The Guide for Tagging

Conferences often use tags on sessions, either to guide users in session selection or for schedulers to avoid parallel offerings of sessions with overlapping audience interest.

The purpose of this document is to develop standards for tagging as used by technical conferences, particularly C++ conferences, and to document the rationale for those standards. Using this guide, different individuals should be able to read a session title and abstract and each create a set of tags similar to that created by others.

Adoption

This document is the official tagging guide for C++ On Sea, C++Now, Core C++, and CppCon and may have information specific to those conferences. This document is public so that it can be used as a reference for any organization that wants to leverage it. Please contact the maintainers if you'd like to have editing permissions and/or you'd like to have content specific to your organization.

Purpose of Tags

The reason that conferences want to tag sessions is so that sessions can be grouped with other sessions of overlapping audience and/or interest.

Describing a talk is the job of the session abstract, not the tags.

Tags should indicate the subject of the talk, not its presentation format. For example, "workshop" or "overview" are not good tags. Someone interested in a particular workshop is not necessarily interested in all workshops and someone looking for a basic introduction on a subject, may not be interested in an introduction on every subject.

For these reasons, tags should be general in nature and focus on the subject of the talk. For example, a tag such as "lock-free containers" might be a good description of a talk, but it is too specific to be useful as a tag. A better tag is "concurrency," because anyone interested in a session on lock-free containers is likely to be interested in other sessions on concurrency, even if they aren't about lock-free data structures.

Value of Quality Tagging

If a conference schedules its sessions in such a way that no two sessions with any matching tag are scheduled at the same time, and if the individual sessions are well tagged, the conference experience is improved for attendees.

Essentially, each tag is a “mini-track.” Whether an attendee is most interested in “games,” “concurrency,” or “metaprogramming,” that attendee can be assured that they can attend all the sessions on whichever topic is their greatest interest, because no two will be offered in parallel.

Because deciding which sessions to attend is a tough decision for attendees, quality tagging can have an important impact on the attendee experience.

Common Mistakes

The single biggest mistake made by session taggers is *being too specific* with the tags.

Guideline: *Tag the broad subject area of the talk, not the narrow details.*

The second biggest mistake is to tag *information about the session*, rather than the content of the talk.

Guideline: *Tag the content, not the format of the talk.*

Style Guide

Rationale

In order to be effective, tags must match exactly. Software for grouping and/or scheduling may not be aware that “data” and “date” have very different meanings, but “game” and “games” do not. There is no particular way to argue, on first principles, why “game,” “games,” “game development,” or “game programming” is the correct tag, but we do need to agree on the spelling of specific tags or we defeat the whole purpose of tagging sessions.

General rules

In this section of the style guide, we'll discuss general rules, some of which are arbitrary, but which guide us toward agreeing on specific tags. Later in this document we'll simply catalog specific tags that are to be used or avoided.

Case

Tags should be in lowercase except where the tag is either a proper noun or an acronym. Even some proper nouns may be in lowercase if that is how they are generally written, for example "asm.js." Note that languages start with a capital letter and JavaScript has an embedded capital "S." Examples:

as-if rule
Android
ARM
Bluetooth
C
C++17
contracts
DLLs
Java

Acronyms

Jargon and acronyms can be in-grouping and off-putting–hardly something a conference should embrace. Although acronyms can save space and time when used properly, tags are generally not the proper place for them. On the other hand, some acronyms are so well established that attendees might be more likely to recognize the acronym than the spelled-out version.
In general, use acronyms only where users are more likely to recognize the acronym than the spelled out phrase. Examples:

artificial intelligence *not ai or AI*
APIs *see "Plural" below*
ARM *not Advanced RISC Machines*
CUDA
DLLs
entity-component system *not ECS*
general purpose GPUs *not GPGPUs*
high-performance computing *not HPC*
HTML
RISC *not Reduced Instruction Set Computer*
Visual C++ *not VC++*

Plural

In general, technical talks strive to be as detailed as they need to be, but attempt to be general enough to cover as many cases as applicable. So, talks are more likely to talk about items (plural) in general and not a single item.

In most cases, the plural form works better for tags, so to be as consistent as possible, we pluralize nouns, except where called out otherwise. Examples:

abstractions
algorithms
architecture -- note the exception
atomics
best practices
concepts
dependencies
games
GPUs

Verb form

Technical talks sometimes describe how things are, but more commonly describe how to do things. For that reason we prefer the verb form to the noun form in most cases. There are exceptions. Examples:

benchmarking *instead of benchmark or benchmarks*
compiling *instead of compiler or compilers*
debugging
exception handling
interrupts *not interrupting*
mocking *not mock objects*
naming *instead of names*
teams *not teaming*
testing

Assumed words

Presenters at a technical conference are not here to talk about asynchronous dance, they are here to talk about asynchronous programming. Probably half of the tags we use could end in “programming,” but that is just noise. If “programming” (or “C++”) can be assumed, then leave it off. Note that in some cases it does seem necessary for meaning. Examples:

tooling *not “C++ tools” or “C++ tooling”*
naming *not “C++ naming”*
concurrency *not concurrent programming*
event-driven *not event-driven programming*

functional *not functional programming*
generic programming *note exception, “generic” is just too… well, generic*
heterogeneous computing *note exception*
metaprogramming *instead of template metaprogramming or TMP*

Hyphenation

When a noun is used with another word to modify another noun, it is hyphenated. This applies in many tagging situations. Examples:

best practices *note: not hyphenated*
event-driven
lock-free
object-oriented

Less specific

To avoid tags that may be overly detailed, we select tags that are less specific. Examples:

asynchronous *instead of async I/O*
atomics *instead of atomic operations*
Bluetooth *instead of Bluetooth Low Energy*
Clang/LLVM *instead of Clang or LLVM*
documentation *instead of documentation tooling*
testing *instead of * testing*
Unicode *instead of UTF-8*

Useless and Semi-Useless Tags

Some tags are either too common, too vague, or too overused to be meaningful for grouping or scheduling sessions. Useless tags should not be used as they can break scheduling software. Some tags describe characteristics of programming that are so ubiquitous that in some sense, they could be applied to (almost) every talk, but do make some sense when applied to talks that focus on that characteristic.
Examples of tags that either shouldn’t be used or should be used advisedly are:

C++: At a C++ conference, every session should be about C++ at some level. The exception to this might be a session on the history of C++ or comparing other languages to C++. Even then, suspect.

correctness: No talk is about how to write incorrect code (except as examples of fails), so all talks are about correctness in some sense. But this tag does make sense for a talk on proving, measuring, guaranteeing, or otherwise focusing on correctness.

coding: We are engineers. Coding is what we do. Almost every session is about coding in some sense.

designing, engineering, programming: See Coding.

performance: Performance is one of the most important features of C++ as a language and it motivates much of what we do. Not all, but a clear majority of presentations will have performance as, at least, a minor theme. If every session that touched on performance in some way had this tag, it would be useless. But like "correctness," performance can be meaningful if its use is restricted to presentations that focus on performance itself, rather than just considering techniques from a performance perspective. For example, presentations on benchmarking or other performance measurements, performance testing, or talks in which performance is the overriding theme and not just one criterion on which techniques or code is judged.

BoF, case study, introduction, lecture, lessons learned, overview, panel, tutorial, workshop: Tags should reflect the content of the presentations, not their format. The conference should not schedule two sessions on concurrency at the same time, but there is no reason not to schedule two panels (on different topics) at the same time.

Note that conference management may wish to create a "tutorial track," in which beginner sessions are offered in series rather than parallel, but perhaps tagging is not the best approach for this. Whether or not a particular session should be in such a track requires information about the set of other offerings available and can't be determined just by understanding the particular offering. Tagging is something that should be determinable based only on the information available about a particular session and not dependent on knowing about all other offerings.

Boost Libraries

See the Boost library naming rules for how Boost libraries names should be presented. In the tagging context, Boost libraries should follow this pattern:

Boost.Asio
Boost.Beast
Boost.Python
Boost.Algorithm
Boost.Serialization

Contact

The email address for maintainers of this document is:
tagging@cppcon.org
If you'd like to be on this email list, please send mail to that address.

Specific Tags

These are some real-world cases that have come up, and the tags that were recommended. Note that many decisions are subjective and so, may be reconsidered. Please share your thoughts, either as a comment to this document or as an email to the maintainers (see Contact).

The text before the ⇒ is the lowercase version of the proposed tag. The text after the ⇒ is the recommended tag. When you see the same value on each side, that means that the value on the right is specifying the correct upper/lowercase of the tag.

- "as-if" rule ⇒ as-if rule
    - "as-if" ⇒ as-if rule
    - <random> ⇒ random
    - a ⇒ <do not use this tag>
    - aa ⇒ <do not use this tag>
    - abstract machine ⇒ abstract machine
    - abstraction ⇒ <do not use this tag>
    - abstractions ⇒ <do not use this tag>
    - accumulate ⇒ accumulate
    - acquisition ⇒ acquisition
    - actor model ⇒ actor model
    - address sanitizer ⇒ AddressSanitizer
    - address-sanitizer ⇒ AddressSanitizer
    - addresssanitizer ⇒ AddressSanitizer
    - ai ⇒ artificial intelligence
    - algorithm ⇒ algorithms
    - algorithms ⇒ algorithms
    - allocation ⇒ allocators
    - allocator ⇒ allocators
    - allocators ⇒ allocators
    - android ⇒ Android
    - anti pattern ⇒ anti-pattern
    - antipattern ⇒ anti-pattern
    - api design ⇒ design
    - api ⇒ APIs
    - apis ⇒ APIs
    - apple ⇒ Apple
    - application programmer ⇒ application programming
    - application programming ⇒ application programming
    - archecture ⇒ architecture
    - archeture ⇒ architecture
    - arm ⇒ ARM
    - artificial-intelligence ⇒ artificial intelligence
    - as-if rule ⇒ as-if rule
    - as-if ⇒ as-if rule
    - asan ⇒ AddressSanitizer
    - asio ⇒ Boost.Asio
    - asm.js ⇒ asm.js

assert ⇒ assertions
assertion ⇒ assertions
assertions ⇒ assertions
asserts ⇒ assertions
ast ⇒ AST
async io ⇒ asynchronous
async programming ⇒ asynchronous
async ⇒ asynchronous
asynchronous events ⇒ asynchronous
asynchronous programming ⇒ asynchronous
asynchronous ⇒ asynchronous
asynchrony ⇒ asynchronous
atomic operations ⇒ atomics
atomic ⇒ atomics
atomics ⇒ atomics
automated testing ⇒ testing
automatic testing ⇒ testing
aws ⇒ AWS
backward compatibility ⇒ backward compatibility
backwards compatibility ⇒ backward compatibility
bde ⇒ Bloomberg Development Environment
behavior driven development ⇒ behavior-driven
behavior driven ⇒ behavior-driven
behavior-driven development ⇒ behavior-driven
behavior-driven ⇒ behavior-driven
benchmark ⇒ benchmarking
benchmark ⇒ benchmarking
benchmarking libraries ⇒ benchmarking
benchmarking ⇒ benchmarking
benchmarks ⇒ benchmarking
best practice ⇒ best practices
best practices ⇒ best practices
best-practice ⇒ best practices
best-practices ⇒ best practices
blaze ⇒ Blaze
ble ⇒ Bluetooth
blom ⇒ Boost Library Official Maintainer
bluetooth LE ⇒ Bluetooth
bluetooth low energy ⇒ Bluetooth
bluetooth-lowenergy ⇒ Bluetooth
bluetooth ⇒ Bluetooth
bof ⇒ <do not use this tag>
boost ⇒ Boost

boostache ⇒ Boostache
build ⇒ build systems
building ⇒ build systems
builds ⇒ build systems
c compatibility ⇒ C compatibility
c ⇒ C
c# ⇒ C#
c++ amp ⇒ C++ AMP
c++ core guidelines ⇒ Core Guidelines
C++ libraries ⇒ libraries
C++ library ⇒ libraries
C++ name ⇒ naming
C++ names ⇒ naming
C++ naming ⇒ naming
C++ package manager ⇒ package manager
C++ standard ⇒ ISO C++
c++ templates ⇒ templates
c++ tools ⇒ tooling
c++ ⇒ <do not use this tag>
c++11 ⇒ C++11
c++14 ⇒ C++14
c++17 ⇒ C++17
c++20 ⇒ C++20
c++amp ⇒ C++ AMP
c++next ⇒ future standards
c++now ⇒ C++Now
case study ⇒ <do not use this tag>
catch ⇒ Catch
categories ⇒ category theory
celero ⇒ Celero
ci ⇒ continuous integration
civil time ⇒ time
clang ⇒ Clang/LLVM
clang-tidy ⇒ Clang/LLVM
clojure ⇒ Clojure
cloud ⇒ cloud
cmake ⇒ CMake
cms ⇒ CMS
co-array ⇒ Coarray
coach ⇒ coaching
coaches ⇒ coaching
coaching ⇒ coaching
coarray ⇒ Coarray

code ⇒ <do not use this tag>
code browsing ⇒ code comprehension
code comprehension ⇒ code comprehension
code generation ⇒ code generation
code generator ⇒ code generation
code health ⇒ code quality
code quality ⇒ code quality
code sharing ⇒ code sharing
code visualization ⇒ code comprehension
code-generation ⇒ code generation
coding ⇒ <do not use this tag>
com ⇒ COM
committee ⇒ ISO Committee
compilation ⇒ compiling
compile time ⇒ compile-time
compile-time ⇒ compile-time
compile ⇒ compiling
compiler ⇒ compiling
compilers ⇒ compiling
component ⇒ components
concept ⇒ concepts
concepts ⇒ concepts
conceptual ⇒ concepts
concurrency ⇒ concurrency
concurrent programming ⇒ concurrency
condition variable ⇒ concurrency
condition_variable ⇒ concurrency
const ⇒ const
constraint based programming ⇒ constraint-based
constraint based ⇒ constraint-based
constraint-based programming ⇒ constraint-based
constraint-based ⇒ constraint-based
containers ⇒ containers
continuous integration ⇒ continuous integration
contract ⇒ contracts
contracts ⇒ contracts
core guidelines ⇒ Core Guidelines
coroutine ⇒ coroutines
cpp ⇒ C++
cpp11 ⇒ C++11
cpp14 ⇒ C++14
cpp17 ⇒ C++17
cpp20 ⇒ C++20

cppcast ⇒ CppCast
cppcon ⇒ CppCon
cppnow ⇒ C++Now
cpu cache ⇒ CPUs
cpu caches ⇒ CPUs
cpu ⇒ CPUs
cpus ⇒ CPUs
cross platform ⇒ cross-platform
cross-platform ⇒ cross-platform
cuda ⇒ CUDA
custom allocator ⇒ allocators
custom allocators ⇒ allocators
data flow ⇒ dataflow
data oriented ⇒ data-oriented
data-flow ⇒ dataflow
data-oriented ⇒ data-oriented
dataflow ⇒ dataflow
dataoriented ⇒ data-oriented
debugger ⇒ debugging
debuggers ⇒ debugging
debugging ⇒ debugging
declarative programming ⇒ declarative
declarative ⇒ declarative
dependencies ⇒ dependencies
dependency injection ⇒ dependency injection
dependency management ⇒ dependencies
dependency ⇒ dependencies
deprecate ⇒ deprecation
deprecating ⇒ deprecation
deprecation ⇒ deprecation
design by contract ⇒ design-by-contract
design patterns ⇒ design patterns
design-by-contract ⇒ design-by-contract
design ⇒ <do not use this tag>
designs ⇒ <do not use this tag>
designing ⇒ <do not use this tag>
developer tools ⇒ tooling
development ⇒ <do not use this tag>
device drivers ⇒ device drivers
devops ⇒ DevOps
di ⇒ dependency injection
diffusion ⇒ diffusion
dijkstra ⇒ Dijkstra

distributed computing ⇒ distributed
distributed containers ⇒ distributed
distributed systems ⇒ distributed
distributed ⇒ distributed
distribution ⇒ distributed
dlls ⇒ Dlls
docs ⇒ documentation
documentation tool ⇒ documentation
documentation ⇒ documentation
domain specific language ⇒ domain specific language
dsl ⇒ domain specific language
dwarf ⇒ DWARF
ecs ⇒ entity-component system
edsl ⇒ domain specific language
education ⇒ education
engineering ⇒ <do not use this tag>
embedded domain specific language ⇒ domain specific language
embedded systems ⇒ embedded
embedded ⇒ embedded
emotional intelligence ⇒ emotional intelligence
emscripten ⇒ Emscripten
error ⇒ error handling
event based programming ⇒ event-driven
event driven programming ⇒ event-driven
event handling ⇒ event-driven
event processing ⇒ event-driven
event-based programming ⇒ event-driven
event-based ⇒ event-driven
event-driven programming ⇒ event-driven
event-handling ⇒ event-driven
event-processing ⇒ event-driven
exception handling ⇒ exception handling
exception safe code ⇒ exception handling
exception safe ⇒ exception handling
exception safety ⇒ exception handling
exception-handling ⇒ exception handling
exception-safe ⇒ exception handling
exception-safety ⇒ exception handling
exception ⇒ exception handling
exceptions ⇒ exception handling
fiber ⇒ fibers
fibers ⇒ fibers
fibre ⇒ fibers

fibres ⇒ fibers
file system ⇒ filesystems
file systems ⇒ filesystems
file-system ⇒ filesystems
file-systems ⇒ filesystems
filesystem ⇒ filesystems
filesystems ⇒ filesystems
finance ⇒ financial engineering
financial data ⇒ financial engineering
financial ⇒ financial engineering
fixed point ⇒ fixed-point
fixed-point ⇒ fixed-point
fixedpoint ⇒ fixed-point
format ⇒ formatting
formats ⇒ formatting
formatting ⇒ formatting
foss ⇒ open source
fpga ⇒ field-programmable gate array
functional paradigm ⇒ functional
functional programming ⇒ functional
functional-programming ⇒ functional
functional ⇒ functional
future ⇒ futures
futures ⇒ futures
game development ⇒ games
game programmers ⇒ games
game ⇒ games
games ⇒ games
gcc ⇒ GCC
gdb ⇒ GDB
general purpose gpus ⇒ general purpose GPUs
general scientific interface ⇒ General Scientific Interfaces
general scientific interfaces ⇒ General Scientific Interfaces
general-purpose gpus ⇒ general purpose GPUs
general ⇒ <do not use this tag>
generic lambda ⇒ lambdas
generic programming ⇒ generic programming
generic-programming ⇒ generic programming
generics ⇒ generic programming
gpgpu ⇒ general purpose GPUs
gpgpus ⇒ general purpose GPUs
gpu ⇒ GPUs
gpus ⇒ GPUs

graph ⇒ graphs
graphic ⇒ graphics
graphics ⇒ graphics
graphs ⇒ graphs
gsi ⇒ General Scientific Interfaces
gui library ⇒ GUI
gui programming ⇒ GUI
gui ⇒ GUI
hash table ⇒ hashing
hash-table ⇒ hashing
hash ⇒ hashing
hashes ⇒ hashing
hashing ⇒ hashing
hashtable ⇒ hashing
heterogeneous computing ⇒ heterogeneous computing
heterogeneous programming ⇒ heterogeneous computing
heterogeneous-computing ⇒ heterogeneous computing
heterogeneous-programming ⇒ heterogeneous computing
heterogeneous ⇒ heterogeneous computing
hft ⇒ financial engineering
high performance computing ⇒ high performance computing
high-performance computing ⇒ high performance computing
higher level API ⇒ APIs
higher level APIs ⇒ APIs
hippomocks ⇒ Hippomocks
hoare ⇒ Hoare
hpc ⇒ high performance computing
hpx ⇒ HPX
html ⇒ HTML
http ⇒ HTTP
human factors ⇒ human factors
hypervisor ⇒ hypervisor
ide ⇒ IDE
image analysis ⇒ image analysis
image recognition ⇒ image analysis
image-analysis ⇒ image analysis
image-recognition ⇒ image analysis
imagerecognition ⇒ image analysis
immutability ⇒ immutable
immutable ⇒ immutable
in depth ⇒ <do not use this tag>
in-depth ⇒ <do not use this tag>
include ⇒ includes

includeos ⇒ IncludeOS
includes ⇒ includes
infrastructure ⇒ <do not use this tag>
inherit ⇒ inheritance
inheritance ⇒ inheritance
inherits ⇒ inheritance
interactive software ⇒ interactive
interactive ⇒ interactive
interactivity ⇒ interactive
internet of things ⇒ Internet of Things
interrupt handling ⇒ interrupts
interrupt ⇒ interrupts
interrupts ⇒ interrupts
introduction ⇒ <do not use this tag>
introspection ⇒ reflection
io ⇒ I/O
ios ⇒ iOS
iostreams ⇒ iostreams
iot ⇒ Internet of Things
iso c++ ⇒ ISO C++
isr ⇒ interrupt service routine
itanium abi ⇒ Itanium ABI
itanium-abi ⇒ Itanium ABI
itaniumabi ⇒ Itanium ABI
java native interface ⇒ Java Native Interface
javascript ⇒ JavaScript
jenkins ⇒ Jenkins
jni ⇒ Java Native Interface
json ⇒ JSON
kernel mode ⇒ kernel
kernel-mode ⇒ kernel
kernel ⇒ kernel
kernelmode ⇒ kernel
lambda ⇒ lambdas
lambdas ⇒ lambdas
language design ⇒ languages
language designs ⇒ languages
language ⇒ languages
languages design ⇒ languages
languages designs ⇒ languages
languages evolution ⇒ languages
languages mechanics ⇒ languages
languages ⇒ languages

large code base ⇒ large-scale
large scale ⇒ large-scale
large scale architecture ⇒ large-scale
large scale development ⇒ large-scale
large scale software development ⇒ large-scale
large-scale ⇒ large-scale
large-scale architecture ⇒ large-scale
large-scale development ⇒ large-scale
large-scale software development ⇒ large-scale
latency ⇒ latency
launder ⇒ launder
launder() ⇒ launder
lecture ⇒ <do not use this tag>
legacy ⇒ legacy
legacy code base ⇒ legacy
legacy software ⇒ legacy
learning ⇒ education
lessons learned ⇒ <do not use this tag>
libclang ⇒ Clang/LLVM
libraries ⇒ libraries
library ⇒ libraries
library design ⇒ libraries
library development ⇒ libraries
library building ⇒ libraries
libtooling ⇒ Clang/LLVM
license ⇒ licenses
licenses ⇒ licenses
licensing ⇒ licenses
link ⇒ linking
link time ⇒ linking
linkage ⇒ linking
linker ⇒ linking
linking ⇒ linking
linktime ⇒ linking
linux ⇒ Linux
linux ⇒ Linux
literal ⇒ literals
literals ⇒ literals
literal type ⇒ literals
literal types ⇒ literals
lldb ⇒ Clang/LLVM
llvm ⇒ Clang/LLVM
llvm/clang ⇒ Clang/LLVM

load balancing ⇒ load balancing
locality ⇒ locality
lock free programming ⇒ lock-free
lock free ⇒ lock-free
lock-free programming ⇒ lock-free
lock-free ⇒ lock-free
locking ⇒ lock-free
lockless programming ⇒ lock-free
lockless readers ⇒ lock-free
lockless ⇒ lock-free
logging ⇒ logging
logic ⇒ logic
loops ⇒ loops
low latency ⇒ latency
low latency programming ⇒ latency
low level ⇒ low level
low level programming ⇒ low level
low-latency ⇒ latency
low-latency programming ⇒ latency
low-level ⇒ low level
low-level programming ⇒ low level
machine learning ⇒ machine learning
machinelearning ⇒ machine learning
maintainability ⇒ maintainability
maintanence ⇒ maintainability
maintenance ⇒ maintainability
manyrepos ⇒ manyrepos
measurement ⇒ measuring
measurements ⇒ measuring
measuring ⇒ measuring
memory ⇒ memory
memory management ⇒ memory
memory model ⇒ memory
mersenne twister ⇒ random
messaging ⇒ messaging
message ⇒ messaging
message broker ⇒ messaging
message passing ⇒ messaging
messages ⇒ messaging
meta ⇒ meta
meta classes ⇒ metaclasses
metaclasses ⇒ metaclasses
meta-programming ⇒ metaprogramming

metaprogramming ⇒ metaprogramming
mock objects ⇒ mocking
Mock Objects ⇒ mocking
mocking ⇒ mocking
modern ⇒ modern C++
modern c++ ⇒ modern C++
modern code ⇒ modern C++
modernization ⇒ modern C++
modernize ⇒ modern C++
modular ⇒ modularity
modular design ⇒ modularity
modular development ⇒ modularity
modularity ⇒ modularity
modularization guidelines ⇒ modularity
modules ⇒ modules
monorepo ⇒ monorepo
move semantics ⇒ move semantics
mpi ⇒ messaging
mpl ⇒ metaprogramming
mqtt ⇒ messaging
msm ⇒ Boost.MetaStateMachine
multi-core ⇒ concurrency
multi-threading ⇒ concurrency
multi-device development ⇒ multi-device development
multi-dimensional ⇒ multidimensional
multi-language ⇒ multi-language
multi-method ⇒ multi-method
multi-precision ⇒ multiprecision
multi-precision arithmetic ⇒ multiprecision
multi-precision math ⇒ multiprecision
multicore ⇒ concurrency
multithreading ⇒ concurrency
multidimensional ⇒ multidimensional
multidimensional arrays ⇒ multidimensional
multidimensional data ⇒ multidimensional
multiprecision ⇒ multiprecision
multiprecision arithmetic ⇒ multiprecision
multiprecision math ⇒ multiprecision
mutex ⇒ concurrency
name ⇒ naming
names ⇒ naming
naming ⇒ naming
nasal demons ⇒ undefined behavior

network ⇒ networking
networking ⇒ networking
networking ts ⇒ Networking TS
networkingts ⇒ Networking TS
networks ⇒ networking
neural network ⇒ neural networks
neural networks ⇒ neural networks
next generation ⇒ <do not use this tag>
next generations ⇒ <do not use this tag>
new technologies ⇒ <do not use this tag>
new technology ⇒ <do not use this tag>
node.js ⇒ node.js
non-blocking ⇒ concurrency
nonblocking ⇒ concurrency
nosql ⇒ NoSQL
nuget ⇒ NuGet
numerics ⇒ numerics
nvidia ⇒ Nvidia
object ⇒ objects
object model ⇒ objects
object size ⇒ objects
objects ⇒ objects
object oriented design ⇒ object-oriented
object oriented programming ⇒ object-oriented
object oriented ⇒ object-oriented
object-oriented design ⇒ object-oriented
object-oriented programming ⇒ object-oriented
object-oriented ⇒ object-oriented
observer ⇒ observer
OOP ⇒ object-oriented
open source ⇒ open source
opencl ⇒ OpenCL
openmp ⇒ OpenMP
operating system ⇒ operating systems
operating systems ⇒ operating systems
operator ⇒ operators
operators ⇒ operators
optimization ⇒ optimizations
optimizations ⇒ optimizations
optional ⇒ optional
orm ⇒ object-relational mapping
overview ⇒ <do not use this tag>
overload ⇒ overloading

overloading ⇒ overloading
overloads ⇒ overloading
ownership ⇒ ownership
package ⇒ package management
package dependency management ⇒ package management
package management ⇒ package management
package dependency manager ⇒ package management
package manager ⇒ package management
package repository ⇒ package management
packaging ⇒ package management
panel ⇒ <do not use this tag>
parsing ⇒ parsing
parallel algorithms ⇒ concurrency
parallel programming ⇒ concurrency
parallel stl ⇒ Parallel STL
parallel ⇒ concurrency
parallelism ⇒ concurrency
parallelism ts ⇒ Parallelism TS
parallelismts ⇒ Parallelism TS
parallelization ⇒ concurrency
parallelstl ⇒ Parallel STL
past ⇒ <do not use this tag>
pattern matching ⇒ pattern matching
pattern ⇒ <do not use this tag>
patterns ⇒ <do not use this tag>
perfect forwarding ⇒ perfect forwarding
performance analysis ⇒ performance
performance monitoring ⇒ performance
performance ⇒ performance
persistence ⇒ persistent data types
persistent data ⇒ persistent data types
persistent data types ⇒ persistent data types
persistent data structures ⇒ persistent data types
plenary ⇒ <do not use this tag>
policy based class design ⇒ policy-based
policy based design ⇒ policy-based
policy-based class design ⇒ policy-based
policy-based design ⇒ policy-based
policy-based ⇒ policy-based
postmodern C++ ⇒ postmodern
postmodern ⇒ postmodern
postmodernism ⇒ postmodern
preprocessor ⇒ preprocessor

present ⇒ <do not use this tag>
program ⇒ <do not use this tag>
program design ⇒ <do not use this tag>
programming ⇒ <do not use this tag>
programming by contract ⇒ contracts
programming-by-contract ⇒ contracts
project dependency manager ⇒ package management
property-based testing ⇒ testing
protocol ⇒ protocols
protocols ⇒ protocols
puzzle ⇒ <do not use this tag>
puzzles ⇒ <do not use this tag>
python ⇒ Python
question ⇒ <do not use this tag>
questions ⇒ <do not use this tag>
qt ⇒ Qt
raii ⇒ RAII
radix sort ⇒ radix sort
random ⇒ random
random engine ⇒ random
random engines ⇒ random
random number ⇒ random
random numbers ⇒ random
random number toolkit ⇒ random
ranges ⇒ ranges
rcu ⇒ concurrency
readability ⇒ readability
read-copy-update ⇒ concurrency
reader-writer lock ⇒ concurrency
real time ⇒ <do not use this tag>
real time graphics ⇒ real-time graphics
real-time graphics ⇒ real-time graphics
reference ⇒ references
references ⇒ references
reference semantics ⇒ reference semantics
reference-semantics ⇒ reference semantics
reflection ⇒ reflection
regular types ⇒ generic programming
remote procedure calls ⇒ remote procedure calls
requirements ⇒ requirements
requirements in code ⇒ requirements
resource ⇒ <do not use this tag>
resources ⇒ <do not use this tag>

resumable ⇒ resumables
resumables ⇒ resumables
resumable expressions ⇒ resumables
resumable functions ⇒ resumables
rest ⇒ REST
RISC ⇒ RISC
RISC-V ⇒ RISC
robot ⇒ robotics
robotics ⇒ robotics
robots ⇒ robotics
robustness ⇒ robustness
rocket ⇒ rocket science
rocket science ⇒ rocket science
rockets ⇒ rocket science
rocketry ⇒ rocket science
rpc ⇒ remote procedure calls
rtc ⇒ time
rule of zero ⇒ rule of zero
runtime reflection ⇒ reflection
rust ⇒ Rust
safe types ⇒ safe types
safety ⇒ type safety
safety systems ⇒ safety systems
sanitizers ⇒ sanitizers
sanity ⇒ <do not use this tag>
science ⇒ scientific computing
scientific computing ⇒ scientific computing
scons ⇒ SCons
scope ⇒ <do not use this tag>
scripting ⇒ scripting
scripting languages ⇒ scripting
sfinae ⇒ SFINAE
sg14 ⇒ SG14
sg 14 ⇒ SG14
sg-14 ⇒ SG14
si unit ⇒ SI units
si units ⇒ SI units
simd ⇒ SIMD
simplicity ⇒ simplicity
simplify ⇒ simplicity
software ⇒ <do not use this tag>
software engineering ⇒ software engineering
sound ⇒ audio

spmd ⇒ concurrency
sql ⇒ SQL
stl ⇒ STL
stacktrace ⇒ debugging
stacktraces ⇒ debugging
static ⇒ <do not use this tag>
static analysis ⇒ static analysis
static analysis tool ⇒ static analysis
static analysis tools ⇒ static analysis
study ⇒ <do not use this tag>
swift ⇒ Swift
sycl ⇒ OpenCL
synchronization ⇒ concurrency
tcp ⇒ TCP
teaching ⇒ education
team ⇒ teams
teams ⇒ teams
technique ⇒ <do not use this tag>
techniques ⇒ <do not use this tag>
template ⇒ templates
template specialization ⇒ templates
templates ⇒ templates
template metaprogramming ⇒ metaprogramming
test ⇒ testing
testing ⇒ testing
tests ⇒ testing
time zone ⇒ time
time ⇒ time
tmp ⇒ metaprogramming
tooling ⇒ tooling
tools ⇒ tooling
trading ⇒ financial engineering
training ⇒ education
transaction ⇒ transactions
transactions ⇒ transactions
transactional memory ⇒ transactional memory
tutorial ⇒ <do not use this tag>
tvos ⇒ tvOS
type ⇒ <do not use this tag>
type deduction ⇒ type deduction
type erasure ⇒ type erasure
type safety ⇒ type safety
types ⇒ <do not use this tag>

ub ⇒ undefined behavior
ucs 2 ⇒ Unicode
ucs 4 ⇒ Unicode
ucs-2 ⇒ Unicode
ucs-4 ⇒ Unicode
ucs2 ⇒ Unicode
ucs4 ⇒ Unicode
uftrace ⇒ uftrace
ui ⇒ user interface
undefined behavior ⇒ undefined behavior
unicode ⇒ Unicode
unified call syntax ⇒ unified call syntax
unikernel ⇒ unikernels
unikernal ⇒ unikernels
unikernals ⇒ unikernels
unit test ⇒ testing
unit testing ⇒ testing
unit-testing ⇒ testing
units ⇒ units
unittest ⇒ testing
unreal engine ⇒ Unreal Engine
user interface ⇒ user interface
utf 16 ⇒ Unicode
utf 32 ⇒ Unicode
utf 8 ⇒ Unicode
utf-16 ⇒ Unicode
utf-32 ⇒ Unicode
utf-8 ⇒ Unicode
utf16 ⇒ Unicode
utf32 ⇒ Unicode
utf8 ⇒ Unicode
utility ⇒ <do not use this tag>
valgrind ⇒ Valgrind
VC++ ⇒ Visual Studio
vcpkg ⇒ Vcpkg
vector ⇒ vector
vectors ⇒ vector
versioning ⇒ versioning
visual c++ ⇒ Visual C++
visual studio ⇒ Visual Studio
wave ⇒ preprocessor
web services ⇒ web services
webgl ⇒ WebGL

websocket ⇒ WebSockets
websockets ⇒ WebSockets
windbg ⇒ WinDbg
windows ⇒ Windows
windows runtime ⇒ Windows
workshop ⇒ <do not use this tag>
zero cost abstractions ⇒ zero-cost abstractions
zero-cost abstractions ⇒ zero-cost abstractions
zero overhead libraries ⇒ zero-overhead libraries
zero-overhead libraries ⇒ zero-overhead libraries
zmq ⇒ zmq