the best programming language!

Your life as a CIO is filled with make-or-break decisions. The size, trajectory, and effectiveness of your employees rest on a handful of choices often made early in or tenure. And maybe a few bigger choices further down the line.

This is normal. A lot of weight rests on an IT department, regardless of what industry your company is a part of. From retail, media, financial services to even engineering, solid IT operations are the key to any modern company's success.

The employees you hire matter. Their training and growth matters. The ideas they generate, their investment in the company's success, your ability to manage them… These all matter.

Also, the tools you let them work with. The hardware, computers, technology. These all matter.

The programming language you choose matters as much as all of these tools. Sometimes, even more so.

A programming language is the lifeblood of nearly all the work you do as a CIO, affecting all aspects of the job. You may not recognize it. From internal matters like employee morale and training to the virtual face that greets your costumers — it's your programming language that's at the base of all these interactions.

It's also one of the more-permanent choices you'll make. Code has a funny knack for becoming tangled with countless small changes and new features being added all the time. Your programming language stays put through hardware changes, server switches, and numerous employees. You don't want your programming language to follow you around like a bad stench.

Which is why you need something stable. Something modern. Something secure. And something adaptable.

That "something" is ColdFusion.

Don't you love seeing happy developers?

ColdFusion Basics

ColdFusion's foundations are built on bedrock IT values security, efficiency, ease of use and evolution. First created in 1995, CF has enjoyed status as one of the IT world's ageless wonders. It's seen the rise and fall of many consumer electronics, tech fads, and rival programming languages. You'll learn what makes ColdFusion so hardy later.

Many CIOs may wonder what are the benefits of using ColdFusion over say, Java or PHP? Perhaps they want to start using CF, but they have some concerns.

Can something well into its third decade compete with newer languages in a fast-changing environment?

Of course, it can.

Most programming languages are like ripening fruit. Time is their enemy. ColdFusion, however, is like fine wine. It evolves and keeps getting better with age.

This article will answer your programming language-related questions. There are plenty, and they merit some thought and exploration. It's one of the few decisions you can make that'll have a daily impact on your department's life. Those questions include:

- What is ColdFusion?
- Is ColdFusion still alive? Even in the new (2020) decade?
- Is ColdFusion secure?
- I'm thinking about switching to ColdFusion but…
- What makes ColdFusion different from other languages?
- What's in ColdFusion's Future?
- How it will make your life easier and more productive.

I'm thinking about switching to ColdFusion but...

Switching an entire programming language — the essential architecture — of your company's app is a big shift. Its demands a seachange in your team's common language, mindset, and workflow. Also, it's something of a pain if done wrong.

You have a demanding job. You need to make sure operations continue humming along without many headaches. That's fair, and Adobe's ColdFusion has taken all those expectations into consideration. We'll address those concerns soon.

Some common concerns

Switching over to a new programming language, or jumping back to ColdFusion, may seem daunting. A phobia of the unknown is common. There are plenty of problems which can appear out of nowhere. Days, whole careers even, can be ruined. Right?

With other, less-stable programming languages

But in this case, it's completely unfounded.

CIOs may have concerns about ColdFusion's limitations. Let's put those to rest as soon as possible.

Technical debt

No, you won't find this listed on your company's balance sheet. But it'll cost you anyway.

Technical debt is the accumulation, over time, of bad habits, code degradation and shoddy design. In short, it doesn't pay to be lazy! cutting corners for short term gain which leads to long-term problems.

This includes deliberate shortcuts made in the development phase in order to get the application out quicker.

Over time, technical debt can lead to lags in updates, performance tuning and projects running at a snail's pace. You don't want that. Your users and customers, especially, don't want that.

Technical debt can also mean not future-proofing your code as best you can. Granted, your company can't forsee all the innovations on the horizon.

Fortunately, ColdFusion has tools and elegant coding necessary to avoid that debt trap altogether. It's all built into the nature of CF itself.

If a programming language is efficient and easy-to-use, you'll be less tempted to find shortcuts and workarounds. It's efficient, so you don't have to be lazy. There are even some frameworks that can prevent technical debt from accumulating.

Price

Is ColdFusion's price an issue for your company? Budgets are tight, and the accounting department must be satisfied with the choices you make.

It could seem hard to justify a nearly $10,000 upfront cost to the higher-ups, which is how much you'll pay for ColdFusion's Enterprise edition, or the $2,500 you need to shell out for the standard edition.

Over time, the efficiencies and peace-of-mind which come with using ColdFusion offers a spectacular return on investment. Better-equipped developers and operations teams testing products in real time can turn projects around at a faster pace. Lower odds of mistakes mean less time troubleshooting. And security? It pays for itself.

But if $10,000 is still too much to pay up front, there's another option which could offer an introduction, as well as later justification for the full investment into Adobe ColdFusion.

Meet Lucee, Adobe ColdFusion's no-cost, open source sibling. About 40 percent of CFers use Lucee, making it one of the more vibrant venues for ColdFusion innovation. And it's a great place for you and your developers to get started.

Lucee includes everything an IT team needs to start using CFML, and does it almost as well as Adobe's proprietary software. It has the same awesome tag-based structure and a set of features designed to make programming easy. It'll give your team a great starting off point, or simply a chance to test drive ColdFusion without the monetary commitment. Obviously, after you've fallen madly in love with CF you'll want to buy the Adobe version.

Or you could stick with Lucee.

The community of developers tweaking and adding features to Lucee has helped keep alive and, by extension, invigorated ColdFusion as well.

Is ColdFusion still alive?

Some naysayers and snobbish developers suggest ColdFusion has run its course. They hear CF mentioned, act confused and say, "People still use that?"

Yes! Well into its 24th year and counting! ColdFusion is not only alive, but it's also thriving among the real movers and shakers of the industry. About 60 percent of companies on the Fortune 100 list use ColdFusion!

Don't mind the naysayers. They're hung up on the few misusers and abusers of ColdFusion, who blame the programming language for their own negligence. They either accumulate too much technical debt and decide it better to scrap everything. Or they do not take advantage of containers to test their code before letting it out into the wild. But the stragglers may be the worst.

A lot of developers and CIOs, for various reasons, do not upgrade to newer versions and miss out on improvements. They're letting zombified ColdFusion programming language still lurking the tech world, dragging down the perception of ColdFusion and leaving it open to vulnerabilities that were fixed in later updates.

Don't take their mistakes as a rule!

The nature and purpose of a programming language are changing rapidly. ColdFusion has a long track record to keeping pace and evolving as necessary. Even now, it's prepared to meet new demands.

Languages live much longer

The whole idea of a dying markup language seems a bit silly in this era. For a few reasons.

First, applications live longer than ever before. During tech's Wild West era in the mid-90s through early 2000s, technology, processor speeds, and overall network capacities grew at a lightning pace, with web applications and sites evolving at a brisk pace. Many were left in the dust.

This happened at the same time as the infamous dotcom bubble, which saw many sites blossom then quickly wither. (Remember back when search engines were not named Google?)

Those days seem to be over, with consumer electronics reaching a stable high point and many of the tech world's services consolidating.

Stability rules and applications live longer as a result. Their programming language does too.

Setting those applications in motion with the right language — one you know will stick around for the long haul — creates a solid foundation for long-term success.

Not all programming languages are hard to learn

Some CIO's fear change comes from employees who dread learning a new programming language.

Well, who's in charge? And why are they scared?

Don't share those fears. Not for CFML! Developers tend to rely on what they already know. That muscle memory makes them stick with familiar languages, rather than move on to other, often better, options.

A developer or member of the operations team fearing a switch to CFML is often fearing the unfamiliar. You'll find, over time, that many of CFML's naysayers are people who've never used the language. Or ditched it in a huff when something else was briefly en vogue. Most CFers have tried the language once and stuck with it ever since.

Besides, any worries about taking on ColdFusion are silly. It's one of the easier-to-learn programming languages around!

Related: Learn ColdFusion Fast

Is it secure?

Good practices and quality control will help your team make the most out of ColdFusion.

With all the revenue and intellectual property in question, picking a programming language can sometimes feel like choosing an armored truck.

Are the doors securely locked? Do the drivers get to choose who can get in and out? How hard does a thief have to work to break in? Is it bullet-proof?

It can all be daunting.

This is often a major concern for CIOs. A hack could compromise user. Company data is at risk. Larger firms face even bigger consequences. Some can even be held for ransom, or seen data sold off to other shady characters.

CIOs worried about security are trying to save themselves and their company some embarrassment. You can forgive them for wanting to sleep easy knowing the application won't be cracked open in the middle of the night.

And with ColdFusion, they will get a full night's rest!

According to CVE Details, Adobe ColdFusion has fewer security vulnerabilities than its competitors — and has been among the most secure since 2006. That's over a decade of fortress-like protection.

Why? Adobe regularly releases security updates and patches as vulnerabilities are discovered. It's a bit of preventative medicine included in every purchase of ColdFusion.

How's that for a "dead" language?

ColdFusion also comes with big stock of security-minded tools to ensure your app remains safe. These include:

- Fixinator

  This tool by Foundeo executive and ColdFusion security guru Pete Freitag is quite a gem.

  Fixinator scans your ColdFusion programming language code for security issues. It seeks out a range of problems, including SQL Injection vulnerabilities and remote code execution, among others. It then automatically fixes any problems, if it can.

  Fixinator also integrates into several appendages within your applications, working with tools outside the bounds of normal CF programming language.

  Freitag, a friend of TeraTech's, also released a CFML Continuous Security Bundle, which includes Fixinator along with his other ColdFusion security-minded applications, FuseGuard and HackMyCF. Used in tandem, they'll make any CF-based web application a fortress.

- Auto Lockdown
  The new release of ColdFusion 2018 included this nifty tool, aimed at IT departments with security on their mind. Previous "manual" lockdowns were a real chore, an annoying multi-step process meant to keep security breaches to a minimum.

  The new auto lockdown does the legwork for you, shortening this process to a single click.

- Security code analyzer
  This feature is similar to Fixinator, but is Adobe's proprietary tool. It automatically searches code, seeking any known vulnerabilities and

assesses its threat level. It even gives you a ready-made solution for any security gaps which might lead to a breach.

What makes ColdFusion better than other languages?

You're considering the switch to CF but are wondering, "What will I gain?" Your company may be using PHP or Java with just-fine results. Why bother changing?

Your IT department could also work better, and ColdFusion allows that in several ways.

Related: Comparison between ColdFusion and other programming languages

CFCs

ColdFusion also uses Components, or CFCs, to turnaround apps at lightning speed. These tools are a shortcut for objects or a group of functions you can deploy onto a single item or entity.

This compounding of tools means a set of problems can be handled all at once. These sorts of efficiencies are multiplicative — the more you use them, the more time they save.

These ColdFusion Components are:

- faster after their first use
- reusable in as simple as a copy-paste manner
- More secure than other objects, since some can have restricted access

There are more differences between ColdFusion and other programming languages, but these two will be the most apparent and most-often used by developers.

CF Tags

CFML isn't a markup language, at least not by the textbook definition of the term. It is, instead, a highly-refined toolbox designed to use Java in an elegant and efficient way.

It does this through a tag-based system. Whereas other languages require a jerry-rigged system to manage its various functions and components, before moving to CFScript, CFML used tags

as a shorthand for commonly-used tools. This means a fraction of the keystrokes and much less time creating something that already exists.

Developers can even make their own custom tags for specially-built tools and reuse them as needed.

What is a Modern ColdFusion Language

25 years of ColdFusion and the vision for the next decade

How long can a programming language stay relevant? For ColdFusion, it's been nearly a quarter of a century and counting! Other languages have come and gone in that time, and ColdFusion Markup Language (CFML) saw them off. In programming terms, Adobe ColdFusion is the ageless wonder.

Why?

It was designed to be timeless.

Read further: 25 Years of Adobe ColdFusion

Adobe ColdFusion is a development platform that uses CFML to quickly build modern web apps. In comparison to other languages, ColdFusion is easier to learn, use, deploy and adapt.

ColdFusion also has built-in functions for commonly-used tools, like database access and PDF creation, which help web apps flourish.

This all adds up to a programming language that's ready to deploy fresh-out-of-the-box. Developers have an easy time learning the language, as many of the tools, they'd have to build natively already come baked into the app.

An up-to-date programming language with a bunch of fancy tricks sounds enticing. But CIOs like yourself are looking for the long haul. Thankfully, Adobe has you covered.

The company behind ColdFusion has a team dedicated to making sure ColdFusion stays relevant. The people are constantly looking to the future and paying attention to what the community wants.

And they're pretty good at it! New technologies and upgrades on the horizon promise an easier app development and deployment process.

Thanks to the folks at Adobe, and the many developers dedicated to ColdFusion, CFML will be even more stable than it is today!

ColdFusion is already being rigged to keep up with the times. Rakshith Naresh, Adobe's ColdFusion Project Manager, gave us a list of different changes set for ColdFusion's future. And it's exciting!

Here's how ColdFusion will be not only future-proof but even ahead of its time:

Cloud computing

Adobe worked overtime to make sure ColdFusion's cloud computing ability is modern. It already plays well with existing cloud frameworks and promises to add more innovations in the future.

But wait…

Why would you want to shift your app and resources to The Cloud? After all, it just produces rain and thunder. Right?

Well, not quite. The evolution of apps and programming has been shifting away from heavy machinery — massive server rooms as cold as refrigerators, humming fans cooling server blades and hard drives, etc. We're now moving towards a world where the rigs hosting your app are far away in some massive server farm. Owned by a mega tech company you know well, like Amazon or Google.

The cloud's much more flexible and scalable, making your company's app and entire operation more efficient, agile and ready to grow.

ColdFusion's working on making sure your app can function in these "clouds", with some new capabilities slated for its 2020 iteration, according to Naresh.

Adobe plans to turn CF into a multi-cloud platform. Meaning it'll behave the same way no matter which major cloud service provider your company chooses.

Naresh even foresees a future where companies keep bits of their applications and IT operations scattered across several cloud providers. Maybe the ID check portion of your web application will be on a cloud provider who has a good reputation for security, while your database will be on a provider with lots of storage at a reasonable price.

Diversifying your cloud services is a surefire way to keep your applications running, should one of the major services crash.

ColdFusion's leap into this arena will make cloud-based service apps within reach for even novice developers by having a unified interface to work with all major cloud services.

Eventually, Naresh said, you'll be able to migrate your application from one cloud service to another without having to rewrite code.

By using special developer tools like Docker (more on that later) and ColdFusion in tandem, your apps can run on the cloud with minimal upfront costs for new servers. This means the initial outlay and overhead to get your project going falls well short of whatever you have budgeted.

Related: Keeping up to date with modernized ColdFusion

IoT or the "Internet of Things"

You know that unusual feeling you get when it's just you and Alexa in the house? How talking to Amazon's smart speaker makes it seem like a member of the family? Or when your Google Home starts learning your habits better than you do?

Many future applications and innovations will involve more seamless integration of tech into our lives. Alexa or Google Home are just the beginning. The start of what's known as "The Internet of Things."

Broadly speaking, this IoT will merge together the other electronic appliances in your life. Your fridge, your toaster, your coffee maker, washer, dryer, television. These are all going to be augmented by new IoT advances.

It'll blur the ways consumers interact with the wider tech world. We've already ditched keyboards and mice in new applications, and programmed our lives into our devices. It only makes sense tech will become our butler.

The folks at Adobe making ColdFusion know what's on the horizon. They are building capabilities into the language that make it play nicely with these "smart" devices. Light bulbs will become "smart" light bulbs. Coffee machines will also get a brain of their own, and prep your morning cup of Joe just like your local barista.

This Internet of Things phenomena will also branch out into other fields, such as medicine. Monitoring devices hooked up to a patient can be networked to communicate a clearer picture to doctors using a single interface.

The IoT will even move over into parenting, health (a scale that reminds your fitness tracker to get you moving) and even your morning commute.

Adobe will make sure ColdFusion is one of the main languages that'll make it all happen.

But groggily thanking your coffee machine in the morning has its downsides, namely security issues. There's a significant threat from hackers who can override IoT-linked devices. And it could lead to chaos.

Vulnerabilities have already been found in IoT devices such as webcams, cardiac devices, and Jeeps. Yes, as in the car.

Thankfully, CF has a solution: Blockchain.

You've probably heard this term used over and over at cocktail parties, board meetings, and business pages. Blockchain has become the shiny new toy the entire tech world wants to exploit.

What's blockchain? Imagine a 2,000-piece puzzle that's needed to log into a website. And 2,000 strangers who don't know each other each have one piece of the puzzle, that they themselves can't identify. That, in layman's terms, is as close one can get to explaining a blockchain.

Blockchain diversifies security by spreading the job of managing access through multiple, loosely-linked servers, each providing only one piece of a broader puzzle.

It is a new frontier in online security and is close to uncrackable.

Deploying it in the internet of things makes devices much less hackable. But it won't be accessible or as well integrated into any other language. Besides ColdFusion, of course.

ColdFusion for the Next Decade – All about the Buzzworthy ColdFusion 2020

Adobe ColdFusion 2020 release date is close. Rakshith Naresh from Adobe team gave us a sneak and peak into the next decade version at the webinar which we are happy to promote and support. Of course, thanks to the Adobe ColdFusion team for giving us permission to do that.

One of the reasons for ColdFusion's success right from its inception is that the platform has been able to pivot at regular intervals to remain relevant for the future. There are very few technologies that have managed to stay in the game for so long and that is something all of us

in the community are proud of. ColdFusion 2020, slated to be released next year, is going one such pivotal release in the history of ColdFusion.

The vision for ColdFusion 2020 goes like this: ColdFusion for the Next Decade – All about the Buzzworthy ColdFusion 2020 (Webinar by Adobe)

ColdFusion makes your life easier and more productive

Now, let's take a look at the ways ColdFusion has made every CIO's life easier. Here are just some of the advantages of using CF:

RAD

The time it takes to transform an idea into a working, testable version is key to any IT operation's success. The difference between six weeks and six months for a beta to emerge could mean the difference between your service coming to market, or a competitor jumping ahead of you in line.

ColdFusion's got your back.

Rapid Application Development, or RAD, is one of ColdFusion's key strengths.

CF uses elegant, easy-to-learn code to help a developer turn an idea into reality faster than he or she could with other languages. This essentially means fewer keystrokes wasted on the commonly-used code. Readymade lines of code mean your developers won't waste precious time creating something already exists

Other languages require multiple lines of coding to be made from scratch. It's like a plumber having to forge his own wrench.

CFML comes ready to use like a well-supplied toolbox. It uses simple tags to trigger existing lines of code for common commands. Programming in CFML means you write less code yet achieve the same result.

This confuse this as a cookie-cutter approach though! ColdFusion still has all the room and freedom for customization you have with any other code. Think of it more like an autocomplete tool built into a programming language. You don't necessarily have to use the suggestions provided, but they can make life easier.

This RAD approach makes developing your app quicker and easier. From inception to development to testing. With ColdFusion, you'll be releasing your app out into the wild much sooner. No other programming language — not Java or PHP — can beat it.

Reliability

It's every CIOs worst nightmare: the unexpected. When we talk about reliability, we're really saying, "No surprises, please!"

This goes beyond the usual daily catastrophes. The key employee who gets poached by a competitor. Or the new orders from on high. Cutbacks, expansions, reorganizations. With all the headaches you face, at least a line of code should work as expected. Right?

Reliability is a cornerstone for all of ColdFusion's features. One of its biggest draws is ensuring it can handle much of what you'll throw at it.

CF includes many tools, with error handling stops, clustering, and regular updates, which prevent your code from toppling onto itself. These stopgaps ensure your company's ambitions won't get the best of you. ColdFusion will let you push it to the max, but also warn you when you're overdoing it.

These tools, combined with basic IT common sense such as defensive coding, load testing and beefed up server specs will make sure your ColdFusion server withstands long, tough stretches of high demand.

So you can worry about more important things. Like what to do when your app is more popular than expected…

Scalability

Leading an IT operation is uniquely problematic: too often, you can fall victim to your own success. And your popularity could lead to your downfall.

It seems ideal: being so popular people are breaking down doors to get to your service or app. Yet high demand for your outrageously awesome product or service causes a spike in traffic and can crash the whole application.

ColdFusion can deal with your popularity. Don't worry, it won't get jealous.

Scaling your app and server infrastructure to handle your popularity is far easier with ColdFusion. The CF programming language's ease and ability to evolve means your IT department can scale up an app or database faster, and with fewer errors.

By developing good habits within your IT department, similar to the reliability tips listed above, you can make sure your company's growth and popularity won't cost it a pretty penny.

ColdFusion makes sure you don't want to fall victim to your own success.

DevOps

Let's say you've got a vision for an app, and two disparate teams — development and operations — set on making it a reality. The old fashioned method would be to send each team to work in a vacuum, then they'd come together and turn the two resulting halves of the project into a Frankenstein's monster of an app. It'd work, sure. But boy was it ugly.

Wait — there's a better way. At least in ColdFusion

The DevOps model will become a standard within the IT industry, and Adobe ColdFusion fits it like a well-tailored suit.

This model allows development and operations teams to work in tandem when creating a new app. ColdFusion essentially has labs where lines of code can be tested under real-world condition. Sounds pretty neat, right? It's called "containerization."

Containerization helps speed along the development process by running your application in virtual "containers", where you can watch bits and pieces behave on their own and interact with others. Tweaks can be made quickly, and bugs show their ugly faces in an environment where they can be squashed and discarded well away from your customers.

It's a perfect way to get all the tougher, rougher parts of app development done in a safe environment. Besides, you want to find the weaknesses in your code before your users do, right?

It's a modern take on development and deployment, with the added benefit of letting collaboration flourish.

Tools like Docker add multiple new facets to your CF-based operation by speeding up the app-building process in a hybrid cloud environment. This is the virtual sandbox which makes ColdFusion ultra-fast development process also much more effective

CF (still) rocks, and YES, you should switch

Changes are tough, especially making a switch in programming languages. Not just for companies but for developers and even CIOs trying to manage the change. Picking the right language is key.

ColdFusion, with its ease of use and elegant solutions to complicated problems, would be a logical and natural fit.

And to continue learning how to make **your ColdFusion apps more modern and alive**, I encourage you to download our free ColdFusion Alive Best Practices Checklist.

Because… perhaps you are responsible for a mission-critical or revenue-generating CF application that you don't trust 100%, where implementing new features is a painful ad-hoc process with slow turnaround even for simple requests.

What if you have no contingency plan for a sudden developer departure or a server outage? Perhaps every time a new freelancer works on your site, something breaks. Or your application availability, security, and reliability are poor.

And if you are depending on ColdFusion for your job, then you can't afford to let your CF development methods die on the vine.

You're making a high-stakes bet that everything is going to be OK using the same old app creation ways in that one language — forever.

All it would take is for your fellow CF developer to quit or for your CIO to decide to leave the (falsely) perceived sinking ship of CFML and you could lose everything—your project, your hard-won CF skills, and possibly even your job.

Luckily, there are a number of simple, logical steps you can take now to protect yourself from these obvious risks.

No Brainer ColdFusion Best Practices to Ensure You Thrive No Matter What Happens Next

ColdFusion Alive Best Practices Checklist

Modern ColdFusion development best practices that reduce stress, inefficiency, project lifecycle costs while simultaneously increasing project velocity and innovation.

√ Easily create **a consistent server architecture** across development, testing, and production

√ A **modern test environment** to prevent bugs from spreading

√ **Automated continuous integration** tools that work well with CF

√ A **portable development environment** baked into your codebase… for free!

Learn about these and many more strategies in our free **ColdFusion Alive Best Practices Checklist**.

Michaela Light is the host of the CF Alive Podcast and has interviewed more than 100 ColdFusion experts. In each interview, she asks "What Would It Take to make CF more alive this year?" The answers still inspire her to continue to write and interview new speakers. Michaela has been programming in ColdFusion for more than 20 years. She founded TeraTech in 1989. The company specializes in ColdFusion application development, security and optimization. She has also founded the CFUnited Conference and runs the annual State of the CF Union Survey.