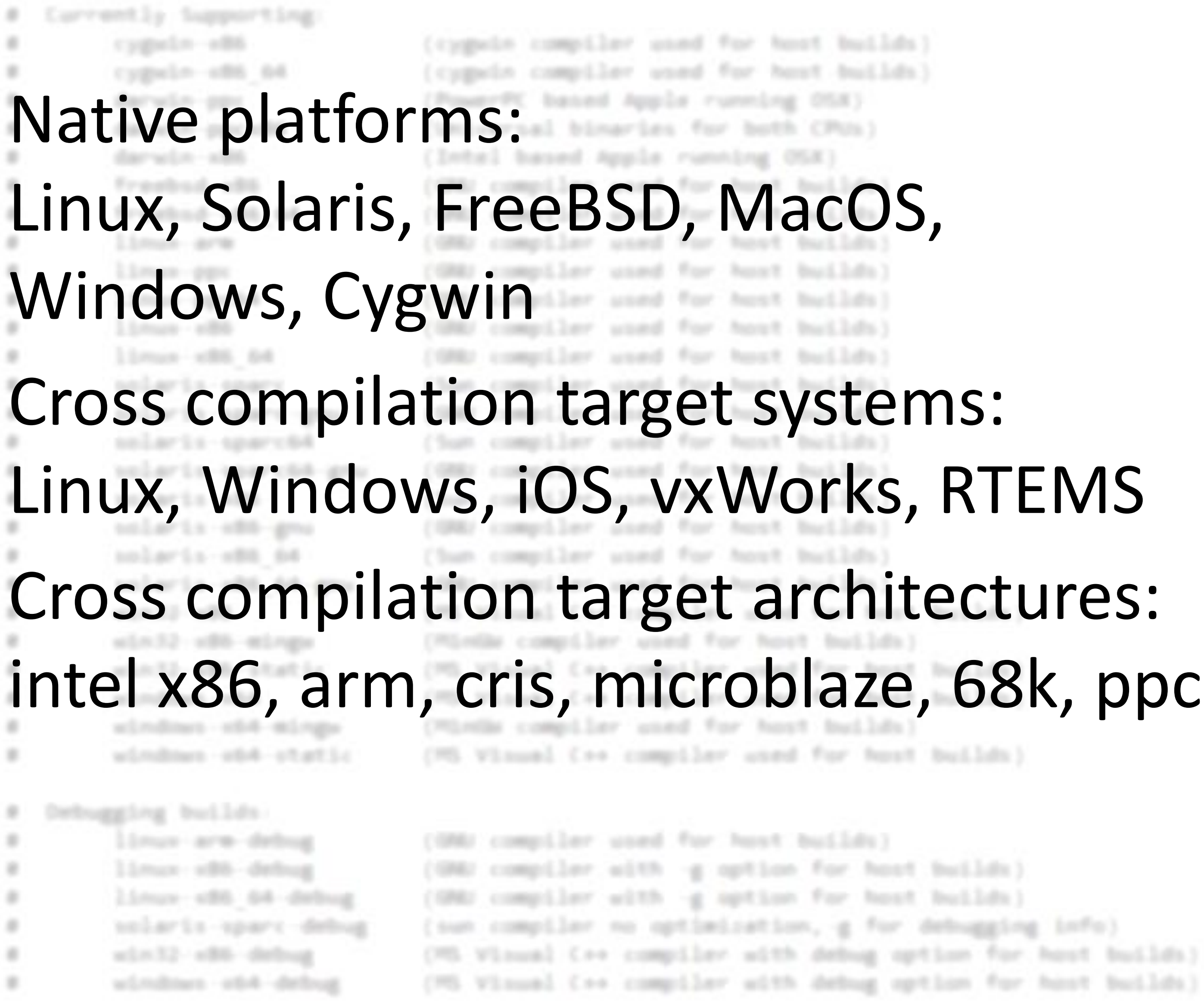
CI Setup for EPICS Base

Ralph Lange (ITER),
Andrew Johnson (APS),
Michael Davidsaver (Osprey DCS)

EPICS Platforms

- Native platforms:
  Linux, Solaris, FreeBSD, MacOS,
  Windows, Cygwin
- Cross compilation target systems:
  Linux, Windows, iOS, vxWorks, RTEMS
- Cross compilation target architectures:
  intel x86, arm, cris, microblaze, 68k, ppc

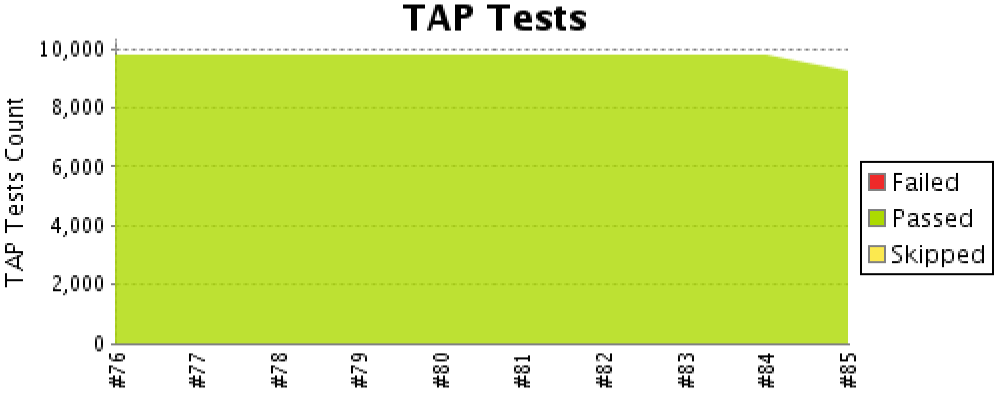


```
#   Currently Supporting:
#       cygwin-x86              (cygwin compiler used for host builds)
#       cygwin-x86_64           (cygwin compiler used for host builds)
#       darwin-ppc              (PowerPC based Apple running OSX)
#       [illegible]             [illegible]al binaries for both CPUs)
#       darwin-x86              (Intel based Apple running OSX)
#       freebsd-x86             (GNU compiler used for host builds)
#       [illegible]             [illegible]
#       linux-arm               (GNU compiler used for host builds)
#       linux-ppc               (GNU compiler used for host builds)
#       [illegible]             [illegible]iler used for host builds)
#       linux-x86               (GNU compiler used for host builds)
#       linux-x86_64            (GNU compiler used for host builds)
#       solaris-sparc           [illegible]
#       [illegible]             [illegible]
#       solaris-sparc64         (Sun compiler used for host builds)
#       solaris-sparc64-gnu     (GNU compiler used for host builds)
#       [illegible]             [illegible]
#       solaris-x86-gnu         (GNU compiler used for host builds)
#       solaris-x86_64          (Sun compiler used for host builds)
#       [illegible]             [illegible]
#       [illegible]             [illegible]
#       win32-x86-mingw         (MinGW compiler used for host builds)
#       win32-x86-static        (MS Visual C++ compiler used for host builds)
#       [illegible]             [illegible]
#       windows-x64-mingw       (MinGW compiler used for host builds)
#       windows-x64-static      (MS Visual C++ compiler used for host builds)

#   Debugging builds:
#       linux-arm-debug         (GNU compiler used for host builds)
#       linux-x86-debug         (GNU compiler with -g option for host builds)
#       linux-x86_64-debug      (GNU compiler with -g option for host builds)
#       solaris-sparc-debug     (sun compiler no optimization, -g for debugging info)
#       win32-x86-debug         (MS Visual C++ compiler with debug option for host builds)
#       windows-x64-debug       (MS Visual C++ compiler with debug option for host builds)
```

Unit Tests in Base

- Number of tests and test coverage are increasing
  3.14: **2.6k** → *3.15:* **8.5k** → *3.16:* **9.2k**
- Test output is formatted as TAP (Test Anything Protocol)

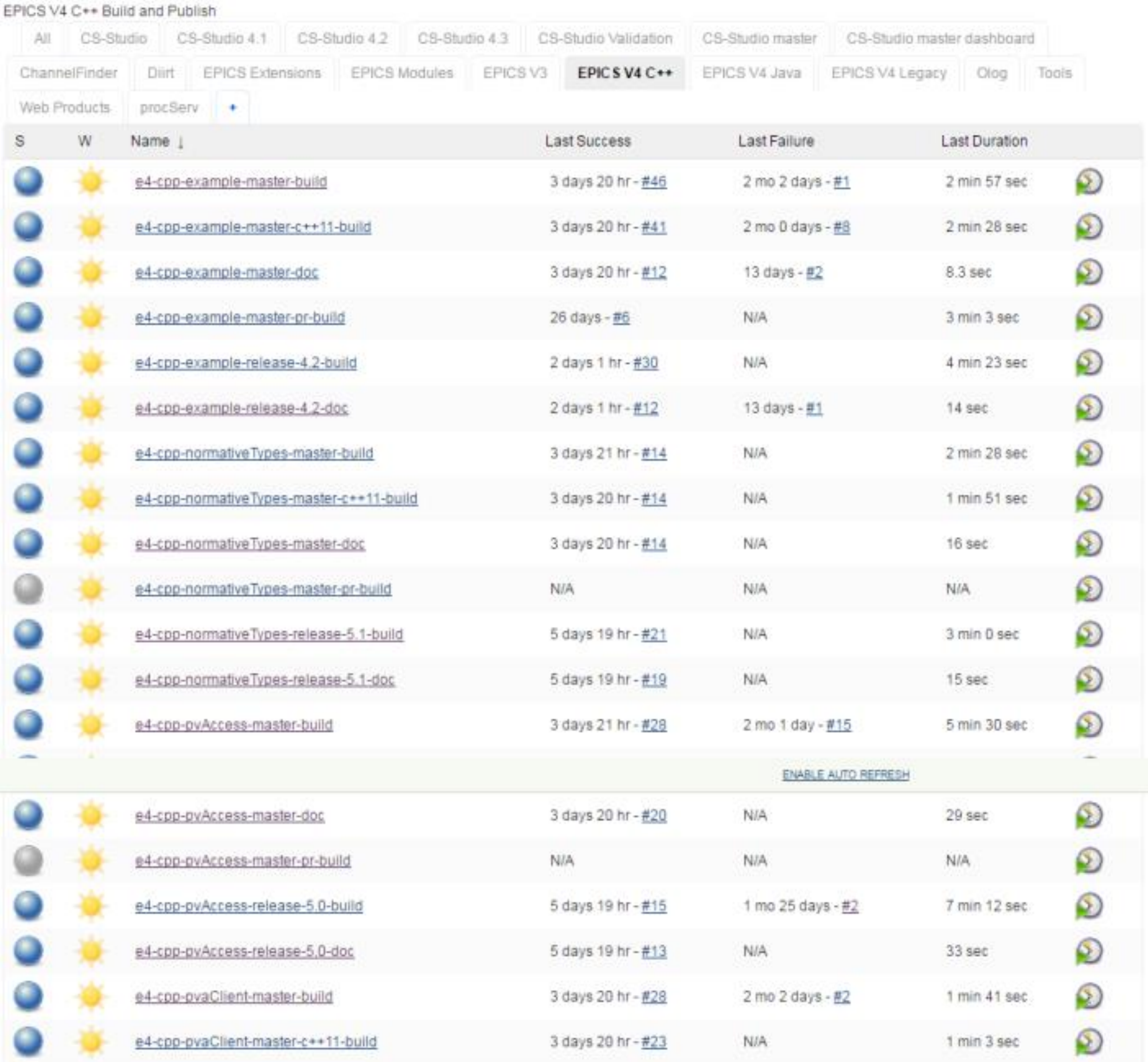


Java Parts of V4

- Maven as build tool
- Two layers of Maven submodules: Aggregation and inheritance
- Parent project contains Git submodules
  - Does not play well with some Maven plugins
- Deployment to Maven Central

Two Jenkins Instances

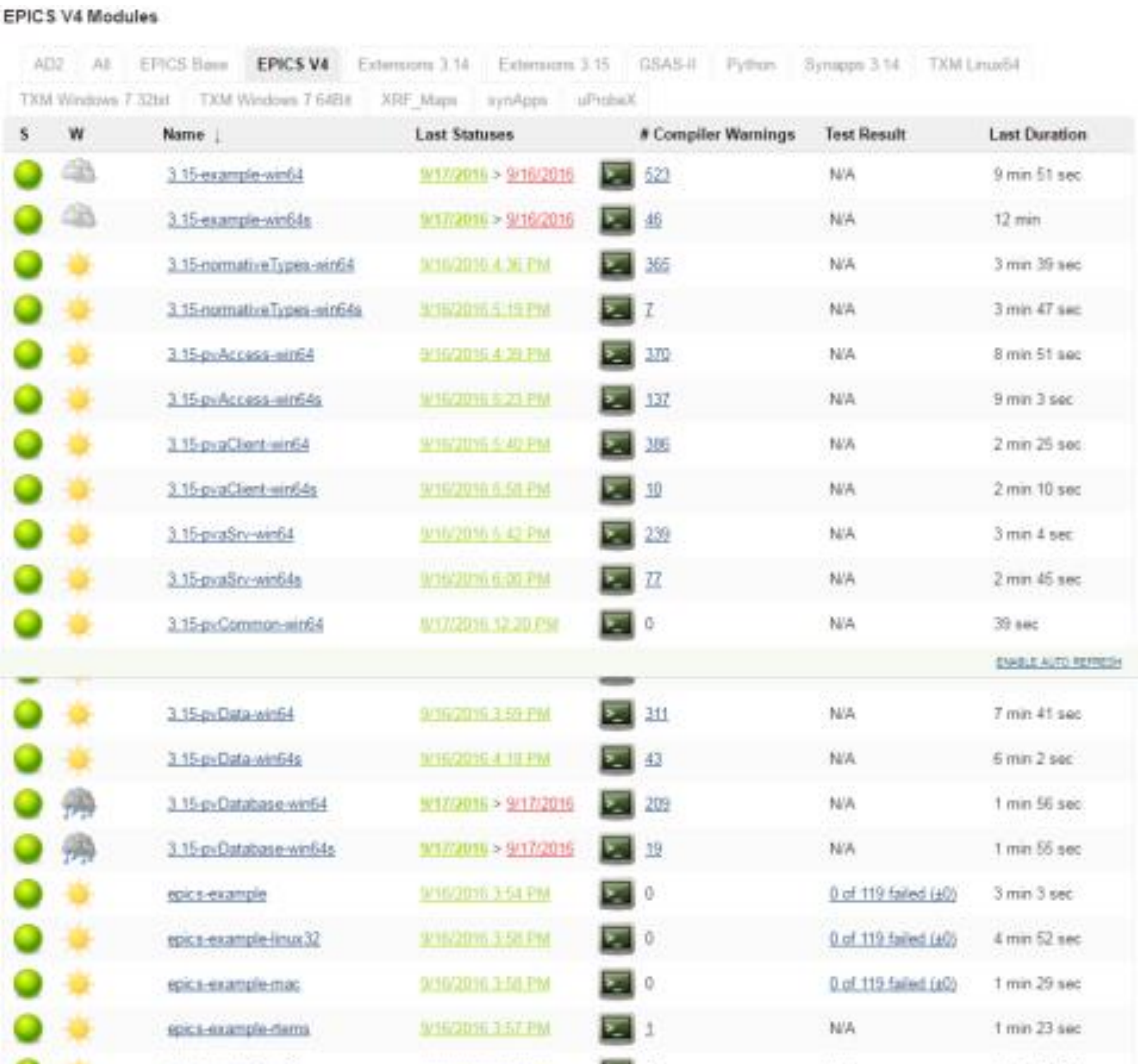
- **CloudBees**:
  - C++ and Java
  - Linux host
  - Release builds
  - Pull request builds



S	W	Name ↓	Last Success	Last Failure	Last Duration
●	☀	e4-cpp-example-master-build	3 days 20 hr - #46	2 mo 2 days - #1	2 min 57 sec
●	☀	e4-cpp-example-master-c++11-build	3 days 20 hr - #41	2 mo 0 days - #8	2 min 28 sec
●	☀	e4-cpp-example-master-doc	3 days 20 hr - #12	13 days - #2	8.3 sec
●	☀	e4-cpp-example-master-pr-build	26 days - #6	N/A	3 min 3 sec
●	☀	e4-cpp-example-release-4.2-build	2 days 1 hr - #30	N/A	4 min 23 sec
●	☀	e4-cpp-example-release-4.2-doc	2 days 1 hr - #12	13 days - #1	14 sec
●	☀	e4-cpp-normativeTypes-master-build	3 days 21 hr - #14	N/A	2 min 28 sec
●	☀	e4-cpp-normativeTypes-master-c++11-build	3 days 20 hr - #14	N/A	1 min 51 sec
●	☀	e4-cpp-normativeTypes-master-doc	3 days 20 hr - #14	N/A	16 sec
○	☀	e4-cpp-normativeTypes-master-pr-build	N/A	N/A	N/A
●	☀	e4-cpp-normativeTypes-release-5.1-build	5 days 19 hr - #21	N/A	3 min 0 sec
●	☀	e4-cpp-normativeTypes-release-5.1-doc	5 days 19 hr - #19	N/A	15 sec
●	☀	e4-cpp-pvAccess-master-build	3 days 21 hr - #28	2 mo 1 day - #15	5 min 30 sec
●	☀	e4-cpp-pvAccess-master-doc	3 days 20 hr - #20	N/A	29 sec
○	☀	e4-cpp-pvAccess-master-pr-build	N/A	N/A	N/A
●	☀	e4-cpp-pvAccess-release-5.0-build	5 days 19 hr - #15	1 mo 25 days - #2	7 min 12 sec
●	☀	e4-cpp-pvAccess-release-5.0-doc	5 days 19 hr - #13	N/A	33 sec
●	☀	e4-cpp-pvaClient-master-build	3 days 20 hr - #28	2 mo 2 days - #2	1 min 41 sec
●	☀	e4-cpp-pvaClient-master-c++11-build	3 days 20 hr - #23	N/A	1 min 3 sec

Two Jenkins Instances

- **APS**:
  - C++ builds
  - Only master
  - Host builds: Linux, MacOS, Solaris, Windows
  - Cross builds to many targets



EPICS V4 Modules

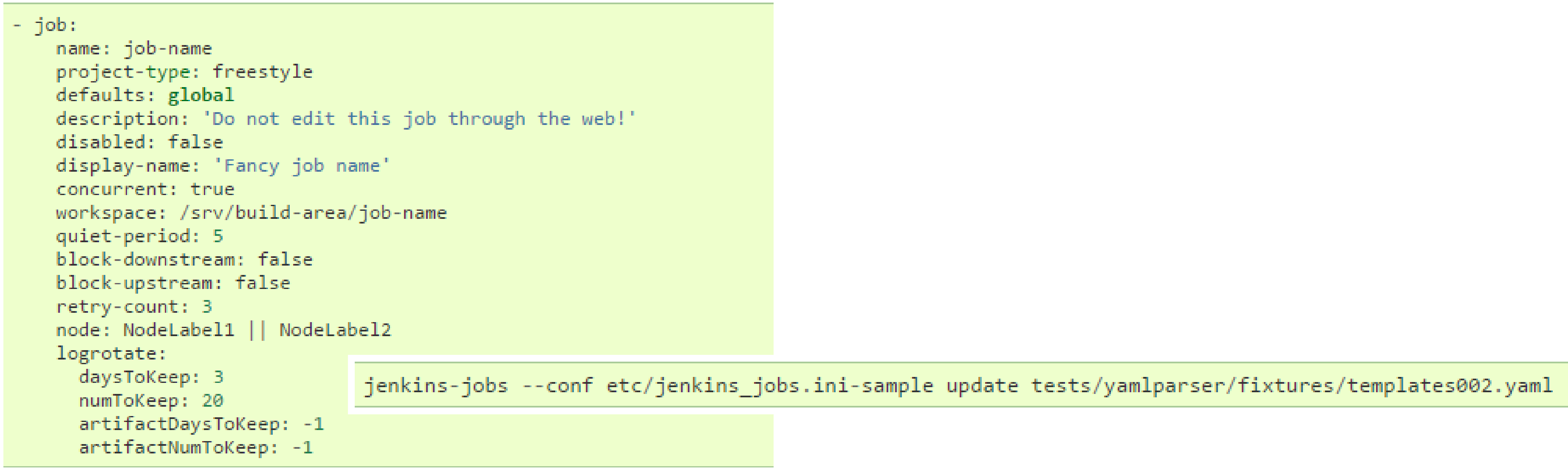
S	W	Name ↓	Last Statuses	# Compiler Warnings	Test Result	Last Duration
●	☁	3.15-example-win64	9/17/2016 > 9/16/2016	523	N/A	9 min 51 sec
●	☁	3.15-example-win64s	9/17/2016 > 9/16/2016	46	N/A	12 min
●	☀	3.15-normativeTypes-win64	9/16/2016 4:36 PM	365	N/A	3 min 39 sec
●	☀	3.15-normativeTypes-win64s	9/16/2016 5:19 PM	7	N/A	3 min 47 sec
●	☀	3.15-pvAccess-win64	9/16/2016 4:39 PM	370	N/A	8 min 51 sec
●	☀	3.15-pvAccess-win64s	9/16/2016 6:23 PM	137	N/A	9 min 3 sec
●	☀	3.15-pvaClient-win64	9/16/2016 5:40 PM	386	N/A	2 min 25 sec
●	☀	3.15-pvaClient-win64s	9/16/2016 6:58 PM	10	N/A	2 min 10 sec
●	☀	3.15-pvaSrv-win64	9/16/2016 5:42 PM	239	N/A	3 min 4 sec
●	☀	3.15-pvaSrv-win64s	9/16/2016 6:00 PM	77	N/A	2 min 45 sec
●	☀	3.15-pvCommon-win64	9/17/2016 12:20 PM	0	N/A	39 sec

ENABLE AUTO REFRESH

S	W	Name ↓	Last Statuses	# Compiler Warnings	Test Result	Last Duration
●	☀	3.15-pvData-win64	9/16/2016 3:59 PM	311	N/A	7 min 41 sec
●	☀	3.15-pvData-win64s	9/16/2016 4:10 PM	43	N/A	6 min 2 sec
●	🌧	3.15-pvDatabase-win64	9/17/2016 > 9/17/2016	209	N/A	1 min 56 sec
●	🌧	3.15-pvDatabase-win64s	9/17/2016 > 9/17/2016	19	N/A	1 min 55 sec
●	☀	epics-example	9/16/2016 3:54 PM	0	0 of 119 failed (±0)	3 min 3 sec
●	☀	epics-example-linux32	9/16/2016 3:58 PM	0	0 of 119 failed (±0)	4 min 52 sec
●	☀	epics-example-mac	9/16/2016 3:58 PM	0	0 of 119 failed (±0)	1 min 29 sec
●	☀	epics-example-rtems	9/16/2016 3:57 PM	1	N/A	1 min 23 sec

Generating CloudBees Jenkins Jobs

- CloudBees does not allow the Job DSL Plugin
  - – Script running on the server is too powerful
- Jenkins Job Builder
  - – Script creates job XML from YAML declarations (client side), then uploads to Jenkins

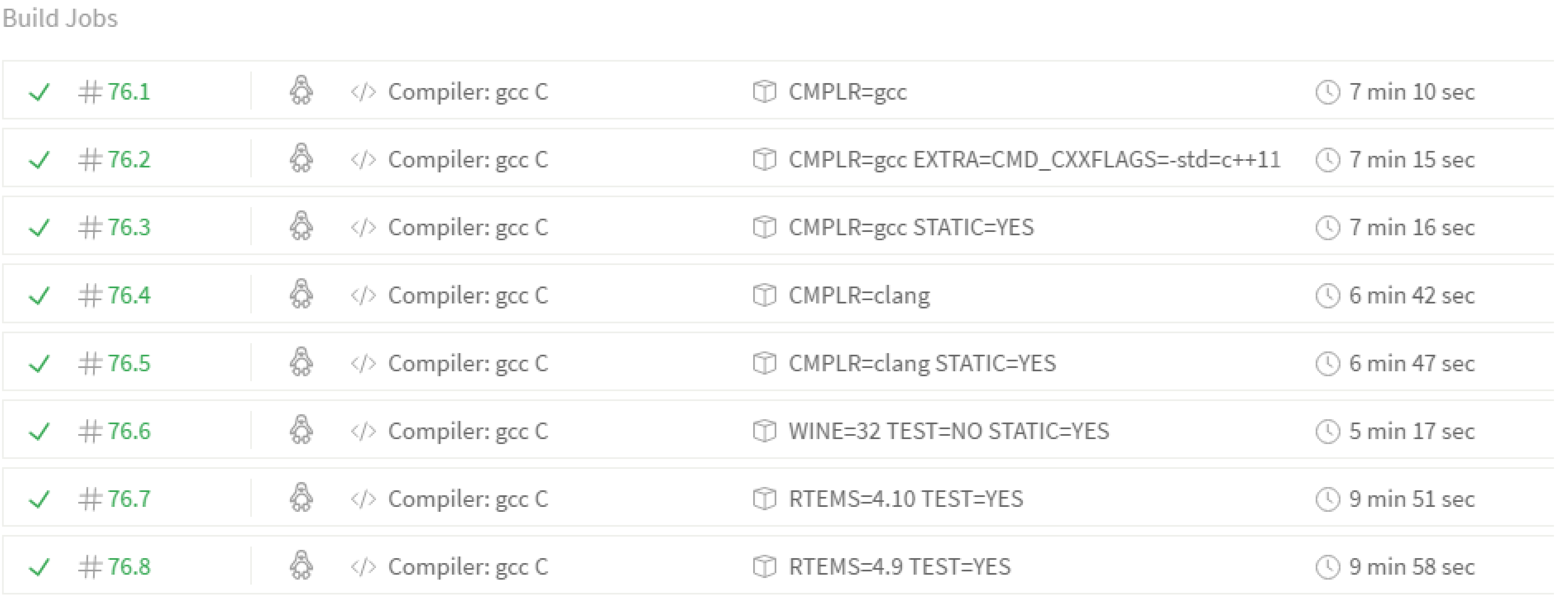


```yaml
- job:
    name: job-name
    project-type: freestyle
    defaults: global
    description: 'Do not edit this job through the web!'
    disabled: false
    display-name: 'Fancy job name'
    concurrent: true
    workspace: /srv/build-area/job-name
    quiet-period: 5
    block-downstream: false
    block-upstream: false
    retry-count: 3
    node: NodeLabel1 || NodeLabel2
    logrotate:
      daysToKeep: 3
      numToKeep: 20
      artifactDaysToKeep: -1
      artifactNumToKeep: -1
```

```
jenkins-jobs --conf etc/jenkins_jobs.ini-sample update tests/yamlparser/fixtures/templates002.yaml
```

Next Generation CI

- Travis CI – cloud based builds
  - host builds for Linux (gcc, clang), WINE
  - cross builds for RTEMS

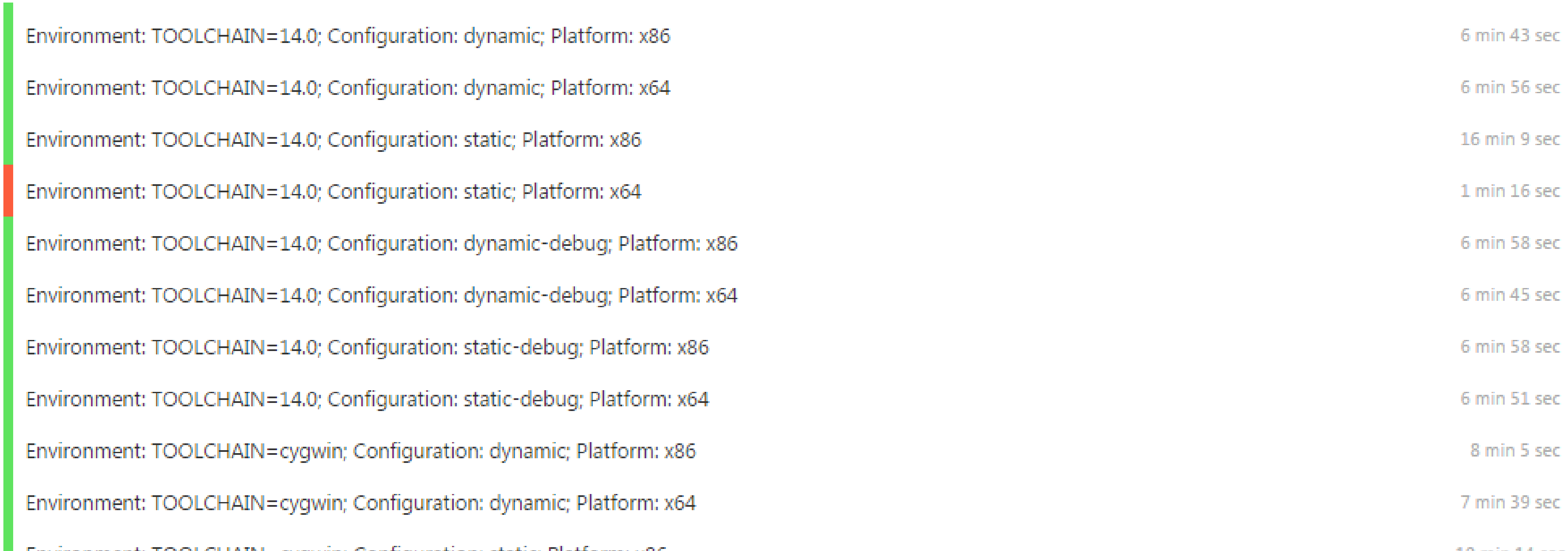


Build Jobs

✓	#76.1	Compiler: gcc C	CMPLR=gcc	7 min 10 sec
✓	#76.2	Compiler: gcc C	CMPLR=gcc EXTRA=CMD_CXXFLAGS=-std=c++11	7 min 15 sec
✓	#76.3	Compiler: gcc C	CMPLR=gcc STATIC=YES	7 min 16 sec
✓	#76.4	Compiler: gcc C	CMPLR=clang	6 min 42 sec
✓	#76.5	Compiler: gcc C	CMPLR=clang STATIC=YES	6 min 47 sec
✓	#76.6	Compiler: gcc C	WINE=32 TEST=NO STATIC=YES	5 min 17 sec
✓	#76.7	Compiler: gcc C	RTEMS=4.10 TEST=YES	9 min 51 sec
✓	#76.8	Compiler: gcc C	RTEMS=4.9 TEST=YES	9 min 58 sec

Next Generation CI

- AppVeyor: Travis for Windows
  - VS 2008, 2010, 2012, 2013, 2015, Cygwin, MinGW
  - 32bit, 64bit – DLL, static build – debug, nodebug
- Full matrix: ~50 jobs, taking ~7 hours



```
Environment: TOOLCHAIN=14.0; Configuration: dynamic; Platform: x86          6 min 43 sec
Environment: TOOLCHAIN=14.0; Configuration: dynamic; Platform: x64          6 min 56 sec
Environment: TOOLCHAIN=14.0; Configuration: static; Platform: x86          16 min 9 sec
Environment: TOOLCHAIN=14.0; Configuration: static; Platform: x64           1 min 16 sec
Environment: TOOLCHAIN=14.0; Configuration: dynamic-debug; Platform: x86    6 min 58 sec
Environment: TOOLCHAIN=14.0; Configuration: dynamic-debug; Platform: x64    6 min 45 sec
Environment: TOOLCHAIN=14.0; Configuration: static-debug; Platform: x86     6 min 58 sec
Environment: TOOLCHAIN=14.0; Configuration: static-debug; Platform: x64     6 min 51 sec
Environment: TOOLCHAIN=cygwin; Configuration: dynamic; Platform: x86         8 min 5 sec
Environment: TOOLCHAIN=cygwin; Configuration: dynamic; Platform: x64        7 min 39 sec
Environment: TOOLCHAIN=cygwin; Configuration: static; Platform: x86        10 min 14 sec
```

Conclusion

- Build systems for both C++ and Java are in a good state
- Test coverage is good and improving
- Current Jenkins based CI systems already cover a lot
- Work to be done for next generation CI tools
- Please copy and contribute