A framework for Preview1, Preview2, Preview3, and WASI 1.0

Preview1

- Supported in lots of languages
- Supported in lots of engines
- In use in some production environments
- Support existing users
  - https://github.com/WebAssembly/WASI/pull/510
  - Add documentation, clarify behavior, write tests

Preview2: Wit

- Answer long-standing questions with Preview1
  - Modularize WASI?
  - What is a file descriptor?
  - What should be WASI’s job, and what should happen in lower-level standards?
  - How should functions returning strings or lists work?
  - How to use WASI from languages that aren’t C?
  - How to make WASI interfaces fully virtualizable?
  - We now have answers to all of these!
- Incorporate lessons learned
  - File descriptor “rights” bitfield is not worth the complexity for WASI
  - fd_readdir is too complex
  - etc.
- Add features, like sockets, timezones, file locking

Preview3: Integrated async

- Component-model async proposal
- Integrated async: Two new types in Wit
  - `future<T>`: A single value that you might have to wait for
  - `stream<T, E>`: A sequence of values, followed by an end value
- Advantages
  - Composability
  - Idiomatic source-language bindings
  - Typed streams

WASI 1.0

- Standardization
  - Work with the Wasm CG
  - Standardization is all about process
- Not all Interfaces or all Worlds will be ready for standardization at the same time.
  - Standardize what’s ready (phase 4)
  - Look forward to WASI 2.0, and beyond

The framework

- Preview1: Support existing users, portability
- Preview2: Rebase WASI on Wit
- Preview3: Level up Async: `future` and `stream`
- WASI 1.0: Standardization

- At each step, Preview2, Preview3, and WASI 1.0, we’ll incorporate real-world implementation feedback and add features.
- Preview3 and WASI 1.0 may make breaking changes if needed.
- But, the transitions to Preview3 and WASI 1.0 will be smoother.