

Improving the Tango Documentation in 2016

by A.Götz for Tango Controls

Version of 31/8/2016,

Introduction

Although the Tango Controls documentation is fairly extensive, newcomers (but also experienced users) complain that it is difficult to find, read and a number of concepts and features are not documented. Therefore the Tango Controls Collaboration has decided to improve the documentation by subcontracting some of the tasks for improving the documentation to an external company. This document describes the first phase of improving the documentation which will be done in 2016. Not all tasks will be subcontracted and the main responsibility will still be on the community to write good documentation which is easy to read. This document will mainly describe what needs to be done in 2016 to unify the documentation and make it easier to find, read and maintain in the future.

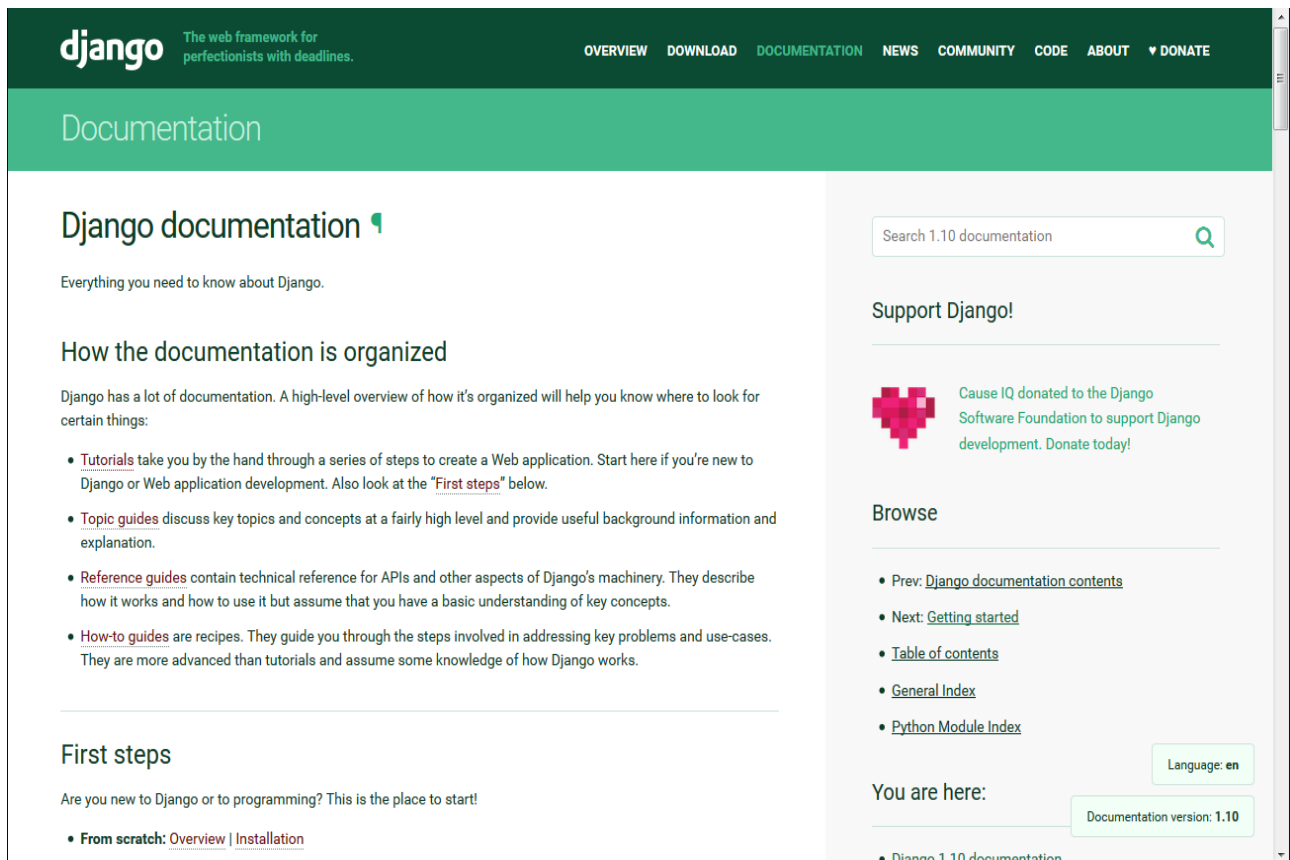
Vision

Our vision is to have a complete set of high quality documentation covering all aspects of Tango Controls. The documentation should be easy to read, write, and maintain. It should follow best practices of documentation for open source software. It must be available for multiple versions of Tango and be online and available for downloading in common formats (pdf, epub, etc.). It should be easy for developers to write. It should attract contributions from non-developers and not only be seen as the burden of the developers.

Best practices

In terms of the best practices [Django](https://docs.djangoproject.com/en/1.10/) is the open source project which is often cited as having excellent documentation. Looking at the Django documentation page

<https://docs.djangoproject.com/en/1.10/>



Example of good documentation for an open source project - Django web framework

We can see from the Django documentation that Tango Controls needs (at least) Tutorials, Topic guides, Reference guides and How to guides and they should all be easily accessible from one page.

Django uses the Sphinx (<http://www.sphinx-doc.org>) to write and generate the documentation.

What exists

A number of guides and documents exist for Tango:

1. **Tango Controls System Handbook** – the “Book” as it is known is the best reference guide for Tango. It is slightly outdated in some sections (especially Getting Started) and difficult to read in others (for example Setting up Access Control). It is written with Lyx and requires Lyx to be edited. There is no online version since version 9 (only pdf).
Author(s) – Emmanuel Taurel
2. **Tango Device Server Guidelines** – a set of Guidelines for writing Device Servers.
Authors – Alain Buteau, Nicolas Leclercq, Emmanuel Taurel, Pascal Verdier, Jens Meyer, Jean-Michel Chaize, Andy Götz, Nexeya Team, ...
3. **Tango Java Device Server User Guide** – how to write device servers in Java
Author(s) – Gwenaelle Abeille

4. **C++ API classes reference guide** – generated from the source code using doxygen
Author(s) – Emmanuel Taurel
5. **Java API reference guide** – generated from the source code using javadoc
Author(s) – Pascal Verdier, Gwenaelle Abeille
6. **Python classes reference guide** – generated from the source code using sphinx (?)
Author(s) – Tiago Couthino, Vincent Michel, ...
7. **PyTango Getting Started, Quick Tour and How To guides** – written with sphinx, available on readthedocs (<https://pytango.readthedocs.io>). Often cited as an example of good documentation for Tango due to the examples and contents.
Author(s) – Tiago Couthino, Vincent Michel, ...
8. **Pogo** – online only (no pdf). Mixes old and new versions in same document. Not detailed enough and difficult for beginners.
Author(s) – Pascal Verdier
9. **Jive** – recently updated and rewritten with sphinx. Online only i.e. no downloadable file (yet).
Author(s) – Jean-Luc Pons
10. **Astor+Starter** – online version only. Up-to-date but could be extended to have more text.
Author(s) – Pascal Verdier
11. **Mango** – link broken
Author(s) -
12. **Archiving** – includes HDB, TDB and SNAP. Has extensive documentation but it is not online instead it is embedded in the zip file with the jar files etc. Has been updated recently but would need revision to be sure it is complete for beginners.
Author(s) – Raphael Girardot, ...
13. **PyTangoArchiving** – online only.
Author(s) – Sergi Rubio
14. **Mtango** – very complete documentation on Bitbucket. Online only.
Author(s) – Igor Kokhrakiov
15. **ATK** – complete tutorial (pdf file only) and reference documentation generated with javadoc from source.
Author(s) – Faranguiss Poncet, Jean-Luc Pons
16. **Taurus** – complete doc available on <http://www.taurus-scada.org/>. Written with sphinx. Available for multiple versions.
Author(s) – Carlos Pascual, Tiago Couthino
17. **Qtango** – documentation mainly in the form of presentations

Author(s) – Giacomo Strangolino

18. **Canone** – link broken

Author(s) -

19. **Fandango** – function programming with Tango is available online on the old website (plone.tang-controls.org)

Author(s) – Sergi Rubio

20. **YAT, YAT4Tango** - library documentation generated by doxygen and available: http://www2.synchrotron-soleil.fr/controlle/docs/yat4tango/yat4tango_html/

Author(s) – Nicolas Leclercq

21. **Panic** – brief documentation available online. Needs to be extended to be more complete.

Author(s) – Sergi Rubio

22. **Matlab**: there's a README and each Tango related function is self documented (online documentation - e.g. `help tango_read_attribute`). Difficult and time consuming to port the online doc (in .m files) for the 90 functions.

Author – Nicolas Leclercq

23. **LabVIEW**: there's a README and a lot of undocumented examples. The upcoming version (i.e. the one with server support) has a Quick Start Guide.

Author – Nicolas Leclercq

24. **Igor Pro**: there's a README and some documented (commented) examples. The existing "documentation" must be updated in order to focus on the embedded code generator (the latter completely change the way the binding is supposed to be used).

Author – Nicolas Leclercq

25. **Tango VM** – documentation is available (pdf and as Word document) but needs updating for Tango 9

Author(s) – Andy Götz, Cosylab

26. **Device Server User Guides** – hundreds of guides are available online automatically generated from the xmi definition file by a script run nightly. Many of the guides are very basic and need to be extended with much more text to become full guides.

Author(s) – Tango Controls community

27. **How To** – a number of how to guides exist online on the tango-controls.org website.

Author(s) – Tango Controls community

28. **Recipes** - a number of how to guides exist online on the tango-controls.org website. The distinction between Recipes and How Tos is not so clear. They could be converted to How Tos.

Author(s) – Igor Khokhrakiov, Tango Control community

What is missing

In addition to missing or outdated documents the main problem with the current Tango documentation is it is not unified, not easy to produce and different versions are not easy to find. The following documents are missing for newcomers (in order of priority):

1. **Installation** – a complete up-to-date installation guide for Linux and Windows from binary and source packages.
2. **Getting started** – an up-to-date getting started guide to writing a device server and managing a Tango-based control system.
3. **What is Tango** – a guide on the main concepts of Tango, what is a Tango-based system is, what Tango provides
4. **How to manage Tango** – a topic guide on how to manage a Tango system
5. **Cross references** – the documentation should be cross referenced to avoid repeating concepts each time
6. **Reference docs integration** – the reference documentation should be included in the general documentation
7. **Device Server User Guides** – should generate sphinx compatible documentation

TO DO

The following tasks have been identified (in order of priority):

1. **Convert existing doc to Sphinx** - a decision was taken at the last Tango Meeting in Toulouse in 2016 to adopt Sphinx as the main documentation tool. The first step therefore is to convert the existing documentation to Sphinx. A first attempt was made by Thomas Braun and posted on the forum:
<http://www.tango-controls.org/community/forum/post/1599/>
2. **Move documentation to central git repository** – the community has decided that the Tango source code should move to github. The documentation must move there too as a separate repository.
3. **Update and complete existing documentation** – some of the existing documentation is out-of-date and needs updating and completing
4. **Writing the missing documents** – see above for list
5. **Easy access to documentation** – make documentation look like a coherent whole with easy access from a single page (follow example of Django).
6. **Unify look and feel** – use same template for all Sphinx documents. If need be develop a Tango specific one
7. **Integrate reference documentation** – link the main documentation to the doxygen/javadoc generated documentation
8. **Cross reference documentation** – make cross references between all documents and publish the list of references
9. **Publish multi-versions of documentation online and offline** – integrate

documentation into website (if not possible link to external page with similar look and feel) and make multiple versions available online and downloadable as pdf, epub etc.

10. **Provide a getting started guide for writing documentation** – the documentation will need to be continuously updated and new documentation needs to be written for new projects. Users will need a simple guide how to write documentation which is compatible with the Tango documentation guidelines.

Conclusion

The Tango Controls documentation needs to be re-factored to be easy to find, read, complete and maintainable. This document outlines the state of the current documentation and identifies what needs to be done in a first step to improve the documentation. Please send your comments on what else should be done in the first phase of the Tango Docs project.

References

1. <http://www.writethedocs.org/guide/> - useful information and tips on writing documentation
2. <https://github.com/PharkMillups/beautiful-docs> – list of beautiful / good documentation for open source software which can be used as inspiration of how to write good and beautiful documentation

Appendix – Comments from the Community

The following comments were made on the forum as part of RFC on Tango V10 :

<http://www.tango-controls.org/community/forum/c/general/rfc/tango-v10/>

They highlighted the need for improving the documentation and pointed out which parts are missing.

1. Simone Riggi for SKA DISH

(<http://www.tango-controls.org/community/forum/post/1563/>)

- The documentation is still a bit fragmented. Adding more on tools would be great, for example Astor (each time I realize I need to remember how I did some stuff)/Jive. It would be useful to show a complete example covering: "How to create a Device with Pogo", "how to register your device with Jive". Documentation on how to create a device with additional user thread (Yat/omniThread) and shared data/signals is useful. I needed to pick devices from the community to understand a bit the practices used.

- Populating the HowTo would be perfect for some recurrent task that currently you have to search/extrapolate (not so immediate) from the manual. Some possibility are:

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.

2. Sonja for SKA CSP

(<http://www.tango-controls.org/community/forum/post/1564/>)

Need to improve documentation:

a) Document on Java Server lacks useful and complete example of a TANGO Server with multiple devices (at least two instances of two different devices) and the data base configuration for that (how to add devices to the database).

b) Most people I talked to encounter the same problem: installation steps are not well documented.

c) Suggestion: hire a developer/technical writer who would generate detailed instructions for installation for all three supported languages, and for two or more flavours of Operating Systems. Include concrete examples and detailed

instructions how to download and install TANGO, Taurus, data base, and get whole system running. Create instructions and examples for the case when the code for TANGO Servers and Devices are generated using POGO and for the case when developer choses to implement everything manually.

d)Improve the overall description. In many cases well-known general concepts are discussed in detail, and TANGO specific details are skimmed over.

3. Vatsal for SKA TelMgt

[\(http://www.tango-controls.org/community/forum/post/1566/\)](http://www.tango-controls.org/community/forum/post/1566/)

Suggestion on Tango Documentation

- Provide step-by-step guide for installation of Tango Control System Framework and associated tools such as Taurus, HDB++ Archiver for multiple operating systems such as Windows, Ubuntu, Fedora, RedHat Linux.*
- Chapter #1 Introduction of the Tango Control System Manual starts with introduction/explanation of Tango Device Server. For first time readers it is difficult to follow the details provided in this chapter (Chapter #1). Instead this chapter shall provides a brief description of the Tango Control System Framework and its core concepts such as Tango Device Server, Tango Device, Tango Class. Details of the Tango Concepts shall be addressed in the subsequent chapters.*
- Known errors and resolutions shall be included in the documentation- One has to browse through the forum, to be able to come to know about it.*
- Details related to preferred operating system platforms/ variants / versions etc. shall also be mentioned in the Tango documentation.*

Specific to the Tango Tools and Components

- Tango Access Control (TAC) - Very little documentation available. Detailed instructions are required for someone to get started with it. More information on configuration to use TAC is required. For example, checking/setting class properties, how to configure multiple hosts etc. is missing in the documentation. Information about some limitations like inability of Starter DS to directly start/stop DS written in JAVA/python through Astor, if mentioned, would be helpful.*
- POGO and JIVE - There are many options provided with these tools, but how the selection of specific choice affect the code / behavior etc. is not available*
- Tango Logging Service - The documentation of TANGO logging service, especially for Java specific implementation is inadequate.*
- TAURUS - Comparatively, documentation for PyQT and Taurus is good. It can be further improved by including relevant examples*
- HDB++ and PANIC - Installation procedure for HDB++ and PANIC needs to be simplified. Manual of these tools, should have a clear information about configuration of archiving/alarm rules through .csv/.txt files. All available Syntax/ functions supported for configuring the alarms rule needs to be documented properly. Details of HdbEventSubscriber/HdbConfigurationManager DS commands should be explained with example.*

4. Matteo for SKA TM.LMC

(<http://www.tango-controls.org/community/forum/post/1587/>)

We can only try to give you our perspective of what is important to have for a documentation.

The main difficulties we found was on 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. Another documentation missing is an 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.

5. Paul for SKA TMS

(<http://www.tango-controls.org/community/forum/post/1587/>)

From my (little) exposure to tango 9 documentation, I would like the following to be added via sections and diagrams:

- an architecture overview for TANGO,
- diagrammatic representations of how it all fits together (here included enabling things like Taurus).

6. Neilen for SKA SA LMC

(<http://www.tango-controls.org/community/forum/post/1587/>)

Comments about TANGO documentation / Finding stuff

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.

7. Rajesh, Ralph, and Uli for SAT.LMC provided a detailed analysis of their experience with TANGO including the documentation. Their comments are very useful and must be taken into account when improving the documentation. A copy of their report can be found here: <https://cloud.esrf.fr/index.php/s/DZiaUxF4Ejw1S4S>

8. Guifre for ALBA would like to see the following explained in the documentation :

We would like to add some points that are difficult to find or missing and could help people running a Tango system for fixing bugs, debugging, optimizing, etc..

The following points could be explained in details and even sketched in form of the UML diagrams:

** Command execution, attribute read/write processes and how they differ (if so) in case: the client and the server are the same process, the client and the server runs on the same PC (Linux, Windows), the client and the server are distributed over the network that could help in debugging "CORBA Transient Timeout Errors"*

** Establishing and maintaining connection between the client and the server*

** Event system extreme situations and how the infrastructure (e.g. network) can affect them (e.g. loosing or interleaving of events)*

** Explain "Heartbeat" and "Keep Alive" mechanisms*