

Tango Roadmap

Presenters Thomas Braun, Thomas Juerges (volunteered by the organisers)

The roadmap as presented in 2023:

- New alarm event  (Tango 10: IDLv6)
- Enhanced logging system and distributed tracing support  (Tango 10: IDLv6, experimental)
- Warning and alarm hysteresis  (Tango 10: IDLv6)
- New DevInfo_6 with version information  (Tango 10: IDLv6)
- New datatype DevDict  (Tango 11, IDLv7)
- Multi-parameter commands  (Tango 12, IDLv8)
- Multi-dimensional arrays  (Tango 13, IDLv9)

Has the roadmap aged well?

Yes.

- Implemented 75% of major new features
 - Only hysteresis support missing

But...

- Only 75%: Did not implemented 25% of planned features

Why?

- **Lack of manpower**

So?

- 👉 Changed release cycle
 - Minor releases: Deadline has priority
 - Major releases: Scope has priority
 - 👉 Planning for major version upgrades could become a challenge

Major new features on our to do list

- Warning and alarm hysteresis
- New datatype DevDict (std::map/dict like type)
- Multi-parameter commands (removes one parameter limit)
- Multi-dimensional arrays (DevSpectrum & DevImage become specialisations of n-dimensional arrays)
- Encryption support (CORBA & ZMQ, new since 2024)

Or is adding new stuff **NOT** the right thing to do?

Maybe we should first fix bugs?

In cppTango: ~40 issues labelled as bug [\[1\]](#)
(all levels of severity):

- segfault
- latent segfault
- buggy behaviour
- wrong documentation

- New major release per feature as initially planned?
 - Pro
 - More releases more often: IDLv7, IDLv8, IDLv9
 - New features can build on each other
 - Con
 - Facilities might not want to update that often
- All new features in one major release?
 - Estimate: Will likely take the same time as the above
 - Pro
 - Just a big bang release with all new features
 - Facilities have more time to prepare
 - Con
 - Facilities will have to wait longer for an individual feature
- Or bug fixes only?

 **Moving this discussion right now to a Working Group session**

Goals:

- **Find out if all our (yours & Tango kernel) priorities align**
- **Or is there a compromise that we can come to?**

Sponsoring: Write the RFC, write tests, implement the feature, use it

- Finish Open Telemetry
 - Events
 - Runtime configuration
- Encryption (**sponsored**)
 - RFC, tests, implementation
- LTS releases every 3 years
 - LTSes overlap for 2 years
 - LTS release supported for 5 years
- DevDict (**sponsored**)
 - RFC, tests, implementation
- Alarm & Warning hysteresis (**sponsored**)
 - RFC, tests, implementation

- Maintenance
 - Clean up code
 - Remove notifyd
 - Remove pipes (depends on DevDict being available)
 - Locks refactored
- Authentication, Authorisation, Accounting: SIG delivers RFC
- Multi-parameter commands (depends on DevDict)
- C++ High level API (**sponsor needed!!!!**)
- n-dimensional arrays
- Dynamic attributes on device level (**sponsored by ALBA**)