Immutable ArrayBuffers

for stage 3

Mark S. Miller — Agoric

Peter Hoddie — moddable

Richard Gibson — Agoric

Jack-Works

107th Plenary

April 2025

TC39

Status

Stage 3

- [ ] committee approval
- [ ] merge test262 tests
- [x] write test262 tests
  - [x] immutable-arraybuffer tests #4445
- [ ] receive implementer feedback
  - [x] XS: implemented, tested, all good
- [ ] resolve all normative stage 4 issues
  - [x] Document the permanent bidirectional stability of immutable ArrayBuffer contents #44
  - [ ] Order of operations, when to throw ~~or silently do nothing~~? #16 was not a normative requirement for 2.7. It is normative for stage 3

~~Stage 2.7~~

Status

Stage 3

- [ ] commit
- [ ] merge t
- [x] write te
  - [x] im
- [ ] receive
  - [x] XS: implemented, teste all good
- [ ] resolve all normative stage 4 ssues
  - [x] Document the permanent bidirectional stability of immutable ArrayBuffer contents #44
  - [ ] Order of operations, when to throw ~~or silently do nothing~~? #16 was not a normative requirement for 2.7. It is normative for stage 3

25.1.3.2 AllocateImmutableArrayBuffer (*constructor*, *byteLength*, *fromBlock*, *fromIndex*, *count*)

The abstract operation AllocateImmutableArrayBuffer takes arguments *constructor* (a constructor), *byteLength* (a non-negative integer), *fromBlock* (a Data Block), *fromIndex* (a non-negative integer), and *count* (a non-negative integer) and returns either a normal completion containing an ArrayBuffer or a throw completion. It is used to create an immutable ArrayBuffer (i.e., an ArrayBuffer with a an [[ArrayBufferIsImmutable]] slot) with contents from *fromBlock*. The contents of an immutable ArrayBuffer's Data Block are constrained to be permanently stable, and may not be modified by either ECMAScript code or by other activities inside an implementation or observable by it. It performs the following steps when called:

~~Stage 2.7~~

Status

Stage 3

- [ ] committee approval
- [ ] merge test262 tests
- [x] write test262 tests
  - [x] immutable-arraybuffer tests #4445
- [ ] receive implementer feedback
  - [x] XS: implemented, tested, all good
- [ ] resolve all normative stage 4 issues
  - [x] Document the permanent bidirectional stability of immutable ArrayBuffer contents #44
  - [ ] Order of operations, when to throw ~~or silently do nothing~~? #16 was not a normative requirement for 2.7. It is normative for stage 3

~~Stage 2.7~~

Status

Stage 3

- [ ] committee a
- [ ] merge test2
- [x] write test26
  - [x] immuta
- [ ] receive implementer fe ack
  - [x] XS: implemented, ted, all good
- [ ] resolve all normative sta e 4 issues
  - [x] Document the pe rmanent bidirectional stability of immutable ArrayBuffer contents #44
  - [ ] Order of operations, when to throw ~~or silently do nothing~~? #16 was not a normative requirement for 2.7. It is normative for stage 3

As close as possible to `slice(s,e)`

25.1.6.8 ArrayBuffer.prototype.sliceToImmutable (*start*, *end*)

This method performs the following steps when called:

1. Let *O* be the **this** value.
2. Perform ? RequireInternalSlot(*O*, [[ArrayBufferData]]).
3. If IsSharedArrayBuffer(*O*) is **true**, throw a **TypeError** exception.
4. If IsDetachedBuffer(*O*) is **true**, throw a **TypeError** exception.
5. Let *len* be *O*.[[ArrayBufferByteLength]].
6. Let *bounds* be ? ResolveBounds(*len*, *start*, *end*).
7. Let *first* be *bounds*.[[From]].
8. Let *final* be *bounds*.[[To]].
9. Let *newLen* be max(*final* - *first*, 0).
10. NOTE: Side-effects of the above steps may have detached or resized *O*.
11. If IsDetachedBuffer(*O*) is **true**, throw a **TypeError** exception.
12. Let *fromBuf* be *O*.[[ArrayBufferData]].
13. Let *currentLen* be *O*.[[ArrayBufferByteLength]].
14. If *currentLen* < *final*, throw a **RangeError** exception.
15. Let *newBuffer* be ? AllocateImmutableArrayBuffer(%ArrayBuffer%, *newLen*, *fromBuf*, *first*, *newLen*).
16. Return *newBuffer*.

~~Stage 2.7~~

Status

Stage 3

- [ ] committee approval
- [ ] merge test262 tests
- [x] write test262 tests
  - [x] immutable-arraybuffer tests #4445
- [ ] receive implementer feedback
  - [x] XS: implemented, tested, all good
- [ ] resolve all normative stage 4 issues
  - [x] Document the permanent bidirectional stability of immutable ArrayBuffer contents #44
  - [ ] Order of operations, when to throw ~~or silently do nothing~~? #16 was not a normative requirement for 2.7. It is normative for stage 3

~~Stage 2.7~~

tc39 / test262

Type / to search

Code | Issues 199 | Pull requests 46 | Actions | Projects | Wiki | Security | In

merge immutable-arraybuffer tests #4445

Open **phoddie** wants to merge 1 commit into `tc39:main` from `Moddable-OpenSource:main`

Conversation 0 | Commits 1 | Checks 5 | Files changed 50

phoddie commented 3 days ago — Member

Suite of test262 tests for the Immutable ArrayBuffer proposal.

merge immutable-arraybuffer tests — b77ff72

phoddie requested a review from **tc39/test262-maintainers** as a code owner 3 days ago

erights mentioned this pull request 14 minutes ago

Path to Stage 4! tc39/proposal-immutable-arraybuffer#1 — Open

44 tasks

Review required

Code owner review required by reviewers with write access.

1 pending review

Some checks were not successful

1 failing, 10 successful checks

1 failing check

Checks / Lint tests (pull_request) Failing after 28s — Required

Questions? Stage 3?

Stage	Status	Entrance Criteria	Purpose
3	The proposal has been recommended for implementation. No changes to the proposal are expected, but some necessary changes may still occur due to web incompatibilities or feedback from production-grade implementations.	• The feature has sufficient testing and appropriate pre-implementation experience	Gaining implementation experience and discovering any web compatibility or integration issues.