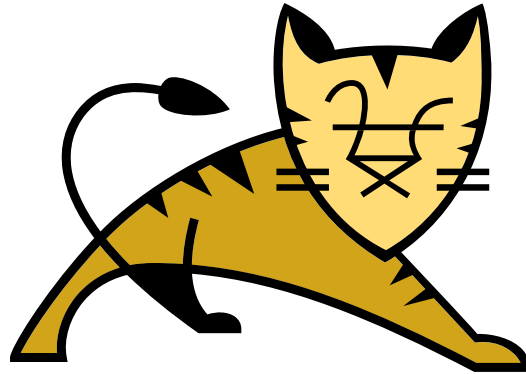


Getting Started Hacking Tomcat



Christopher Schultz
Total Child Health, Inc.
ASF Member, Tomcat PMC, Security Team

QR for Slides:



Why Hack Tomcat?

- Why not?
- Fix a bug
- Add a feature
- Find a vulnerability
- Academic interest
- Help the Tomcat community

Getting Started

- Get dependencies (Java, git, svn, ant)
- Get Tomcat source
- Build binaries
- Build docs / web site

Getting Started

- Revision-Control Dependencies
 - git
 - From you package manager (e.g. `yum install git` / `apt-get install git`)
 - Binaries / source from <https://git-scm.com/downloads>
 - Subversion
 - From your package manager (e.g. `yum install subversion` / `apt-get install subversion`)
 - Only needed for web site / documentation

Getting Started

- Build Dependencies
 - Apache ant
 - Binaries from `https://ant.apache.org/bindownload.cgi`

Getting Started

- Tomcat Source

- Java Sources

- From GitHub: `https://github.com/apache/tomcat`

- From ASF: `https://gitbox.apache.org/repos/asf?p=tomcat.git`

- Documentation

- From `https://svn.apache.org/repos/asf/tomcat/site/trunk`

Getting Started

- Quick script for the impatient
 - See the script `hurry.sh` alongside these slides
 - Requires git, Java
 - Downloads, unpacks Apache ant if necessary
 - Clones Tomcat's master branch

Building Tomcat

- `$ ant`
 - Builds a working Tomcat
 - ... into `${working}/output/build`

- **Starts as usual**

```
$ cd output/build
```

```
$ ./bin/catalina.sh start
```


Building Documentation

- Different repository; in Subversion
 - svn checkout
<https://svn.apache.org/repos/asf/tomcat/site/trunk>
 - It's *big* (1GiB+)
 - ant
 - Builds documentation into `docs/` directory

Now What?

- Run unit tests
 - To help test a release build
 - To test a change you are making to source

Running the Unit Tests

- Ant test
 - Takes a while (~25 minutes on my 2018 MBP)

- Some settings you might want in **build.properties**

```
test.threads=[ something > 1 ] (threads=8 → 5 min on my MBP)
execute.test.apr=false
execute.test.nio=false
execute.test.nio2=false
```

Running the Unit Tests

- Unit test logs

- Found in `output/build/logs/TEST-[class].[IOtype].txt`

- Running just *one* unit test

```
$ ant -Dtest.entry=org.apache.catalina.valves.TestRemoteIpValve \  
test
```

Configuring Logging

- <http://tomcat.apache.org/tomcat-9.0-doc/logging.html>
- TLDR; edit `conf/logging.properties`
 - Use `java.util.logging-style` configuration, *plus*
 - `org.apache.catalina.core.ContainerBase.[engine].[host].[context]`

Configuring Logging

- **Example: send all Realm logs to realm.log**

```
handlers = ....., realmLog
realmLog.directory=${catalina.base}/logs
realmLog.prefix=realm
realmLog.level=FINEST
org.apache.catalina.realm.level=FINEST
org.apache.catalina.realm.handlers=realmLog
```

Configuring Logging

- Example: send your webapp's logs to `app.log`

```
handlers=..., appLog
appLog.directory=${catalina.base}
appLog.prefix=app
appLog.level=FINEST
org.apache.catalina.core.ContainerBase.Catalina.localhost.myapp.handlers=appLog
com.yourapp.level=FINEST
com.yourapp.handlers=appLog
```

Other Useful Build Settings

- OpenSSL

```
base.path=[ somewhere ]
```

```
execute.validate=false
```

```
test.openssl.path=/usr/local/Cellar/  
openssl@1.1/1.1.1b/bin/openssl
```

```
test.openssl.unimplemented=ARIA,IDEA,GOST
```


Configuring for IDEs

- Prep Tomcat's working directory

```
$ ant ide-eclipse
```

```
$ ant ide-intellij
```

```
$ ant ide-netbeans
```

- Import → IDE as existing project

Working with a Debugger

- Startup Tomcat in remote-debugging mode

```
$ ./bin/catalina.sh jpda start
```

- Attach IDE to JPDA port :8000
 - Eclipse: Run > Debug Configurations > Remote Java Application > New
 - Choose Project
 - Set JPDA connection info (localhost:8000)

Use the Debugger

- Just as you would normally use a debugger
 - Set breakpoints, then
 - Inspect
 - Step
 - Resume
 - Live-edit code

Building tcnative

- Fetch from repo

```
$ git clone https://github.com/apache/tomcat-native
```

- Build

```
$ cd tomcat-native/native
```

```
$ ./configure --with-apr=[...] --with-ssl=[...] \  
    --with-java-home=[...] --exec-prefix=NONE
```

```
$ make
```

Install tcnative Binaries

- After make, binaries are in tomcat-native/.libs
- Copy libs to `${catalina.home}/lib`
- Run Tomcat with `APRLifecycleListener` enabled
 - This is the default
 - Provides APR connector + JSSE+OpenSSL

Submitting a Patch

- Multiple ways to submit
 - Pull Request via GitHub
 - <https://github.com/apache/tomcat/>
 - Patch attached to Bugzilla issue
 - <https://bz.apache.org/bugzilla/>
- Do not email committers directly

Submitting a Patch

- Small patches are best
- Multiple patches/commits are helpful
 - Patch/commit 1: re-factor existing code
 - Patch/commit 2: add new feature
- Code comments are *very* helpful
- Please respect the coding conventions

How Can *You* Help?

- Test release candidates
 - Look for [VOTE] threads on dev@ mailing list
 - Download, build, run unit tests, test your webapp
 - Reply to the [VOTE] thread on dev@
 - Note: your vote will be termed “non-binding”, but will appear in the vote results
 - ...but your negative vote may block the release (which is *good!*)

How Can *You* Help?

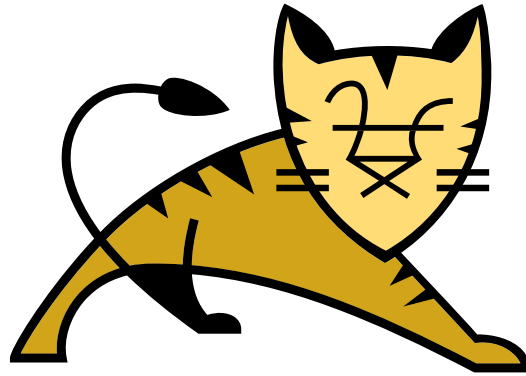
- Report Bugs
 - Bugzilla
 - GitHub Issue, though BZ is better for us
- Report Security Bugs
 - To security@tomcat.apache.org
 - Please don't put in Bugzilla/GitHub

How Can *You* Help?

- Improve code quality
 - SpotBugs, IDE warnings,
- Improve documentation
 - Typos, formatting, layout, etc.
 - Add missing information or clarify confusion
- Add localizations into your native language

How Can *You* Help?

- Add New Features
 - Optimizing startup time
 - Delay 100-Continue handling until webapp reads
 - Multi-threaded @Annotation scanning
- Expand Feature Support
 - Control WebSocket session read/write idle timeouts
 - Expand JASPIC Support



Questions



[https://people.apache.org/~schultz/presentations/ApacheCon NA 2020/Getting Started Hacking Tomcat.pdf](https://people.apache.org/~schultz/presentations/ApacheCon%20NA%202020/Getting%20Started%20Hacking%20Tomcat.pdf)
Sample code available in the same directory.