

TANGO DOCUMENTATION & PACKAGING

“This improved a lot since I started using Tango (three years ago) from scratch so I'm happy to see the efforts from the developers. Still there is room for improvements that can help a new developer exposed to Tango for the first time.”

Documentation requests by SKA DISH

- Improve documentation and coherency
- More How To
 - 1) How to define READ_WRITE attributes and properly write their read_XXX and write_XXX methods.
I needed again to search for community devices to understand.
 - 2) How to define different kind of events in a device and subscribe from a client invoking a callback on receipt
 - 3) How to create and fill pipe blobs (also complex)
 - 4) How to create command response for some types: DevVarLongStringArray. This can be found in the manual, but I was asked by sub-elements on how to set command response.
- Instructions how to build Pogo

Documentation requests by SKA CSP

- Documentation for installation, Java servers, ...

Documentation requests by SKA AAVS + LFAA

- Improve documentation of PyTango and TANGO tools

Documentation requests from SKA TelMgt, GMRT and TCS

- Documentation, documentation, documentation, ... long list in forum post

Documentation requests from SKA TM

- Technical software documentation - understanding **how the framework works**, for instance how it realizes the polling mechanism works and so on. Ideally an UML class diagram together with other activities diagrams could help a lot in understanding it.
- E/R model for the database: for instance, what happens if you add a device? what are the main tables affected? So basically I am talking about technical software documentation.
- Hard to find documentation about the "**built in**" **tango tools**, there is a list of tools on the tango home page, but for many it does not even mention the name of the executable.
- Many things are **quite spread out**. "Official" Tango components have documentation hosted over several sites. Many links between these sites are dead, due to web reorganisation?
- Bit of confusion about where **source code repos** live, PyTango moving to github was somewhat hidden if you were not already part of the community.
- an architecture overview for TANGO,
- diagrammatic representations of how it all fits together (here included enabling things like Taurus).

Two types of documentation:

- Newby help – how-to's and examples – for Tango users
- Technical documentation on the framework architecture – for Senior software engineers that really want to understand the inner workings

Quick wins ?

- An index of **everything** that already exist:
 - current docs
 - packages, source repositories
 - tools
 - device servers
- Quick cook book by the community, through on-line collaboration

Using Sphinx for other things than code:

yes, others do use it this way

used for common management of documentation

Suggestions:

- How to make the system run (start db, add db, etc).
There is a readme, but it just needs to be updated, good base
- Command line for "services"
- Installation guide for different platforms
- Selection of a suite of tools to make a "system":
provide a catalogue;
list all solutions for each category (e.g. archiving, alarms handling, GUIs);
update Tango tool map and links
- Invest in packaging, to solve dependencies, with documentation
- Start with Description of system, diagrams and relationships
- FAQ can be useful, collect answers to questions in a "formal" place
- Generate user-stories (how-to's)
- Best practices for solving certain problems (like the SKA use cases)
- Advanced users – missing UML diagram for core features – e.g. heartbeats, connection management, alarms.
Can help for fault finding problems in these areas
- Use cases
- Test suite for newcomers (there is a C++ based test suite already that can be made available)
- 10 points you need to know about Corba (C++) (e.g. how to duplicate strings) [it is in the Tango book]
- Incorporate Tango Device server guidelines to the Tango book, add diagram; explain best practices for using different command invocation methods;

Suggestions 2

- Examples – same example using different tools
- Take the suggestions and categorise the types of documentation required
-