

Meeting	Java/C++ Kernel developers meeting
Date	23-24.11.2016
Location	Synchrotron Soleil
Version	1.0 – distributed to Tango SC
Contact	Reynald Bourtembourg reynald.bourtembourg<at>esrf<dot>fr
Participants	Gwenaëlle Abeille – Soleil Reynald Bourtembourg – ESRF Alain Buteau – Soleil Andrew Götz – ESRF Igor Khokhriakov – IK Frédéric-Emmanuel Picca – Soleil Nicolas Leclercq – Soleil Pascal Verdier – ESRF

Introduction

A Tango Java kernel developers meeting was held on 23rd-24th November 2016 at Soleil synchrotron. Present were Tango kernel Java core developers, some Tango kernel C++ core developers, their managers as well as a Debian integrator. The goals of the meeting were to present the status of the current migration of Tango kernel core development to git as well as the current C++ core re-factoring for Tango V10 and to help to move the projects currently maintained by Soleil in Sourceforge tango-cs to Github and to discuss about Tango V10 development.

Agenda

- Presentation of the current status of the migration to git
- Presentation of the work done by Igor on refactoring the C++ API for Tango V10
- Debian packages maintenance
- Move of Tango kernel Java API to GitHub and mavenization
- Help in moving projects maintained by Soleil from Sourceforge to GitHub

Migration to Git Status

Igor presented the latest status of the migration of the tango-cs projects from Sourceforge to GitHub.

At the time of the meeting, the following projects had already been moved to GitHub:

- cppTango

Tango kernel developers meeting @ Soleil – November 2016 – Minutes

- Tango-idl
- PANIC
- Fandango
- Taurus
- SimulatorDS
- TangoTest
- TangoDatabase
- TangoAccessControl
- JTango
- Astor
- LogViewer
- Atk & co
- Jive
- Pogo
- Starter

The ticket migration was not achieved yet for all of these projects.

To get an up to date status of the migration, please visit: <https://github.com/tango-controls/svn2git-migration/wiki>.

The projects listed above which are in Java have been restructured and mavenized. They support automatic release generation. Dependencies are configured using Maven. ATK tools set can be submitted to Jcenter. It was agreed to use bintray as the official maven repository: <https://bintray.com/tango-controls/maven>.

Igor converted the following projects to CMake:

- TangoTest
- TangoAccessControl
- TangoIDL
- Starter

Action: Tiago has been asked to transfer PyTango to tango-controls organization on GitHub and to remove tango-cs GitHub organization.

C++ API refactoring

cppTango project has been restructured and is now using CMake.

The files generated from the IDL are now generated automatically during CMake configuration phase. CMake is able to generate cxx tests, generate doxygen documentation and to create a Debian package.

Continuous Integration was set up using Travis on cppTango. Docker containers had to be created in order to be able to run tests on Travis. The following repositories have been created:

- Docker-mysql
- Tango-cs-docker

A tango-9-lts branch has been created for the long term support of Tango 9 from GitHub. It is possible to use CMake to compile this branch on Debian 8. It is still possible to use configure and traditional Makefiles to build the tango-9-lts branch.

Java API refactoring

No major need for refactoring on the server part.

The client part would need a refactoring to be purely Java (and not C++ like code) and the notifd event management removed.

Tango V10 (C++)

It was confirmed as proposed at the previous kernel meeting that absorbing CORBA is the way to go. This means making TANGO support a pluggable protocol with the first plugin being CORBA. The dependency on CORBA (omniORB in C++) would be removed and the code needed by Tango moved into the plugin as part of the Tango library. For Java the dependency on JacORB would be removed and replaced by the OMG classes (already in the standard Java library).

Proof of concept refactoring were done on cppTango to:

- isolate the ORB
- implement DevVarDouble type using an architecture allowing plugins
- replace omni_thread with C++ 11 threads

A big effort has been spent on trying to rewrite the polling mechanism (triggered by the move from omni_thread to C++ 11 threads).

Igor presented some of his ideas of improvements which could be implemented in the future in cppTango (see slides here <https://www.slideshare.net/secret/yMtds3IsCnUiRS>):

- Merge ApiUtil and Util
- Util is basically a DS, i.e. merge Util and server

Tango kernel developers meeting @ Soleil – November 2016 – Minutes

- DeviceProxy (Database) has a connection (not inherit)
- Replace ifs with polymorphism (Db, FileDb; DeviceServer, PyDeviceServer etc)
- Replace map<device,smth> with corresponding Device field
- Distinguish Heartbeat, StoreSubDevices and Polling threads
- Implement event driven service bus

The main idea of having an event driven architecture would be the following:

- Transport (currently ORB) layer generates events for incoming requests (attr read/write, command, pipe, ...)
- Server generates internal events: new attribute value, state changed, etc...
- Micro services as the ones listed below could listen and react on these events:
 - AccessControl
 - Logging
 - Blackbox
 - polling
 - attribute read/writer
 - command executor
 - user defined services

Action: Igor will try compiling on Windows to evaluate the amount of work required to build with CMake on Windows.

Tango V10 (Java)

Actions:

- Setup the github continuous integration travis server. SOLEIL has to provide Igor the configuration needed in TangoDB for the tests to run.
- Remove the IDL2 server API since it is deprecated
- Study the possibility to remove the dependency to Jacorb and rely directly on the standard java ORB implementation.
- Test the integration with openjdk
- Ideas discussed on client refactoring:
 - Add standard logging (SLF4J), no dependency to an implementation (log4j, logback...)
 - Better Exceptions: should not rely on the generated DevFailed but on pure Java exceptions
 - Merge the 2 two TangORB modules (“common” and “dao”). The issue is that SOLEIL is still using the code organization for the project WebTangORB which relies on this separation. -> *I have checked at SOLEIL with my colleagues, we can merge these modules and SOLEIL can maintain a local version of the “common” module. Are you certain that this project is no more used at ESRF?*
 - Merge «TangORB», «JTangoClientLang» and «EZ» modules in a single one to integrate all Tango client services
- Server: add annotations for Events
- Add the new features decided by the TangoV10 roadmap

According to Gwenaëlle, the amount of work seems to be “reasonable” compare to the C++ part. She estimates the amount of work to 6 man-months maximum.

The Tango EC may decide how to organize the work and if it is possible to sub-contract some of it.

Tango V10 roadmap

Action: Andy and Alain will prepare an e-mail describing the roadmap.

The basic roadmap is:

- V10.0: absorb omniORB + protocol API V0.1
- V10.1: refactoring + enhancements
- V10.2: refactoring + more enhancements + protocol API V1.0

Tango kernel task force

Igor emphasized the fact that a re-factoring is necessary in order to be able to maintain the C++ API code without the help from Emmanuel.

Some help from Emmanuel will be necessary at the beginning in order to guide the new maintainers in order to understand the existing code and design.

The management of the whole TangoV10 project should be common for Java and C++ and all sub-tasks should be split into sub-task forces.

The main roles of this task forces members should be at least the following:

- *Analytics*: define the roadmap and priorities, organize work, documentation, test cases (implemented by Developer and QA), business logic
- *Quality Assurance*: code reviews and acceptance tests, **integration tests**, feedback on API and business logic
- *Developers*: code and unit/integration tests, code quality, unit tests, packaging, distribution

The ESRF will be responsible to coordinate the task forces which should be created at the beginning of 2017.

Once the C++ kernel task force will be created, a training should be organized with Emmanuel for all the members of this task force. The goal of this training will be to explain:

- the current architecture
- the important design choices
- what must be kept
- what could be dropped/modified
- the tricks

Ideally, all the Tango C++ kernel task force members should be present at this training in order to minimize the time spent by Emmanuel to transfer his knowledge.

A coordination will be necessary between the different languages supported by Tango in order to harmonize the features supported by the different languages.

Action: Tango kernel task forces should be created with members from several institutes in order to help to develop and maintain the Tango kernel.

Debian packaging

Frédéric Picca described quickly the current Tango Debian packages release process.

The Debian general release management process is documented on <https://release.debian.org/>.

Frédéric insisted on the fact that we need to focus on the migration scripts to be able to support MariaDB (new default DB in Debian9) and MySQL (still available).

He also explained the coming milestones for Debian 9 release:

- December 5th: Forced 10-day migration delay
- January 5th: soft freeze
- February 5th: full freeze

This basically implies that all the problems related to the new Debian release should be fixed at the latest 10 days before the 5th February (~26th January).

A solution has been found to solve the incompatibility between Tango and the latest ZMQ version 4.2.0 before this date. More info on this subject can be found on the bug reports which have been created:

- for ZMQ: <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=844479>
- for Tango: <https://bugs.debian.org/cgi-bin/bugreport.cgi?bug=743508>

Tango Debian packaging will still rely on autotools to build the next Tango release (9.2.5), then CMake will probably be used.

Action: Add default device servers installation directory variable in tango package config file. You can follow this action on the Pull Request [#311](#) from cppTango GitHub repository.

Help in the migration to Git

Labview, Matloab and IgorPro bindings have been moved to Github.

A GIT training is foreseen at Soleil at the beginning of December.

Action: The other tango-cs projects maintained by Soleil will be moved to Github after this training.