

PMIx Standard ABI

- Initial discussion on mailing list in September '21
 - OpenMP headers as example of "standard" API definitions (<https://github.com/OpenMP/sources/tree/main/include>)
- <https://github.com/pmix/pmix-standard/issues/365>
 - Collected issues and todo items

PMIx Standard ABI - Developments

- PMIx headers define an Application Binary Interface (ABI)
 - <https://github.com/pmix/pmix-abi>
 - constants
 - data structures
 - functions
 - macros
 - Successfully tested by compiling with ABI headers, running with libpmix.so
 - ABI compliance checker used to verify contents against OpenPMIx headers
 - TODO: automate testing

PMIx Standard ABI – Developments

- ABI Stability Guidance PR
 - <https://github.com/pmix/governance/pull/34>
 - Stable ABI additions OK in minor releases
 - Stable ABI breaking changes (removals, changes) only in major releases
 - Provisional ABI changes OK in any release
- Standard ABI PR
 - <https://github.com/pmix/pmix-standard/pull/375>
 - Explain what is the standard ABI is defining and how it can be used

PMIx Standard ABI - Developments

- Constants
 - Assign values to constants in standard
 - <https://github.com/pmix/pmix-standard/pull/359>
 - “Boundary” constants not yet assigned values in standard
 - Reading/vote Q2 2022 meeting
 - <https://github.com/pmix/pmix-standard/pull/394>
 - Already in pmix-abi repo
- Macros
 - Several macros in OpenPMIx call internal library functions not in standard
 - Convert those macros to functions in the standard (and pmix-abi headers)
 - <https://github.com/pmix/pmix-standard/pull/383>
 - Static inline “support” functions
 - pmix_abi_support.h and pmix_abi_support_bottom.h
 - Defined with “pmixabi_” prefix to differentiate from standard APIs

PMix Standard ABI

“You never know if an API is really useful, until you have too many people using it to ever be able to change it.”—Greg Kroah-Hartman

- Data structures
 - pmix_key_t (512 byte char array)
 - <https://github.com/pmix/pmix-standard/issues/367#issuecomment-1017079248>
 - pmix_nspace_t (256 byte char array)
 - <https://github.com/pmix/pmix-standard/issues/366#issuecomment-1017083494>
 - Concern that allocating arrays containing these types uses space inefficiently
 - Leave as-is for backward compatibility
 - pmix_value_t
 - Could limit the size to sizeof(void*) by changing any larger elements to pointers
 - Leave as-is for backward compatibility
- Functions
 - Already well-defined. Nothing to do.

PMIx Standard ABI Query

- Q: How to determine if a dlopen library is compatible?
 - dlopen bypasses library version checks normally done by the linker
- A: Use PMIx_Query_info
 - <https://github.com/pmix/pmix-standard/pull/397>
 - PMIX_QUERY_STABLE_ABI_VERSION
 - PMIX_QUERY_PROVISIONAL_ABI_VERSION
 - Report "major.minor" ABI version(s). Minor updates are backward compatible. Major updates have backward incompatibilities.
 - Example: <https://github.com/pmix/pmix-abi/pull/10>

PMIx Standard ABI

- What next?
 - <https://github.com/pmix/pmix-standard/issues/365#issuecomment-1095650601>
 - All ABI items announced
 - If any updates or changes needed, discussions in the IAWG
- PMIx Standard v5 timeline (major)
 - All v5 items announced and read by Q2 2022
 - Vote/merge for final release candidate Q4 2022
 - Ratification Q1 2023