CLC activities 2021–2024: a personal memoir

Andrew Lelechenko
andrew.lelechenko@gmail.com

MuniHac, 12.10.2024

What is CLC?

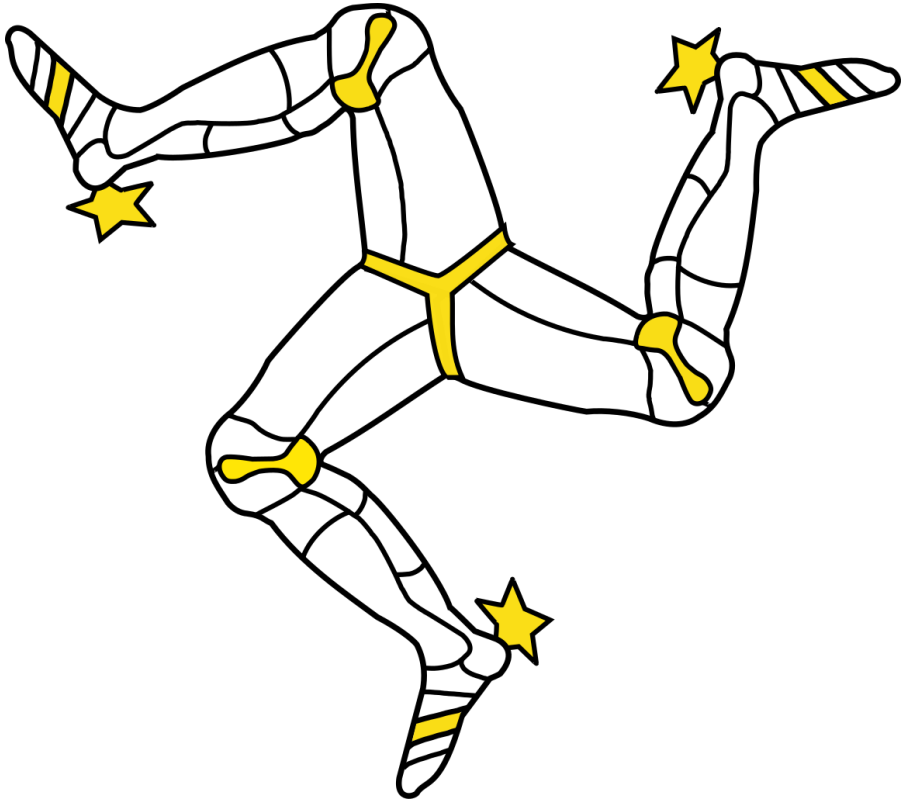
- CLC stands for Core Libraries Committee.
- Core Libraries are libraries governed by CLC.
- `let x = x in x`

Quiz time — 1

- CLC appoints Hackage trustees.
- CLC maintains `bytestring` package.
- CLC approves changes to `Data.Text`.
- You need CLC assent to add new GHC RTS flags.
- CLC can amend Haskell Package Versioning Policy.
- CLC owns `haskeline` and `terminfo` packages.
- Bug fixes in `base` do not need CLC assent.
- Changes to `ghc-internal` are out of scope for CLC.
- Moving implementation of `base` elsewhere involves CLC.
- CLC is responsible for stability of `base`.
- Adding class laws for `Num` falls under CLC purview.
- Documentation for `Data.List` must be verified by CLC.

CLC purview

- User-visible changes to API of `base` package.
- Governance of designated so called “Core” packages.
- Haskell Package Versioning Policy aka PVP.

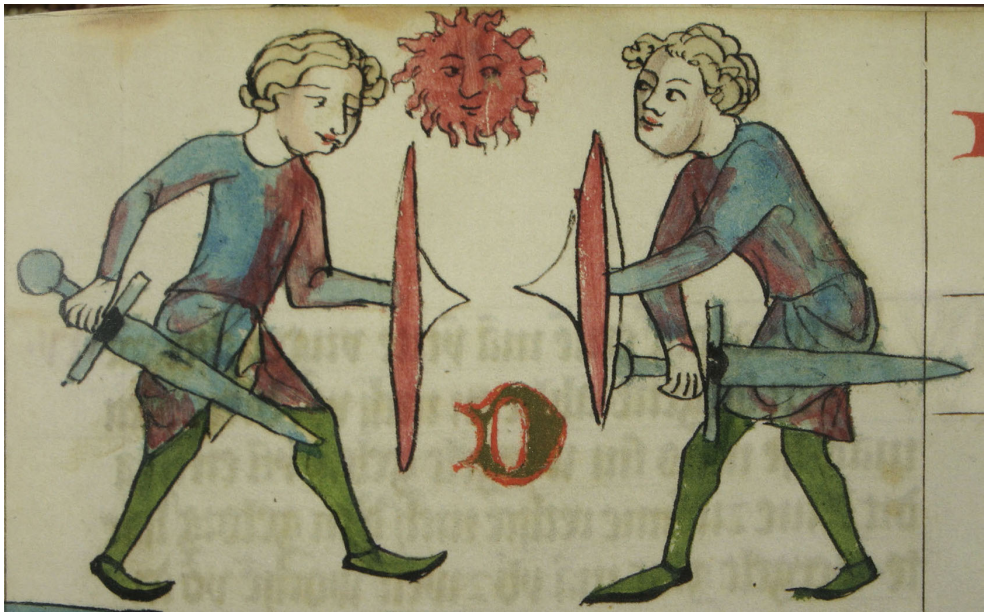


Quiz time — 2

- CLC appoints Hackage trustees. No.
- CLC maintains `bytestring` package. No.
- CLC approves changes to `Data.Text`. No.
- You need CLC assent to add new GHC RTS flags. Yes.
- CLC can amend Haskell Package Versioning Policy. Yes.
- CLC owns `haskeline` and `terminfo` packages. Yes.
- Bug fixes in `base` do not need CLC assent. Yes.
- Changes to `ghc-internal` are out of scope for CLC. No.
- Moving implementation of `base` elsewhere involves CLC. No.
- CLC is responsible for stability of `base`. No.
- Adding class laws for `Num` falls under CLC purview. Yes.
- Documentation for `Data.List` must be verified by CLC. No.

PVP: Haskell Package Versioning Policy

- PVP 1.0 lives at `https://pvp.haskell.org`
- PVP 1.1 is work in progress since 2019.
- PVP contains normative and non-normative sections.
- Non-normative changes (FAQ, decision tree, etc.) can be made by CLC alone.
- The definition of breaking changes and versioning schema require also Hackage admins to agree.



Boot and core packages — 1

Package	Boot	Core 2021	Core 2024
array	+	+	+
binary	+		+
bytestring	+	+	+
Cabal	+		
Cabal-syntax	+		
containers	+		
deepseq	+	+	+
directory	+	+	+
entropy			+
exceptions	+		
file-io	+		+
filepath	+	+	+
haskeline	+		+
hpc	+		
mtl	+	+	+
os-string	+		+

Boot and core packages — 2

Package	Boot	Core 2021	Core 2024
parsec	+		
pretty	+		
primitive		+	+
process	+	+	+
random		+	+
stm	+	+	+
template-haskell	+	+	+
terminfo	+		+
text	+	+	+
time	+		
transformers	+		+
unix	+	+	+
vector		+	+
Win32	+	+	+
xhtml	+		+

CLC proposals — 1

- By summer 2021 CLC was in crisis, down to 2 active members.
- In August CLC was rebooted with a goal to provide for better engagement of committee members.
- In September a fresh batch of CLC members was elected.

- To be able to get things done,
  we got a new home for proposals at GitHub instead of mail.
- To manage community expectations,
  we developed a lightweight, but precise proposal process.
- To prevent burning out,
  we brought our remit in line with our capabilities.

- CLC proposal process is asynchronous, transparent and written communication.

CLC proposals — 2

- Total number of CLC proposals: **249**
- Rate of proposals: **7** per month
- Approved proposals: **125**
- Declined proposals: **9**

- Median time from creation to decision: **39** days
- Average time from creation to decision: **80** days
- Median time from creation to approval: **50** days
- Average time from creation to approval: **81** days

- The fastest approval: **2** hours
  for *Use HasCallStack and error in GHC.List and .NonEmpty*
- 2^{nd} fastest approval: **34** hours
  for *Remove default method from Bitraversable*
- 2^{nd} slowest approval: **375** days
  for *Exception backtrace proposal: Part 4: Rethrowing*
- The slowest approval: **395** days
  for *base changes for exception backtrace proposal*

CLC proposals — 3

- Total activity: **6302** comments
- Median activity per proposal: **17** comments
- Average activity per proposal: **25** comments
- Median activity per approved proposal: **21** comments
- Average activity per approved proposal: **30** comments
- The least active approved proposal: **1** comment
  for *Export List from GHC.List*
- 2^{nd} least active approved proposal: **6** comments
  for *Make casts between words and floats real primops*
- 2^{nd} most active: **171** comments
  for *Remove method (/=) from class Eq*
- The most active: **177** comments
  for *Add warning to Data.List.{head,tail}*

CLC proposals — 4

Release	Date	Count
base-4.16	28 Feb 2022	**1**
base-4.17	7 Aug 2022	**11**
base-4.18	11 Mar 2023	**24**
base-4.19	9 Oct 2023	**25**
base-4.20	14 May 2024	**33**
base-4.21	TBA	**21**

- Open proposals: **13**
- Median age for open proposals: **6** days
- Average age for open proposals: **63** days
- The newest open proposal: **1** day
  for *Formally deprecate modules informally marked unstable in #146*
- The oldest open proposal: **691** days
  for *Make NonEmpty functions less gratuitously lazy*

How to get your proposal approved?

- Be patient.
- Be persistent.
- Be clear.
- Be precise.
- Stand on the shoulders of giants: find prior art.
- Think hard about possible breakage and its mitigation.
- Follow the ~~white rabbit~~ proposal process.

CLC Election January 2025

- Bandwidth to analyse and contribute opinions for 6–8 proposals per month.
- Communication and governing-the-commons skills.
- Determination to see proposals through to their timely completion.
- We are not looking for the biggest galaxy brain in the room.
- We hope to build a broad sample of the community.

How you can help CLC

- Be welcome to participate in CLC discussions.
- Pick up an abandoned proposal and work towards completion.
- Volunteer to write CLC blog at Haskell.org.
- Raise more new proposals.
- Like and subscribe to `github.com/haskell/core-libraries-committee`.

Thanks all!