

Document Number: P0446R0  
Date: 2016-10-16  
Authors: Michael Wong  
Project: Programming Language C++, SG5 Transactional Memory  
Reply to: Michael Wong <michael@codeplay.com>

## **SG5: Transactional Memory (TM) Meeting Minutes 2016/07/18-2016/10/10**

### **Contents**

Minutes for 2016/07/18 SG5 Conference Call .....	2
Minutes for 2016/08/01 SG5 Conference Call .....	4
Minutes for 2016/08/15 SG5 Conference Call .....	6
Minutes for 2016/08/29 SG5 Conference Call .....	9
Minutes for 2016/09/12 SG5 Conference Call .....	12
Minutes for 2016/09/26 SG5 Conference Call .....	14
Minutes for 2016/10/10 SG5 Conference Call .....	17

## Minutes for 2016/07/18 SG5 Conference Call

Meeting minutes by victor

Participants: Hans, Maged, Mike Spear, Brett, Michael Wong, Victor (scribe)

Secretary rota: Torvald, Tatiana,, Michael Wong, Michael Scott, Jens Maurer, Mike Spear, Maged,Hans, Victor

### DISCUSSION OF FINLAND MEETING

Parallel Algorithms TS is in C++17  
- but it no longer throwing exceptions out of parallel algorithms

Next C++ standards meetings are in  
Issaquah in November 2016  
Kona in Feb/Mar 2017  
Toronto in July 2017

Issaquah and Kona are likely primarily for ballot resolutions for C++17.  
They are likely to still be focused on C++17, of course,  
but there may be more time for other topics if there aren't a lot of comments.  
SG1 is likely to be discussing future-looking things, for example.

What about fixes to the TM TS (e.g., fixing defects we've noted)?  
Hans: we can put new things into the current working draft.

### DISCUSSION OF FUTURE DIRECTIONS

Mike Spear: Do we want TM to be in C++20 (seeing as it's supposed to be "minor")?

Hans/Michael Wong: C++20 will likely have significant additions (e.g., concepts),  
so it won't be minor.

Michael Wong: We need experience to justify inclusion.  
We know that there are some parties with concerns.  
We should get the ideas we've already put forth for review by Evolution.

Victor: We should take fixing defects off backlog.  
AI (Michael Wong): Work on fixing defects.

Mike Spear: Should we should be reviewing concerns raised about TM spec?

Probably.

Action Item (Victor): Summarize concerns expressed by some in email.

Michael Wong: With gcc 6 out, should we take a closer look at how it's done, do some tests.

Victor: Also, on systems with HW TM support.

Victor: One topic is to look at the ideas proposed for in\_transaction, and move towards a conclusion and a proposal (if any).  
(Add to agenda for next meeting.)

## ACTION ITEMS

Michael Wong: Write up fixes for defects <https://issues.isocpp.org/describecomponents.cgi>

- Replace terminate with abort
- Add transaction cancellation to list of allowed copy elision
- Check for changes needed due to new parallel algorithms being added to C++17

Victor: Summarize concerns about TM proposal expressed in email

NEXT MEETING: 1 Aug 2016

## BACKLOG ISSUES

Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

Continue Retry discussion

[https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib\\_PFfc](https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib_PFfc)  
[https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z\\_A](https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z_A)

## Minutes for 2016/08/01 SG5 Conference Call

Meeting minutes by Michael

### 1.1 Roll call of participants

Maged, Michael Wong, Scott, Spear, Victor , Hans

### 1.2 Adopt agenda

Y

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Y

### 1.4 Review action items from previous meeting (5 min)

1.4.1. Michael Wong: Write up fixes for defects

<https://issues.isocpp.org/describecomponents.cgi>

- Replace terminate with abort

- Add transaction cancellation to list of allowed copy elision

- Check for changes needed due to new parallel algorithms being added to C++17

Done.

1.4.2 Victor: Summarize concerns about TM proposal expressed in email

Coming.

## 2. Main issues (50 min)

### 2.1 Discuss defects

Issue1:

All agree

Victor: should be TS1 defect fix

Michael: believe no one depends on this to terminate

Hans: NOte should be changed to normative so it looks like this:

Michael: we should put it into Ts2

unless C++22 comes ahead of Ts2

AI 1: update issue 1 Michael.

Status: move to review

Issue 2: CWG should discuss

Jens suggestion seems correct

during copy construction, compiler should have permission to elide the copy

Michael S: is there any blanket permission

Hans: no 12.8 has explicit list, we need to add to the list

Michael W: throwing something out of atomic-cancel, and you copy the exception, you are allowed to elide if it is already in the right place

we now require that copy even if the rest of tx is rolled back

something similar to second bullet, but we dont have an operand, its an exception object

AI 2. Hans will send it.

Issue 3:

seems to be a placeholder for Jens to do work on adding exception specification to the type system

This was added to C++17 in Kona

AI 3: Jens to follow up to see if anything needs to be done.

Issue 4:

which library is tx-safe

will we need to make parallelism TS tx-safe

not allowed

Spear: exec policy should not be tx-safe

Hans: logically they are tx-safe, they dont do anything

what happens if you try to run a || algo inside a tx

currently cant create child threads inside a tx

Spear: straw-man this may be a use-case for in\_transaction, use that to select execution policy, they will not be parallel when they run inside the tx,

Hans: these || algo can always run sequentially, they dont require concurrency

but we may want to support vectorize

the intent behind exe policy is constraints, dont want to remove that restrictions.

better off technically allowing it and implementation is allowed to run it sequentially

just say it is tx-safe, impl may do it less aggressively

## 2.2 Discuss email concerns

## 3. Any other business

Spear: note the role of unique ptr inside tx

AI 4: Spear to write something on this

## Minutes for 2016/08/15 SG5 Conference Call

Minutes by Michael Scott

SG5 conference call

Victor Luchangco, Maged Michael, Michael Scott, Mike Spear, Hans Boehm, Jens Maurer

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Torvald, Tatiana, Jens Maurer, Mike Spear, Maged, Hans, Victor, Michael Wong, Michael Scott

1.3 Approved minutes from previous meeting; approved publishing previously approved minutes to ISOCPP.org

1.4 Reviewed action items from previous meeting (5 min)

1.4.1: Victor has clarified that SG5 has no intent to include TM in C++17, and included list of pending issues that justify this conservatism.

1.4.2: update issue 1: Michael Wong (terminate w/ abort). Done

1.4.3: Hans will send draft wording for Issue 2 (copy elision on transaction cancellation). Should happen soon.

1.4.4: Jens to follow up to see if anything needs to be done for Issue 3 (type checking for transaction\_safe functions). Still pending.

1.4.5: Spear to write something on unique\_ptr inside tx. Posted. But doesn't really work: unique\_ptr controls storage management, not access. Perhaps there could be a new kind of smart pointer for this purpose? (More on this below.)

2. Main issues (50 min)

2.1 Discuss defects

Issue 1: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/SMVEiVLbdig>

done (noted above)

Issue 2: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Th7IFxFuIYo>  
(copy elision)  
on Hans's to-do list (noted above)

Issue 3: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/CXBycK3kgo0>  
(type checking for transaction\_safe functions)  
on Jens's to-do list (noted above)

Both Issue 2 and Issue 3 will require careful vetting wrt C++17 updates to wording of the standard.

Issue 4: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Ood8sP1jbCQ>  
(relationship to new parallel algs. in C++17)  
needs to be considered eventually, but not urgent

## 2.2 Discuss email concerns

<https://groups.google.com/a/isocpp.org/forum/#!topic/tm/skPgKhfZDj0>

Keep both atomic and synchronized blocks? Probably yes, until we gain experience with the TS.

Michael Scott: no longer a super-strong advocate, personally, for /atomic/ blocks; could live with just /synchronized/. Victor: still an advocate.

Need to remain aware that TM is one of the few recent language changes that significantly impacts the run-time system. Could be a big issue for implementers, esp. for specialized environments.

## 3. Any other business

More discussion of possibility of a new kind of smart pointer.

Hans: not clear how this would work. What about pointers to things that have pointers to things that ... may be shared?

Jens: might interact with -- or at least be analogous to -- proposals for some version of /restrict/ (which is in C but not C++ at present).

NB: a "private" smart pointer is not a solution to the desire (expressed by Chandler) to avoid unnecessary synchronization when transaction-laden code is called in a provably sequential context. Rather it is (or might be?) a mechanism for privatization.

(Michael Scott: for an overview of privatization, see Marathe et al.,

ICPP 2008.)

Jens: We need examples of all sorts -- code that actually uses TM.

Mike Spear: is it ok for SG5 people to write this code, or do we need buy-in to the TS from others? Probably both :-)

Mike Spear: Torvald may know of gcc bug reports related to TM that identify TM users.

#### 4. Review

##### 4.1 Useful links:

N4513 is the official working draft

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf>

N4514 is the published PDTS:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf>

N4515 is the Editor's report:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html>

Github is where the latest repository is (updated for latest PDTS published draft from post-Leneaxa):

<https://github.com/cplusplus/transactional-memory-ts>

Bugzilla for filing bugs against TS:

<https://issues.isocpp.org/describecomponents.cgi>

##### 4.2 Future backlog discussions:

4.2.1 Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

4.2.2 Continue Retry discussion

[https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib\\_PFfc](https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib_PFfc)

[https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z\\_A](https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z_A)

##### 5.1 Establish next agenda

##### 5.2 Future meeting

Next call: Aug. 29

After that: Sept 12, Sept 26, Oct 10 (Oct 17 is mailing deadline), Oct 24.



## Minutes for 2016/08/29 SG5 Conference Call

Minutes by Mike Spear

Torvald, Tatiana,, , Jens Maurer, Maged,Hans, Victor, Michael Wong,Michael Scott, Mike Spear

Agenda:

1. Opening and introductions

1.1 Roll call of participants

Hans, Maged, Mike Spear, Michael Wong

1.2 Adopt agenda

Adopted.

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

Approved.

1.4 Review action items from previous meeting (5 min)

1.4.1 Victor: Summarize concerns about TM proposal expressed in email. Done

1.4.1. Hans will send draft wording for Issue 2: Done

1.4.2: Jens to follow up to see if anything needs to be done for Issue 3

1.4.3: Spear to write something on uniqptr inside tx: Done

2. Main issues (50 min)

## 2.1 Discuss defects

Issue 1: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/SMVEiVLbdig>

Issue 2: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Th7IFxFuIYo>

Hans sent an email about this topic on August 15, but did not receive any replies.

There are two ways to move forward. We can wait for Jens to provide input, or we can add this and wait for comments.

Conclusion: we will leave it on the reflector, so people have time to read and reply.

Issue 3: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/CXBycK3kgo0>

Issue 4: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Ood8sP1jbCQ>

## 2.2 Smart Ptr

<https://groups.google.com/a/isocpp.org/forum/#!topic/tm/TJ2oUYO6bkU>

This is a more substantial change than we are interested in making at this time. We agree that this is not worth pursuing at this time.

## 3. Any other business

Should we be considering the idea of limiting the TS to synchronized blocks?

- We don't have any fundamental technical issues with `atomic_cancel`, so we aren't wasting work by considering both `atomic_cancel` and `synchronized` right now
- But we need to have a good perspective on what impact would accrue if the committee asks that one or the other technique was dropped.

Conclusion: this should be a topic for a future meeting.

## 4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]

N4513 is the official working draft (these links may not be active yet until ISO posts these documents)

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf>

N4514 is the published PDTS:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf>

N4515 is the Editor's report:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html>

Github is where the latest repository is (I have updated for latest PDTS published draft from post-Leneaxa):

<https://github.com/cplusplus/transactional-memory-ts>

Bugzilla for filing bugs against TS:

<https://issues.isocpp.org/describecomponents.cgi>

4.2 Future backlog discussions:

4.2.1 Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

4.2.2 Continue Retry discussion

[https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib\\_PFfc](https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib_PFfc)

[https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z\\_A](https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z_A)

4.3 Review action items (5 min)

None.

5. Closing process

Meeting Adjourned.

5.1 Establish next agenda

5.2 Future meeting

Next call: Sept 12

Past and future Meeting dates

Aug 1: Discuss defects and email concerns

Aug 15: Discuss email and defects (Michael may not be able to call in due to flight)

Aug 29: Review Issue 2 and smart ptr (Michael may be only available for half hour, Victor away)

Sept 12:

Sept 26:

Oct 10: Oct 17 is mailing deadline

Oct 24

## Minutes for 2016/09/12 SG5 Conference Call

Minutes by Maged Michael

### 1.1 Roll call of participants

Victor, Michael Wong, Michael Scott, Mike Spear, Maged

### 1.2 Adopt agenda Adopted

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org  
Approved

### 1.4 Review action items from previous meeting (5 min)

1.4.1: Jens to follow up to see if anything needs to be done for Issue 3.  
Carry over.

## 2. Main issues (50 min)

### 2.1 Discuss defects

Issue 1: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/SMVEiVLbdig>

Issue 2: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Th7IFxFuIYo>

Issue 3: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/CXBycK3kgo0>

Issue 4: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Ood8sP1jbCQ>

Issue 1 is done.

Issue 2 is in review stage.

Issue 3 is pending.

Issue 4 needs discussion

Victor: Should wait until C++17 is finalized.

### 2.2 General discussion on which part of TS we should keep (Synchronized vs atomic blocks)

MW: The purpose of the discussion is to be ready for input from the committee.

MLS: Synchronized blocks semantics fit easier in the language than atomic blocks. Maybe give up on static checking. Tell users to use synchronized blocks in idiomatic ways that has a chance of performing well. No longer a strong opponent of prioritizing atomic blocks.

M Spear: Why did you prefer atomic blocks.

MLS: Cleaner semantics that don't need locking semantics. At the time didn't expect the di

M Spear: If we drop atomic blocks do we need language extensions.

MLS: To help HTM.

M Spear: I would rather use a special lock rather than a synchronized block because it makes it clearer to the user that synchronized blocks may deadlock under certain circumstances.

MLS: Not convinced that the syntax will matter to the programmer.

M Spear: How about Chandler's concern about the implied strong ordering?

Multiple people: Good point.

M Spear: Another issue is conditional synchronization.

Victor: It is unlikely that we'd want to apply the semantics for condition variables to the single global lock of synchronized blocks.

M Spear: Adding conditional synchronization to synchronized blocks is a solved problem. We could make it work with cond vars.

MLS: We should require cond vars that can be used in synchronized blocks to be associated with a special named lock.

M Spear: There are several known solutions known in the literature.

MLS mentioned a proposal by Mark Moir long ago for optionally named synchronized blocks.

MW: The committee asked for synchronized blocks.

Maged mentioned that using synchronized blocks to swap items between arbitrary data structures was a major motivation for the committee to be open to TM. Also mentioned the Issaquah challenge.

M Spear: This can lead to viral use of synchronized blocks in operating on the involved data structures (e.g., iteration).

Victor made a point that viral effects on synchronized blocks are not at the same level of the viral effect of transaction\_safe.

Victor: Concerned that synchronized blocks will not perform well.

M Spear gave example of cases where synchronized blocks will serialize because of how the program is written.

## Minutes for 2016/09/26 SG5 Conference Call

Minutes by Victor

Participants: Victor, Maged, Mike Spear, Michael Scott

The current secretary rota list is (the person who took notes at the last meeting is moved to the end)

Torvald, Tatiana, Jens Maurer, Hans, Michael Wong, Michael Scott, Mike Spear, Maged, Victor

1.4: Review action items:

1.4.1: Jens to follow up to see if anything needs to be done for Issue 3.

Jens not on call.

2. Main issues:

2.1 Discuss defects if any work done since last call.

No new work.

2.2 Discuss Torvald's email

Mike Spear summarizes issue: Use case in which a customer has a low-latency application.

Is this a use case where atomic is more valuable than synchronized?

That is, if using synchronized, the blocks may just serialize, which can have bad latency.

In contrast, if he used only atomic blocks, then it can't be serialized behind some other use of atomic blocks (perhaps in libraries).

Michael Scott:

Why are you getting better guarantees? Atomic blocks may be implemented by global locks, and even if not, if there is conflict, you may fall back to software.

Maged: Back in 2011, I pushed looking at what it would take to have nonblocking transactions.

We didn't go forward with that because they are not compatible with synchronized blocks,

so we either would have to eliminate synchronized blocks,

or we could have these nonblocking transactions operate on different domain (so no possible conflict).

Mike Spear:

What if we allow nonblocking transactions to throw an exception if it runs into a conflict with synchronized blocks?

Maged:

That doesn't allow a programmer to write a nonblocking transaction with a guarantee that we won't get be blocked.

Michael Scott:

How did we get to nonblocking transactions from low-latency guarantees?

Maged:

If we don't have nonblocking transactions, we can't guarantee low-latency because whatever is blocking us might take arbitrarily long.

Discussion about difference between nonblocking and low-latency

Michael Scott:

Allowing independent atomic blocks to be nonblocking is not possible without static separation.

Mike Spear:

Whole program optimization may allow this, at least for some TM implementations. But whether it is possible, it is really hard to do correctly.

Mike Spear:

If low-latency is what people think TM is going to get them, we have a marketing problem.

Victor:

I agree.

Maged:

I don't agree: it is possible.

Mike Spear:

It'll be a dance between programmer and implementation: with a particular implementation, it may be possible.

Maged:

Okay, but it really depends on what we mean by "low latency". Sometimes people really mean real-time guarantees (or close to it), but sometimes they just mean they don't want to get really bad cases, "being stuck". We can perhaps serve the latter set.

Summary:

Torvald's response seems reasonable to us.

Maged: something to keep in mind, but perhaps it's rare case.

Michael Scott:

If your program would behave reasonably most of the time if you wrote it with fine-grained locking, then it will probably behave reasonably most of the time if you wrote it with atomic blocks, and it'll be easier to program.

Mike Spear:

Example of how fine-grained is better than synchronized blocks and atomic blocks due to irrevocability, which requires all atomic blocks to be blocked out.

We need to relax some guarantees in order to be feasible.

Victor:

“Legitimate” use of synchronized blocks are uses in which code contains some unsafe code (or you want to preserve the option to use unsafe code in the future). Otherwise you should use atomic blocks.

Claim: Every legitimate use is bad for performance

(except case in which unsafe code may be executed, but only rarely).

If this is true, perhaps synchronized blocks shouldn't block out atomic blocks from executing.

Mike:

Commit actions may help reduce the need for synchronized block.



## Minutes for 2016/10/10 SG5 Conference Call

Minutes by Hans

Agenda:

### 1. Opening and introductions

#### 1.1 Roll call of participants

- Hans Boehm, Victor Luchangco, Michael Scott, Michael Spear

#### 1.2 Adopt agenda

1.3 Approve minutes from previous meeting, and approve publishing previously approved minutes to ISOCPP.org

1.4 Review action items from previous meeting (5 min)

1.4.1: Jens to follow up to see if anything needs to be done for Issue 3.

- Jens not here.

### 2. Main issues (50 min)

2.1 Discuss defects if any work done since last call

Issue 1: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/SMVEiVLbdig>

Issue 2: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Th7IFxFuIYo>

Issue 3: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/CXBycK3kgo0>

Issue 4: <https://groups.google.com/a/isocpp.org/forum/#!topic/tm/Ood8sP1jbCQ>

- Nothing really to be done with current attendees.

2.2 Discuss What we should submit, if any for Std mailing deadline Oct 17, 10 am ET

- General feeling that this was not pressing, especially in light of the fact that WG21 will be preoccupied with C++17 comment resolution.

2.3 General discussion on which part of TS we should keep (Synchronized vs atomics)

- Do we need atomic blocks?
- But atomic blocks are really the clean construct you should be using?
- Victor: Should take pushback seriously. But we shouldn't volunteer to remove atomic blocks.
- Michael Scott.: If you don't have `transaction_safe` in the type system, a lot of synchronized blocks will require serialization as a result of separate compilation (observation from past meeting).
- Victor: Maged (?) mentioned non-blocking transactions, Synchronized blocks interfere with that unless you can statically prove some sort of separation.
- Hans: Don't volunteer to drop anything But if we have to, drop `atomic_cancel` first.
- Michael Spear: Other arguments against `atomic_cancel`: Hard to implement. `May_cancel_outer` issue may reappear.
- But would remove the closest thing we have to retry.
- But retry doesn't force us to deal with the exception copying issues.

### 3. Any other business

- Michael Scott: Should we still be meeting regularly? What do we need to move forward?
- Need actual experience.
- Should complete "in transaction" and commit action work.
- Victor: Encouraged by Haskell experience, where TM is the standard way to synchronize.
- Issue for broader community. Transact and WTTM will be at PPOPP.
- Gcc support exists, but isn't polished. Don't know of Clang effort.

#### 4. Review

4.1 Review and approve resolutions and issues [e.g., changes to SG's working draft]

N4513 is the official working draft (these links may not be active yet until ISO posts these documents)

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4513.pdf>

N4514 is the published PDTS:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.pdf>

N4515 is the Editor's report:

<http://www.open-std.org/jtc1/sc22/wg21/docs/papers/2015/n4514.html>

Github is where the latest repository is (I have updated for latest PDTS published draft from post-Leneaxa):

<https://github.com/cplusplus/transactional-memory-ts>

Bugzilla for filing bugs against TS:

<https://issues.isocpp.org/describecomponents.cgi>

4.2 Future backlog discussions:

4.2.1 Write up guidance for TM compatibility for when TM is included in C++ standard (SG5)

4.2.2 Continue Retry discussion

[https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib\\_PfFc](https://groups.google.com/a/isocpp.org/forum/?hl=en&fromgroups#!topic/tm/qB1Ib_PfFc)

[https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z\\_A](https://groups.google.com/a/isocpp.org/forum/#!topic/tm/7JsuXIH4Z_A)

4.2.3 Smart Ptr

<https://groups.google.com/a/isocpp.org/forum/#!topic/tm/TJ2oUYO6bkU>

4.3 Review action items (5 min)

5. Closing process

5.1 Establish next agenda

5.2 Future meeting

Next call: Oct 24