

Custom Descriptors & JS Interop

Thomas Lively, January 27, 2026

Agenda

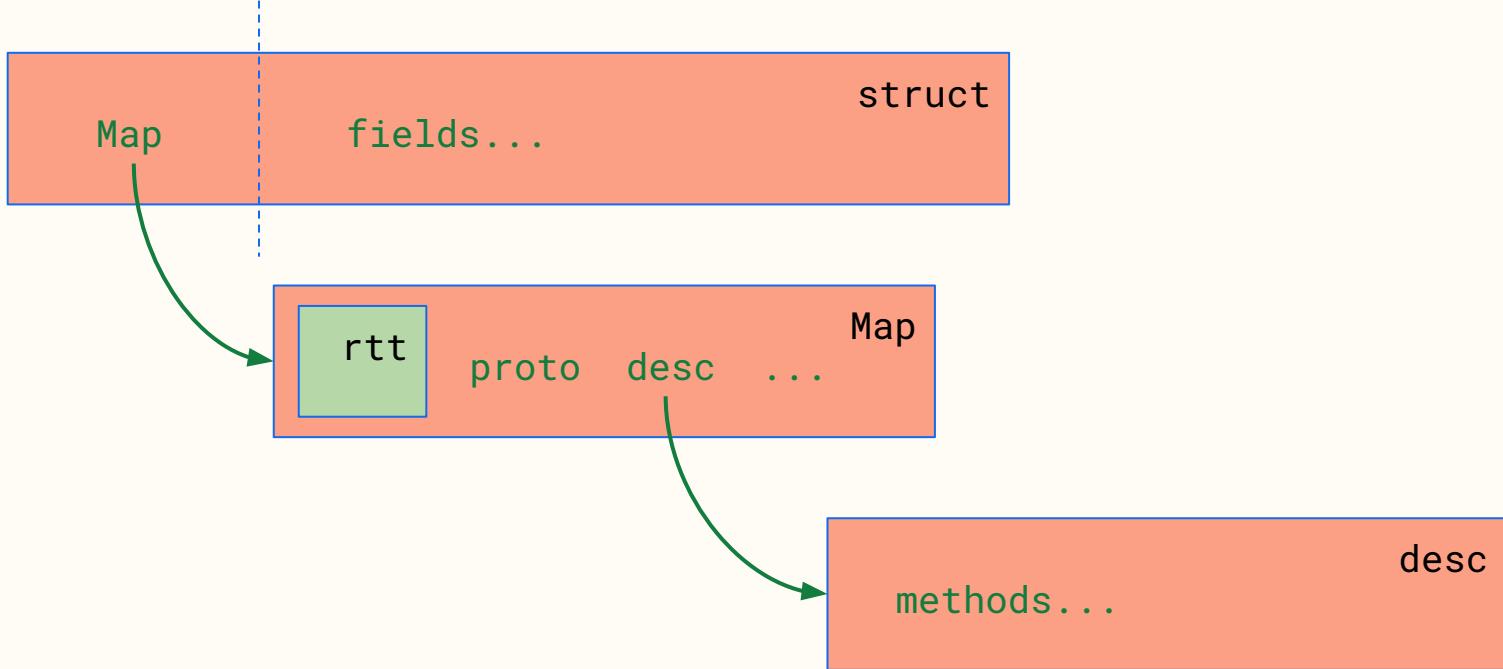
- 1 Updates since October
- 2 Performance Experiments
- 3 Type System Complexity
- 4 Open Questions
- 5 Phase 3 Poll

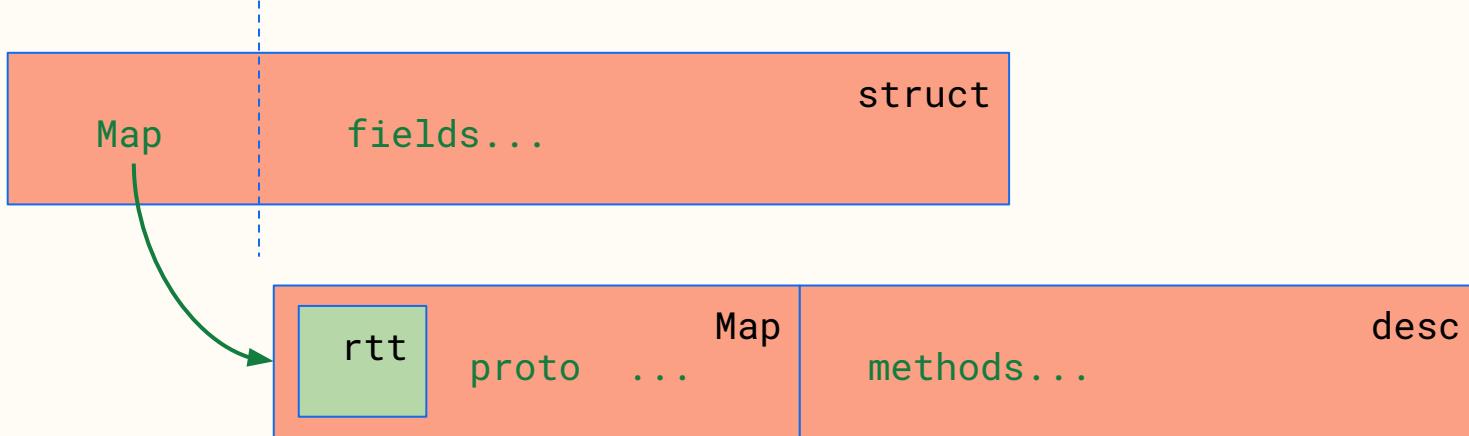
Updates

- Fixed encoding of exact function imports (bit 6 of extertype)
- Add `struct.new_desc` and `struct.new_default_desc`
- Added JS API tests

Performance

- Merged descriptor with Map / Shape





Performance

- Merged descriptor with Map / Shape
- Make null value inaccessible

Average Performance Improvement vs. Shipping Configuration

V8 Patches w/o CD	CD before patches	CD after patches
0.31%	-1.55%	1.00%

Type System Complexity

Mutual Recursion

```
(rec
  (type $A (descriptor $B) (struct ...))
  (type $B (describes $A) (struct ...)))
)
```

Type System Complexity

Mutual Recursion

```
(rec
  (type $A (struct (field $descriptor (ref null $B)) ...))
  (type $B (struct (field $describes (ref null $A)) ...)))
)
```

Type System Complexity

Mutually Recursive Subtyping Constraints

```
(rec
  (type $super (sub (descriptor $super.desc) (struct ...)))
  (type $super.desc (sub (describes $super) (struct ...))))
)
(rec
  (type $sub (sub $super (descriptor $sub.desc) (struct ...)))
  (type $sub.desc (sub $super.desc (describes $sub) (struct ...))))
)
```

Type System Complexity

Mutually Recursive Subtyping Constraints

```
(rec
  (type $super (sub (struct (field (ref null $super.desc)) ...)))
  (type $super.desc (sub (struct (field (ref null $super)) ...))))
)
(rec
  (type $sub (sub $super (struct (field (ref null $sub.desc)) ...)))
  (type $sub.desc (sub $super.desc (struct (field (ref null $sub))...))))
)
```

Future Reified RTTs

type	descriptor	meta-descriptor
foo		
bar		
baz	baz.desc	
quux	quux.desc	quux.meta

New `ref.cast_desc` takes a descriptor as a handle to an RTT

Future Reified RTTs

type	descriptor	meta-descriptor
foo	(desc foo)	(desc (desc foo))
bar	(desc bar)	(desc (desc bar))
baz	baz.desc	(desc baz.desc)
quux	quux.desc	quux.meta

New `ref.cast_desc` takes a descriptor as a handle to an RTT

Open Questions

- Should we rename `ref.cast_desc` to `ref.cast_desc_eq`? ([#81](#))
- Do we need to support descriptors of descriptors? ([#43](#))
- Should the prototypes or methods passed to `configureAll` be non-null? ([#88](#))
- Will we want to split JS Interop into a follow-on proposal?

Phase 3 Requirements

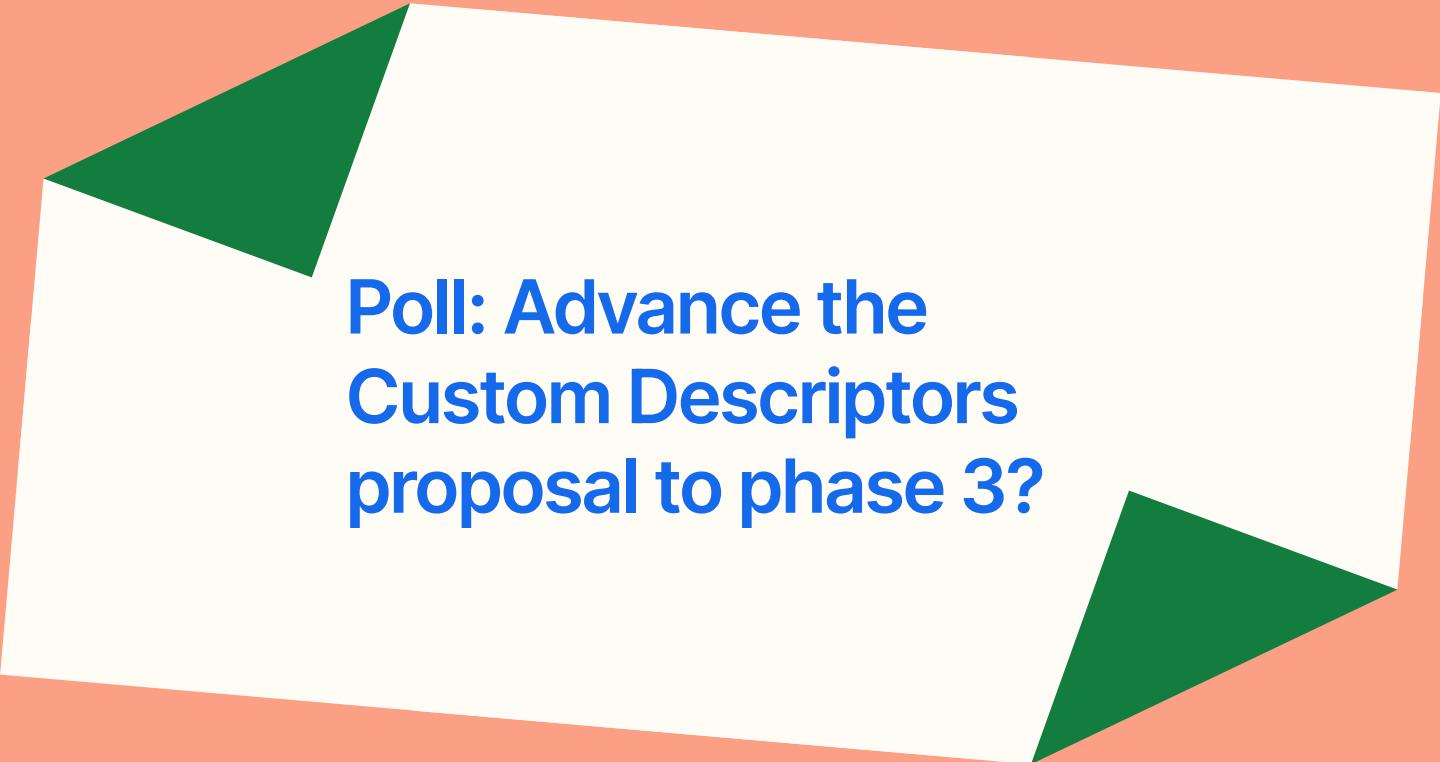
- Test suite has been updated to cover the feature in its forked repo.
- The test suite should run against some implementation, though it need not be the reference interpreter.
- Updates on the actual spec document and reference interpreter are NOT yet required (but can happen earlier).

During Phase 3

- Engines implement the feature (where applicable).
- The spec document in the forked repo is updated to include the full English prose and formalization.
- The reference interpreter in the forked repo is updated to include a complete implementation of the feature.
- The feature is implemented in toolchains.
- Remaining open questions are resolved.

During Phase 3

- Engines implement the feature (where applicable).
- The spec document in the forked repo is updated to include the full English prose and formalization.
- The reference interpreter in the forked repo is updated to include a complete implementation of the feature.
- The feature is implemented in toolchains.
- Remaining open questions are resolved.



**Poll: Advance the
Custom Descriptors
proposal to phase 3?**