

Writing Good Commit Messages

Typical format

1-line summary

<blank line>

Details

as many

as needed

Log of commits (web version)

Commits on May 4, 2022

Improving the 64-bit number formatting to better match the 32-bit alg... [...](#)

 tannergooding committed 12 hours ago ✗

Verified



874c6a9



Update message for unsupported char marshalling (#68865)

 elinor-fung committed 13 hours ago ✓


Verified



881231a



Add a four-element AsyncLocalValueMap type (#68790) [...](#)

 stephentoub committed 14 hours ago ✓


Verified



16b6369



Stop escaping ' and " in generated regex XML comments (#68856)

 stephentoub committed 15 hours ago ✓

Verified



f9a459e



Move `CustomTypeMarshaller` APIs to `System.Runtime.InteropServices.M...` [...](#)

 elinor-fung committed 15 hours ago ✓


Verified



bcded44



Delete stale IgnoreCaseRelation regex tests (#68857) [...](#)

 stephentoub committed 15 hours ago ✓

Verified



3929af4



Fix singlefile on OSX ARM64 (#68845) [...](#)

 VSadov committed 17 hours ago ✓

Verified



d44343b



Log of commits (web version)

Commits on May 4, 2022

Improving the 64-bit number formatting to better match the 32-bit alg... [...](#)

 tannergooding committed 12 hours ago ✗

Verified



874c6a9



Update message for unsupported char marshalling (#68865)

 elinor-fung committed 13 hours ago ✓

Verified



881231a




Add a four-element AsyncLocalValueMap type (#68790) [...](#)

* Add a four-element AsyncLocalValueMap type

We previously special-cased up to three active AsyncLocals in a given async flow, but it seems four is also very common. Special-casing four as well results in four using ~20% less allocation and ~10% less CPU overhead.

* Fix downgrading to FourElementAsyncLocalValueMap, and clean up source

 stephentoub committed 14 hours ago ✓

Verified



16b6369



Stop escaping ' and " in generated regex XML comments (#68856)

 stephentoub committed 15 hours ago ✓

Verified



f9a459e



Move `CustomTypeMarshaller` APIs to `System.Runtime.InteropServices.M...` [...](#)

 elinor-fung committed 15 hours ago ✓

Verified



bcded44



5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters

```
$ git log --oneline -5 --author cbeams --before "Fri Mar 26 2019"
```

```
e5f4b49 Re-adding ConfigurationPostProcessorTests after its brief removal in r814. @Ignore-ing the testCglibClassesAreLoadedJustI  
2db0f12 fixed two build-breaking issues: + reverted ClassMetadataReadingVisitor to revision 794 + eliminated ConfigurationPostPro  
147709f Tweaks to package-info.java files  
22b25e0 Consolidated Util and MutableAnnotationUtils classes into existing AsmUtils  
7f96f57 polishing
```

```
$ git log --oneline -5 --author pwebb --before "Sat Aug 30 2014"
```

```
5ba3db6 Fix failing CompositePropertySourceTests  
84564a0 Rework @PropertySource early parsing logic  
e142fd1 Add tests for ImportSelector meta-data  
887815f Update docbook dependency and generate epub  
ac8326d Polish mockito usage
```

5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- *Team-level convention on*
 - *style (markup syntax, wrap margins, capitalization, punctuation) and*
 - *metadata (how to reference issue tracking IDs, pull request numbers, etc.)*

5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- *Team-level convention on*
 - *style (markup syntax, wrap margins, capitalization, punctuation) and*
 - *metadata (how to reference issue tracking IDs, pull request numbers, etc.)*
- Use the imperative mood in the subject line

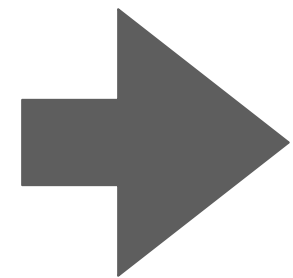
5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
 - *Team-level convention on*
 - *style (markup syntax, wrap margins, capitalization, punctuation) and*
 - *metadata (how to reference issue tracking IDs, pull request numbers, etc.)*
- Use the imperative mood in the subject line
 - *“If applied, this commit will _____”*

5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
- *Team-level convention on*
 - *style (markup syntax, wrap margins, capitalization, punctuation) and*
 - *metadata (how to reference issue tracking IDs, pull request numbers, etc.)*
- Use the imperative mood in the subject line
- *“If applied, this commit will _____”*

```
Refactor subsystem X for readability
Update getting started documentation
Remove deprecated methods
```



```
If applied, this commit will refactor subsystem X for readability
If applied, this commit will update getting started documentation
If applied, this commit will remove deprecated methods
```

5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
 - *Team-level convention on*
 - *style (markup syntax, wrap margins, capitalization, punctuation) and*
 - *metadata (how to reference issue tracking IDs, pull request numbers, etc.)*
- Use the imperative mood in the subject line
 - *“If applied, this commit will _____”*
- Wrap the body at 72 characters
- Use the body to explain **what and why** not *how*

5 Rules of Commit Messages

- Separate subject from body with a blank line
- Limit the subject line to 50 characters
 - *Team-level convention on*
 - *style (markup syntax, wrap margins)*
 - *metadata (how to reference issue trackers)*
- Use the imperative mood in the subject line
 - *“If applied, this commit will _____”*
- Wrap the body at 72 characters
- Use the body to explain what and why not *how*

```
commit eb0b56b19017ab5c16c745e6da39c53126924ed6
Author: Pieter Wuille <pieter.wuille@gmail.com>
Date: Fri Aug 1 22:57:55 2014 +0200
```

```
Simplify serialize.h's exception handling
```

```
Remove the 'state' and 'exceptmask' from serialize.h's stream
implementations, as well as related methods.
```

```
As exceptmask always included 'failbit', and setstate was always
called with bits = failbit, all it did was immediately raise an
exception. Get rid of those variables, and replace the setstate
with direct exception throwing (which also removes some dead
code).
```

```
As a result, good() is never reached after a failure (there are
only 2 calls, one of which is in tests), and can just be replaced
by !eof().
```

```
fail(), clear(n) and exceptions() are just never called. Delete
them.
```