DEBUGGING & ERROR HANDLING IN COLDFUSION

Charlie Arehart, Independent Consultant

CF Server Troubleshooter

charlie@carehart.org

@carehart (Slack, Github, X, Fb, LI, etc.)

Updated Jun 26, 2025

Slide deck:
carehart.org/presentations

TOPICS

- What might "error handling" mean to you?
  - CF features for dealing with errors (there are several)
- What might "debugging" mean to you?
  - CF features for debugging (there are a few)
- Resources for learning more

- There is a LOT to cover, so will just skim surface for much
  - Pointing out resources for more and sharing less obvious tips along the way

Charlie Arehart, CArehart.org
@carehart

- 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, online calendar, etc at *carehart.org/consulting*
- But to be clear, I'm not selling anything in this session! ☺

- Remember: this and all DevWeek sessions are being recorded
- Again, slides for this talk available online:
  - *carehart.org/presentations*

ME.ABOUT()

Charlie Arehart, CArehart.org
@carehart

PART I: ERROR HANDLING
(SEVERAL FEATURES)

Charlie Arehart, CArehart.org
@carehart

- On the surface, the simple process of finding, understanding, resolving errors
- But more specifically that can entail:
  - Tracking when users may report errors
  - Assessing logs or other diagnostics to watch out for errors
  - Using CF features that allows detection and handling of errors
    - showing friendly output to users while perhaps providing diagnostic information to developers/tracking systems
  - Obtaining additional details about causes of errors, if possible
  - And more

WHAT MIGHT "ERROR HANDLING" MEAN TO YOU?

SOME POSSIBLE CAUSES OF ERRORS (SOME MAY BE UNEXPECTED)

- Before discussing features related to error handling, what can cause errors?
  - Mistakes in coding, such as:
    - syntax or runtime/logic/flow errors
  - Unexpected requests, such as:
    - Illegitimate ones (such as calls to "secured" pages without being logged in)
    - Malicious requests
  - Changes in configuration
    - To CF, the JVM underlying CF, the web server connector, the web server, etc.
    - Or regarding database, whether db design, db svr config, db data, etc.
  - Recent CF updates
  - And more

CF FEATURES TO HELP WITH ERRORS

- *Debugging & Logging > Debug Output Settings >* ***Enable Robust Exception Information***
  - Controls level of detail shown on-screen when errors occur, if not "handled" (demo)
- CF Logs (if accessible to you)
  - application.log
    - Shows single-line error info, with template and line number (often wrong)
  - exception.log
    - Shows multi-line error info, including stack trace which has CORRECT template/line number
  - coldfusion-out.log
    - Aggregates log lines from many CF logs, including application.log
  - coldfusion-error.log
    - Rarely has info on CF page request errors, but may have info on other errors
- These things help assess errors that HAVE happened
  - How about detecting and handling them instead, giving friendlier result to users?

Charlie Arehart, CArehart.org
@carehart

CF ERROR HANDLING

- CF Error handling allows detection of errors, control of what users see
  - Can also do additional handling (like storing error in DB, logging, etc.)
- Server-level
  - CF Admin: *Server Settings* > *Settings* > *Site-wide error handler*
  - Gotcha: can only name **relative** path, expects to be found in cfusion/wwwroot
- Application-level
  - `<CFError type="exception" template="apperrorhandler.cfm">` in Application.cfm (or cfc)
  - `OnError` method in Application.cfc
- Code-level
  - `<CFError>` (yes, can be set in a template, to affect just that template at runtime)
  - `<CFTry/CFCatch>` or script-equivalent

Charlie Arehart, CArehart.org
@carehart

CF ERROR HANDLING (CONT.)

- Some additional points:
  - Order of handling: try/catch, then app-level, then admin-level
    - If there's an error in a lower-level handler, the error is passed up to next level
  - Handled errors don't create log entries in application.log, exception.log, etc.
    - Except with site-wide error handler
    - You really **should** consider doing that with CFLOG/writelog, if not stored in a db
  - Compile-time vs runtime errors:
    - Only admin (site-wide) error handler can "handle" compile-time errors
    - cftry/cfcatch, try/catch, onerror, and cferror CANNOT (as they are themselves run-time)
    - If no site-wide handler, the compile-time error is shown on-screen (AND logged)
  - "*Robust exception handling*" setting has **no impact** on info available in error handlers

CF ERROR HANDLING (CONT.)

- There are error handling frameworks that work in or with CFML
  - They can help handle errors in a more well-organized manner
  - See my resource list at *cf411.com/error*
- Resources:
  - *helpx.adobe.com/coldfusion/developing-applications/developing-cfml-applications/handling-errors.html*
  - I have four-part series on error handling from…a LONG time ago…still relevant
    - *carehart.org/articles/#2000_10*
- Let's move on to handling 404's/missing template requests

Charlie Arehart, CArehart.org
@carehart

CF MISSING TEMPLATE HANDLING

- Server-level
  - CF Admin: *Server Settings* > *Settings* > Missing Template Handler
    - Same as site-wide error handler: must specify relative path to fie in cfusion/wwwroot
- Application-level
  - `onMissingTemplate` method of Application.cfc
- cferror and onerror do NOT detect missing requested template
  - Discussion to this point is about user making request for a file that doesn't exist
    - But what if you cfinclude/include a file that doesn't exist?
  - Both CAN detect a missing **include** file, as can try/catch: as `MissingInclude` exception

Charlie Arehart, CArehart.org
@carehart

BONUS: IIS "ERROR PAGES" SETTINGS

- An available IIS setting can keep you from seeing all error details
  - Is designed to prevent sharing "detailed errors" to end users by default
  - See "*Error Pages*" feature, at IIS server or site level, then "*Edit feature settings*"
    - Consider its "*Detailed errors for local requests and custom error pages for remote requests*"
- Related: CF web server config (wsconfig) setting, *iis_skip_custom_errors_enable*
  - In *[cf]/config/wsconfig/[nn]/isapiredirect.properties*
  - Default is false
  - When using CF missing template handler, causes IIS to not show its built-in ("custom") 404 error page
- Some people modify IIS error pages to point to a cfm file (beyond scope today)

Charlie Arehart, CArehart.org
@carehart

- CF monitoring tools offer error tracking
  - If you may use CF's PMT (free since CF2018, but only optionally installed):
    - It offers an *Errors* graph on the *CF Server* page
    - And you can drill in to see requests that had errors, and see the error message
  - If you may use FusionReactor (commercial with free trial and dev edition):
    - It offers *Requests > Error History* page, showing most recent 100 errors (by default)
    - Can drill into any request with errors, details shown as available "error" tab
- Rather than just wait for errors, you can anticipate some via code analysis tools:
  - CF Admin code compat analyzer
  - CF Package Manager "scan" capability (identifies packages needed for code)
  - CF Builder Security Analyzer (identifies security code issues, including potential errors)
  - Fixinator (commercial, fixinator.app): offers both code security and now compat checking

BONUS ERROR HANDLING TOPICS

- Finally, some problems aren't so obviously about "cf" or "your code"
  - It may be a failure of some communication between browser and server
- All modern browsers offer a helpful "developer tools" set of features
  - Within that is a very useful "Network" tab that can show valuable info
- Elaborating on this is beyond the scope of this talk
  - Just consider looking into it. Helpful tutorial here:
    - *developer.chrome.com/docs/devtools/network*
- Can also help with understanding CSS and HTML layout, JS issues, and more
  - See top level of that tutorial for more on these and many more devtools features

LEVERAGING BROWSER DEV TOOLS

Charlie Arehart, CArehart.org
@carehart

PART II: DEBUGGING (3 KINDS)

Charlie Arehart, CArehart.org
@carehart

WHAT MIGHT "DEBUGGING" MEAN TO YOU?

- On surface, the simple process of trying to understand how app/code flow works
- But different folks may regard the topic as being about:
  - The CF debug output which can optionally appear at bottom of CF pages
  - The process of adding code to show diagnostic info on a page or in logs
  - The ability to step through code line-by-line, seeing variables, etc. in an editor
  - Any other definitions/expectations?
- Each of the 3 above are possible with CF
  - Some are better suited to production, others better suited to development

DEBUGGING: A) DEBUG OUTPUT

- What is CF "debug output"? Quick demo, for any unfamiliar, and review of output, queries
- How to enable/disable in CF Admin:
  - *Debugging & Logging > Debug Output Settings > Enable Request Debugging Output*
- Can limit by IP address, within CF Admin:
  - *Debugging & Logging > Debugging IP Addresses*
  - If no value set, debug output shown to ALL users (not suitable for production)
  - Beware: "localhost" may resolve to 127.0.0.1 or ::1, depending on your OS setup
    - Since 2019 CF updates, can specify ipv6 values in that CF Admin setting
    - *carehart.org/blog/2018/8/24/fixing_CF_debugging_output_for_ipv6_localhost*

Charlie Arehart, CArehart.org
@carehart

DEBUGGING: A) DEBUG OUTPUT (CONT.)

- How to control whether debug info is shown on given page/pages:
  - `<cfsetting showdebugoutput="false">` (or script equivalent: see *cfscript.me*)
  - This can be set in a given CFML template, or in Application.cfc/cfm
- Debug output doesn't help with CF pages producing pdf's, spreadsheet's, or json output
  - Indeed, can even lead to trouble if present, in some cases
  - Debug options coming up could help in such cases
- Resources:
  - *helpx.adobe.com/coldfusion/developing-applications/developing-cfml-applications/debugging-and-troubleshooting-applications.html*

- Again, you may opt to add code showing diagnostic info, either on-page or in logs
- Can be as simple as *cfoutput/writeoutput*, *cfdump/writedump*
  - Beware that if a cfc or function is defined with output="false", such output will not appear
- Or can write to CF logs with *cflog/writelog*
- Both cfdump/cflog have more capability than many realize: explore them
- Little-known *isdebugmode()*, introduced in CF4, returns true if debug output enabled
  - Could use that to determine when to output such additional diagnostics
- There are also various related tags (or script equivalents): *cftrace*, *cftimer*, etc.
- We have far more to cover so I leave these as things you can explore on your own

DEBUGGING: B) OUTPUTTING DIAGNOSTICS

- Many languages have long offered line-by-line/step debugging
  - Many CFers feel left out
  - But guess what: it's been possible since CF4!
- Originally was offered in CF Studio (later HomeSite+)
  - I did a talk in 1999, PDF at *carehart.org/presentations/#6*
- Then CF8 made it possible via ColdFusion Builder (the Eclipse-based version)
  - I did a chapter in CFWACK on that, PDF here *carehart.org/articles/#2010_2*
  - *helpx.adobe.com/coldfusion/developing-applications/developing-cfml-applications/using-the-coldfusion-debugger.html*
- Now new (free) Adobe CFBuilder extension for VSCode offers CFML step debugging
  - helpx.adobe.com/coldfusion/coldfusion-builder-extension-for-visual-studio-code/debug-applications.html

DEBUGGING: C) STEP DEBUGGING

Charlie Arehart, CArehart.org
@carehart

STEP DEBUGGING: CF ADMIN SETUP

- With any of the editors/IDEs, CF step debugging can be challenging to setup
  - Must configure BOTH the editor AND your CF server (and any firewall)
- In CF Admin:
  - Must enable step/line debugging
    - *Debugging & Logging > Debugger Settings > Allow Line Debugging*
  - Must enable RDS: *Security > RDS > Enable RDS Service*
    - Then configure an RDS password there
  - Then must restart CF
- Then in CFBuilder, several more steps…

- Must implement the CFBuilder Extension, of course, then open it
- There you must configure a CF "server", pointing to your CF instance
  - Defining how to access your CF instance from the editor (domain/ip, port, RDS password)
  - *helpx.adobe.com/coldfusion/coldfusion-builder-extension-for-visual-studio-code/add-coldfusion-builder-vscode-extension.html* (that's not a mistaken URL)
- Then you must define a CFB "project" holding your CF code, which must be within a VSCode "workspace"
  - And you must edit that project to connect it to your "server" defined above
  - *helpx.adobe.com/coldfusion/coldfusion-builder-extension-for-visual-studio-code/project-manager.html*
- Then you must open a cfm or cfc file as found within that project
- Then you can set breakpoint(s), then turn on debugger
  - helpx.adobe.com/coldfusion/coldfusion-builder-extension-for-visual-studio-code/debug-applications.html
- Then you can either request page in browser or use "run" or "debug" buttons in CFB

STEP DEBUGGING: CF BUILDER SETUP

Charlie Arehart, CArehart.org
@carehart

STEP DEBUGGING: GOTCHAS

- There are so many moving parts that it can be hard to get working first time
  - I did preso on common problems related to old (Eclipse) CFBuilder
    - Concepts still apply broadly to new (VSCode) CFBuilder
    - *carehart.org/presentations/#cfb_debugger_setup*
- Some quick tips:
  - Try using CFB feature to "run" the CF template first: if it can't "run", it can't "debug"
    - IF code is in folder not served by the url defined for "server", use "URL prefix" feature in "Server" definition to map paths
  - If you may sit at a breakpoint long/step through for long time, request may timeout
    - Can increase CF request timeout either in CF Admin Settings page
      - or `<cfsetting requesttimeout="nn">` to set timeout for nn seconds
  - If CFB and CF are on separate machines:
    - May need to configure "debug mappings" (praying hands icon at bottom right of VSCode)
    - Also need to ensure the port for RDS and CF Debugger port (on Admin) are available at server from machine running CFB
  - When all else fails, try restarting CFBuilder (or perhaps even your machine)
    - Sometimes when mistakes are made during first setup, a background java process opened by CFBuilder may be blocking success after correct reconfiguration

Charlie Arehart, CArehart.org
@carehart

STEP DEBUGGING: BONUS TOPICS

- Besides the CFBuilder debugger, there is also a step debugger in FusionReactor
  - Does not involve CFB or VSCode, and it doesn't rely on RDS (but is commercial)
  - I've given talks on that debugger, and there are docs and videos online
- Lucee offers step debugging via the free LuceeDebug extension for VSCode
  - It also requires some configuration of things, in Lucee and in VScode to connect them
- For those on BoxLang, there is the free BoxLang extension with step debugging

TIME TO WRAP UP: RESOURCES

- In addition to links shared so far, here are more on broad topic of CF error handling and debugging
  - *learncfinaweek.com/course/index/section/Error_Handling_and_Debugging/item/Error_Handling.html*
  - *learncfinaweek.com/course/index/section/Error_Handling_and_Debugging/item/Debugging.html*
  - *modern-cfml.ortusbooks.com/cfml-language/exception-management*

SUMMARY

- Lots of ways errors can happen, but with right tools/techniques you can know/solve
  - Error handling and logs may be most important to some…
  - debug output and step debugging to others
- I wanted to whet your appetite to consider and/or learn more about your options
- Again, my contact info for follow-up:
  - Charlie Arehart
  - charlie@carehart.org
  - @carehart (Slack, Github, X, FB, LinkedIn, etc.)
  - Slides: carehart.org/presentations
- With that, hope you enjoyed the talk!
  - Use QR code for slide deck
- …time for questions?