Better Object Builder for IBM i

Why BOB and how it is an improvement

v2.4.10

BOB Conceptual Overview

Functionality

- Intelligent build of QSYS objects from IFS source (manageable by git)
- Allow flexible hierarchy and naming
- Self-contained buildable project that that can be reused in many different contexts
- Minimal specification of dependencies in gmake syntax

Challenges for building programs on the IBM i

- SRC-PF
  - Fixed record length
  - Not accessible to open ecosystem, including git and make
  - 10 char names
  - All source of same type stored in QxxxSRC to avoid name conflicts (member type does not disambiguate)
- Libraries
  - Only 2 level hierarchy to organize, with only short 10 char names
- Source control
  - None – sequence number dates
  - Home grown
  - Proprietary IBM i systems
    - Cost
    - Smaller market = less investment
- Build system
  - Individual CRTXXXMOD + CRTPGM
  - CL Scripts
  - A couple of vendors have dependency-based build

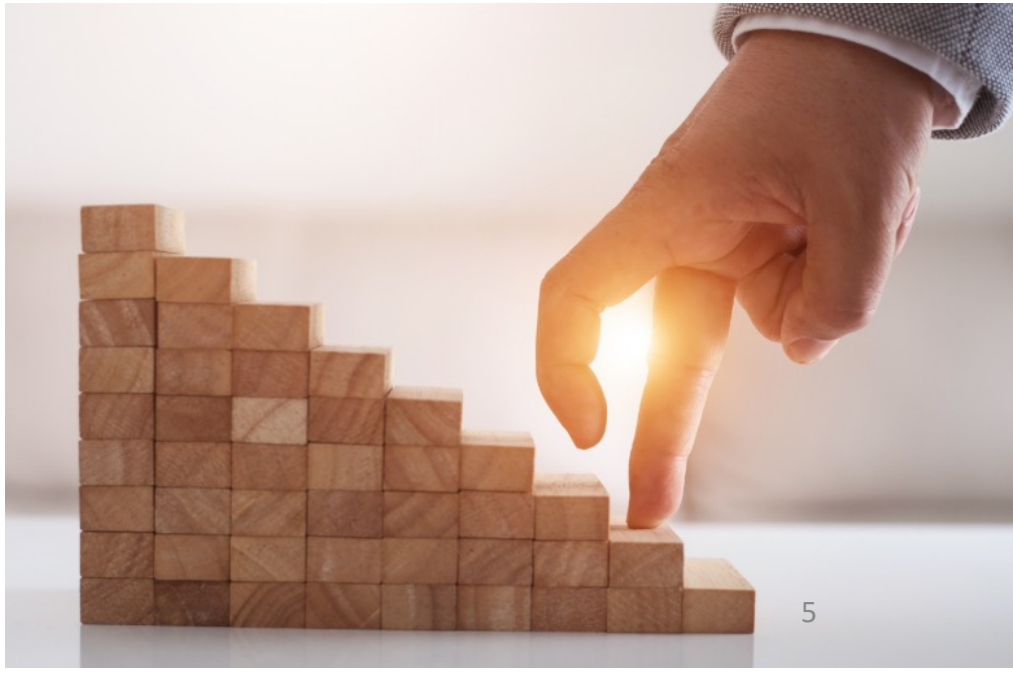
RDi i Projects - lessons learned

- Supports git but …

- Mapping from i Project to exactly one library was too inflexible
  - Some customers target many libraries from one project (program / data / source)
  - Other customers have huge libraries
- Metadata was very hard to maintain
  - Having a parallel directory hierarchy under .ibmi meant that any time a SRC-PF or member changed names the metadata was lost
- Mapping rigidly to SRC-PF meant inheriting all of its limitations
  - Fixed line length
  - Fixed directory hierarchy of basically 1 level with only 10 characters
- Build was very limited,
  - No disambiguating of PGM vs MOD
  - No understanding of binding relationships
  - No incremental ability – i.e. only build what had changed

Existing bob by S4i

- Fascinating open-source project by Jeff Berman then of S4i https://github.com/s4isystems/Bob
  - Incremental compile ability based on gmake
  - Some level of ILE binding understanding
  - No longer bound to library and SRC-PF structure and naming limitations
  - Member level specific metadata using gmake variable
  - Consideration of target EBCDIC CCSID for compiler
  - Support of old languages whose compilers do not have IFS support yet (DDS, UIM)
  - Retrieval of all EVFEVENT files to enable compiler feedback
- Limitations
  - Uppercase names required
  - Single target library
  - Single directory containing source
  - No metadata on environment prerequisites,
    i.e. LIBL, where to find includes, ASP etc.
  - Install was complex – not yum-enabled
  - No 1 to 1 mapping of file extensions to compile
    (i.e. are we targeting MOD or PGM)

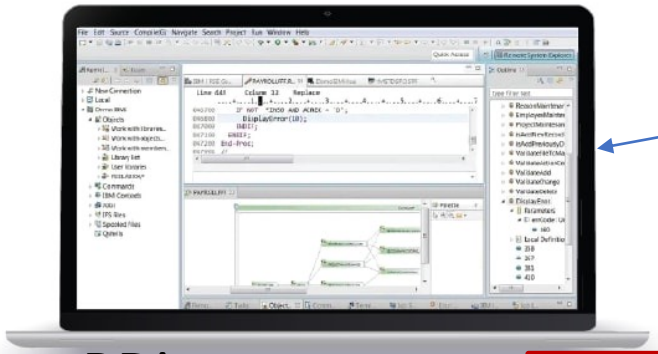


Enhancement

- Project definition
  - Know how to build yourself
  - Know where to resolve includes
  - Know how to set up environment
  - Learn from package.json
  - But make it flexible so that what is stored in git does not have to be modified from 1 developer or deployment scenario to another
- No limit on number of directories and their nesting
- No limit on directory naming
- No limit on number of object libraries
- Unambiguous mapping from file name to compile type

Use cases: Bob should be usable by …

- PASE command line
- Windows/Mac command line with rsync/scp to do file transfer
- Any VS Code extension for IBM i development like Code for IBM i (halcyon tech)
- RDi



Visual Studio Code

Merlin IDE

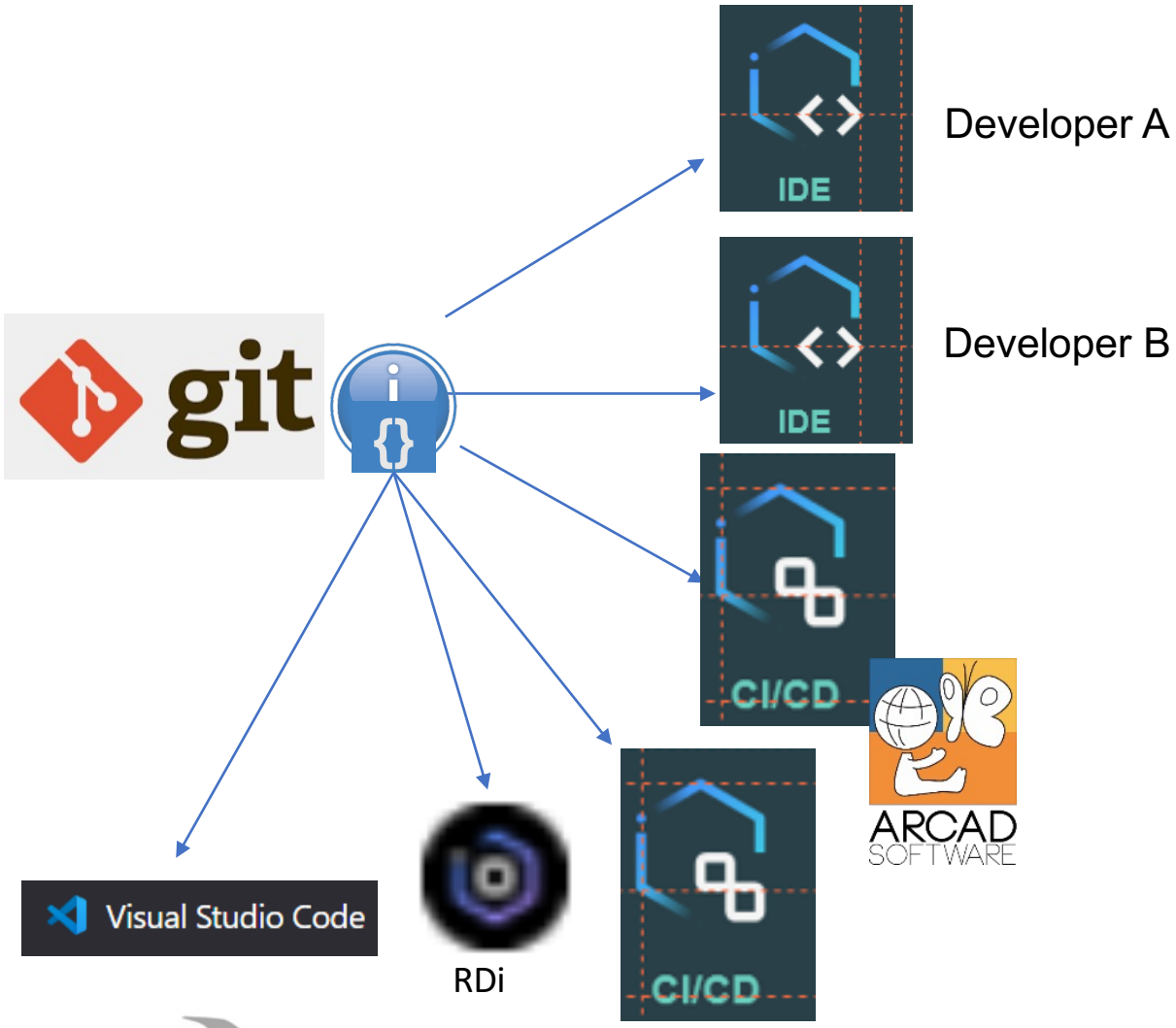
```
Windows PowerShell
-bash-5.1$ touch functionsVAT/VAT300.RPGLE
-bash-5.1$ makei build -e lib1=BOBBUILD1
Set variable <lib1> to 'BOBBUILD1'
> make -k BUILDVARSMKPATH="/tmp/tmpu2ty3v2j" -k BOB="/QOpenSys/pkgs/lib/bob" -f "/QOpenSys/pkgs/lib/bob/mk/Makefile" all
=== Creating RPG module [VAT300.RPGLE]
crtrpgmod module(BOBBUILD1/VAT300) srcstmf('/home/REINHARD/git/bob-recursive-example/functionsVAT/VAT300.RPGLE') AUT() DBGVIEW(*ALL) OPTION(*EVENTF) OUTPUT(*PRINT) TEXT('Functions VAT') TGTCCSID(500) TGTRLS()
✓ VAT300.MODULE was created successfully!

=== Creating service program [FVAT] from modules [VAT300] and service programs []
CRTSRVPGM srcstmf('/home/REINHARD/git/bob-recursive-example/functionsVAT/FVAT.BND') SRVPGM(BOBBUILD1/FVAT) MODULE(VAT300) BNDSRVPGM(*NONE) ACTGRP(*CALLER) TEXT('Functions VAT') TGTRLS() AUT() DETAIL(*BASIC) STGMDL(*SNGLVL)
✓ FVAT.SRVPGM was created successfully!

=== Create Bound RPG Program [ORD100]
CRTBNDRPG srcstmf('/home/REINHARD/git/bob-recursive-example/QRPGLESRC/ORD100.PGM.RPGLE') PGM(BOBBUILD1/ORD100) DBGVIEW(*ALL) TGTCCSID(*JOB) OPTION(*EVENTF) TEXT('') INCDIR(QPROTOSRC)
✓ ORD100.PGM was created successfully!

Objects:            0 failed 3 succeed 3 total
Build Completed!
-bash-5.1$
```

iProj – portable, git-storable project



- Single project definition that can be used in many contexts
  - Cloned into different development environments
    - Merlin
    - VS Code
    - Rdi
  - Used to define build in CI/CD pipelines
    - Buildable by ARCAD builder
    - Buildable by open source BOB
  - Has to specify requirements on external environment
    - LIBL
    - Where to resolve includes
    - ASP, OVRxxx etc.
  - But has to be configurable for each environment
    - Uses &name env variables