



COBOLworx

FOSS COBOL

Bringing Open Source COBOL to
Finance and Administration
(The Old DP Shops)



Symas[®] Corp

- Sponsor & tech support vendor: OpenLDAP
- HYC, Matt Hardin and I founded it in 1999
- Over 90 happy support customers
- We decided FOSS needs better COBOL
- COBOLworx[®]



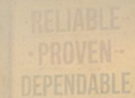
My Mainframe Creds (1972 - 1985)

- Joined IBM 1967 as programmer trainee
- Grew with the S/360
 - SysAdmin in Federal Lab, E. Fishkill chip plant
 - Insurance industry SE
 - National announcements and marketing



Too Much About That; Context Reset

- Here's Your Terminal in 1967



Big Data – 5MB on 80 column cards



Hollerith Cards Ran The Universe

- WW I and WW II: Punched Card Tech
- 1950s tech was too expensive to commercialize
- 1960s promised real availability, many vendors
- Finance and administration had no “FORTRAN”
- The US government stepped in and ...



NIST/DOD – “CODASYL” Committee

- Air Materiel Command, United States Air Force
- Bureau of Standards
- Department of Commerce
- David Taylor Model Basin Bureau of Ships
- U.S. Navy EDP Division
- Minneapolis-Honeywell Regulator Co.
- Burroughs Corporation
- International Business Machines Corporation
- Radio Corporation of America
- Sylvania Electric Products Incorporated
- Univac Division of Sperry Rand Corporation



CODASYL 1968 Report

- History of the project
 - COBOL (1959)
 - COBOL 1961 (and COBOL 1961 extended)
 - COBOL 1965 and ANS COBOL 1968
- Language guidelines for the user
- Formal, semantic description of the language



How did COBOL Fit

- FORTRAN 1954
- COBOL, LISP, and Algol Announced in 1959
- COBOL was the RENEGADE
 - Designed by Finance and Administration
 - Sponsored by ANSI for public and private sector



But Today

- 200-800B COBOL LOC
- 2 proprietary vendors (IBM, Rocket Software)
- Skills shortage
- Anguish but no budgets



COBOL Overall Attributes

- Procedural language
- Structured programming + GOTO
- Subprogramming (CALL)
- Multi-tool for Finance and Admin. programming
 - Self contained: sort, merge, intrinsic functions, etc.
 - Few externals: SQL, CICS, IMS (DL/I) ... (EXEC <foo>)



COBOL is like 9th Grade English Class

- Periods (“.”) will trip you up forever
- Long complex statements with many clauses
- OLD-SCHOOL reference format (IBM Still)
- Data definitions are fiddly and very detailed
- COPY (include) HELL! (many tabs open)



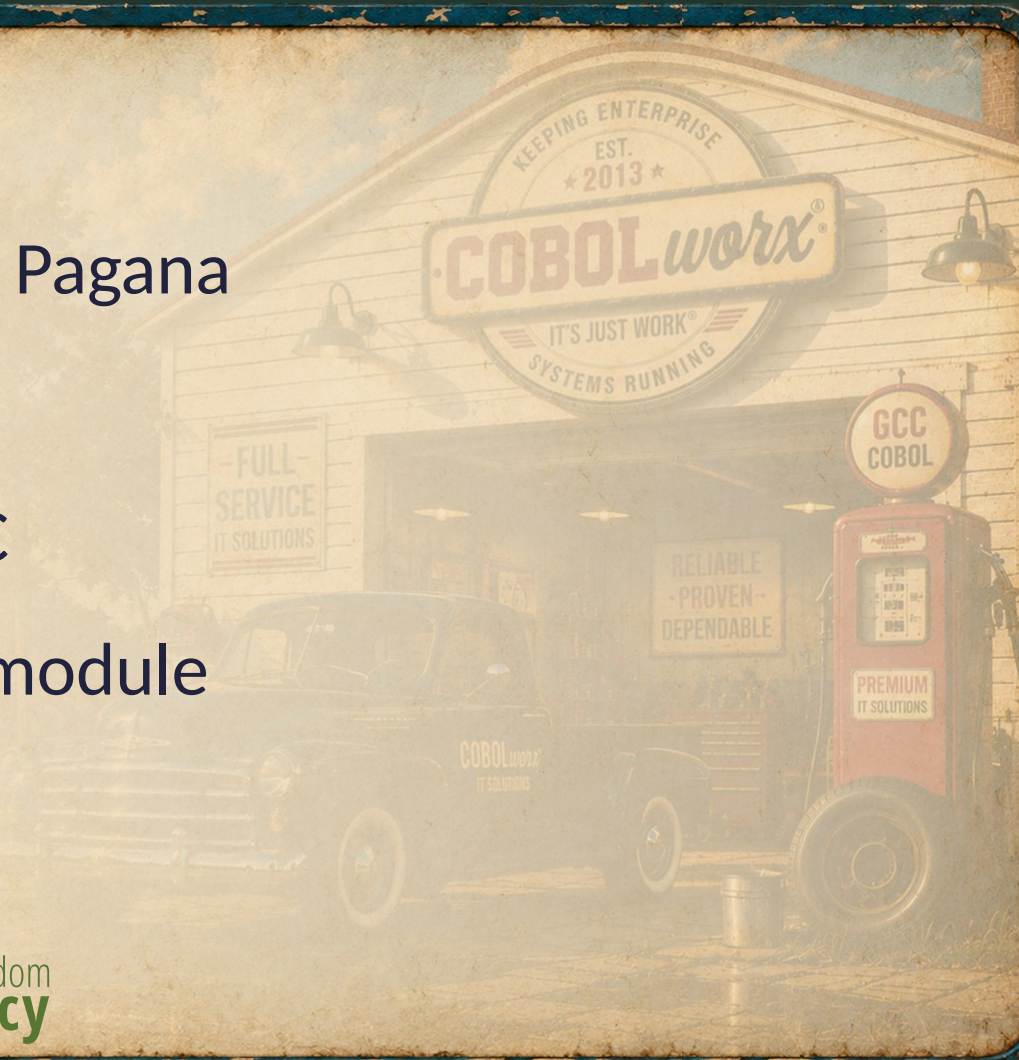
FOSS COBOL: The Short Version

- TinyCOBOL 1999
- OpenCOBOL 2001
- GnuCOBOL 2013 (OpenCOBOL moves to Gnu)
- GCC COBOL 2025



TinyCOBOL

- Original Author: Rildo Pagana
- 32-bit DOS compiler
- Translates COBOL to C
- Compiles C to native module



OpenCOBOL

- Lead developers: Keisuke Nishida, Roger Whiles
- 64-bit Windows, Linux, ... compiler
- Translates COBOL to C
- Compiles C to native module



GnuCOBOL

- Lead developers: Simon Sobisch + MANY
- Evolutionary change to OpenCOBOL
- Many dialects (ibm, mf, rm, acu, bs2000, ...)
- Still COBOL → C → modules
- Actively used though evolution has slowed



GnuCOBOL Gets Many Props!!!!

- Good coverage of the language
- Great testsuite and reliability
- Lots of COBOL dialect coverage
- COBOL → C means gdb sees C source



GnuCOBOL and cbl-gdb Break Ground

- In spite of COBOL → C artifacts
 - Bob Dubner hides the C and shows the COBOL
 - It's the FIRST source-level debugger EVER for COBOL
 - It pushed the team to “gimplify”



We contributed to GnuCOBOL 3.2

- None of us ever wrote a compiler
- No experience with GCC internals (gimple ...)
- The ISO standard is like 2,000 pages
- OF COURSE we decided to do an ISO GCC COBOL
- Exactly what y'all would do, no?



GCC COBOL (gcobol) – New Foundations

- ISO-IEC 2023 “Base”
 - Go with the standard
 - Dialects as needed
- GCC → gdb + VS/CODE and other IDEs
- Better base for future evolution



And Why GCC?

- It's already been vetted ... everywhere
- It's got a solid reputation
- PORTABILITY
- Gnu == FOSS in most peoples' minds



To tell you what most of you know

- Work started in September 2021
- Limped in to GCC 15 ... a credible limp
- Much improved COBOL in GCC 16
- Wait until they see GCC 17!
- Gdb-cobol “works” (demo to come)



MOAR COBOL

COBOL PROGRAMMER



Funny COBOL Programmer Dinosaur by jobshumor

Zazzle



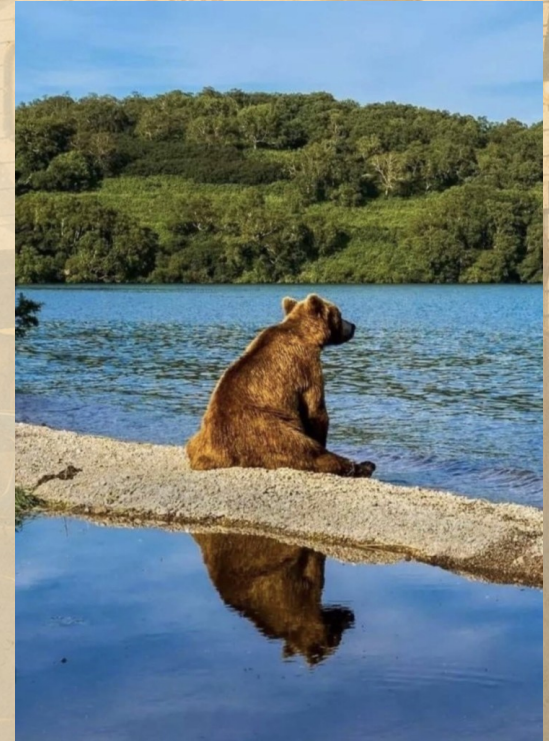
Expectations

- Everybody talks “modernization”
- The pressure to move off legacy tech is strong
- AI, transpilers, and rewrite are riskier than porting
- PS: This code is the bedrock tech. of “civilization”



Activity to Date

- Working with a contributor/user
- Just put up an improved web site
- Enjoying our friends the crickets
- But we're ready ... any day now ...
- And we're not worried



Thanks for putting up with me

