ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-MOPHA050

TOWARDS IMPROVED ACCESSIBILITY OF THE TANGO CONTROLS*

P. P. Goryl, M. Liszcz, S2Innovation, Kraków, Poland
A. Götz, R. Bourtembourg, ESRF, Grenoble, France
V. Hardion, MAX IV Laboratory, Lund, Sweden

Abstract

Tango Controls is successfully applied at more than 40 scientific institutions and industrial projects. These institutions do not only use the software but also actively participates to its development. The Tango Community raised several projects and activities to support collaboration as well as to make Tango Controls being easier to start with. Some of the projects are led by S2Innovation. These projects are: gathering and unifying of Tango Controls documentation, providing a device classes catalogue and preparation of a so-called TangoBox virtual machine. Status of the projects will be presented as well as their impact on the Tango Controls collaboration.

DOCUMENTATION

Initial State

Until recently, the Tango documentation consisted mainly of The Tango Book (a big pdf file provided with the source distribution) and many other documents in different formats (pdf, word, html, etc.). The book contained precious information but it was maintained by a single person and it was difficult to get external contributions. In 2015, it was decided to merge all the available documentation in a single location and to make it easy to contribute to. In order to encourage contributions the idea was to write documentation like code [1]. It was then decided to use Sphinx [2] and to convert and publish all the tango-controls related documentation on *readthedocs* [3].

Tools

The existing documentation was converted to *reStructuredText* [4] and imported into a GitHub repository. Keeping the documentation in a git repository allows the use of standard git flow procedure. The contributions are accepted as pull requests which are then reviewed before merging into a published branch. Thanks to Travis [5], the pull requests are validated prior to the merging. The converted documentation is next published with the use of the Sphinx toolset on *readthedocs*. The *readthedocs* service allows for providing multiple versions of the documentation, making them accessible also for the previous releases of Tango Controls.

New Documentation Structure

The book and various other documentation, including tools' manuals, HOW-TOs and tutorials were grouped into sections suitable for users with different roles and different levels of experience. Such a structure allows the reader to find the right section depending on whether she/he is an end-user, a developer or a system administrator. The intended audience is stored as a meta-data within each document.

Community Involvement

The original documentation of the Tango Controls was written by the community members. However, before it was made available in a public repository, only its authors were able to contribute to specific documents. Now, any community member can propose improvements and contribute to the new Tango Controls Documentation.

TANGO CONTROLS DEMONSTRATION VIRTUAL MACHINE

Goal

There are a lot of tools and libraries in the Tango Controls ecosystem. Although most of the tools including Tango Controls kernel have straightforward installation, newcomers may find it difficult to choose the configuration suitable for their needs. The availability of multitude of tools and applications for the Tango Controls makes it overwhelming for new users who would not like to invest much time into investigating technical details. The so-called TangoBox, a Tango Controls demonstration virtual machine image, provides a way to use the ecosystem without the need to spend time on investigation, selection and configuration of most of Tango Controls tools.

The additional goal is to provide a working configuration example. User may look how certain applications are deployed and configured. There are also shell scripts which serves as installation process documentation and can be referenced for custom deployment.

Contents

The TangoBox comes with pre-installed Tango core libraries and tools, including various desktop and web applications. These are available as launchers, directly from its desktop.

JLinac provides a GUI application backed by simulation device servers. It is a demo based on a real application from the ESRF control room. Jive and Astor let the user learn how Tango Controls administration works. Pogo, JDraw, TaurusDesigner allows starting developing for Tango Controls. There are examples of generic GUI applications and widgets provided in Java, Python and C++ (ATK, Taurus, Cumbia). TangoBox contains various device servers, including a Modbus device server. It is also possible to interact with the Tango Controls via web applications, like Waltz, JupyTango and Egiga.

* Work supported by Tango Controls Collaboration

There is also a release for running on Amazon AWS cloud. This release features the Remote Desktop Protocol.

Towards Containerisation

Over the years, Docker [6] has been widely adopted as a technology for containerizing applications. Since the past few releases of the TangoBox, device servers and other services are progressively moved to dedicated containers running within the virtual machine. This improves maintainability, enables better manageability and endorses software reuse. Running most of the services in their own containers allows for seamless upgrades of these services as well as eases the process of upgrading the Ubuntu operating system. In the latest 9.3 release of the TangoBox, the following services are containerized:

- Archiving suite (including HDB and HDB++ device servers and E-giga web application),
- Waltz web application (without the REST server),
- JupyTango (with complete Jupyter stack),
- JLinac accelerator simulator (multiple device servers),
- Modbus device simulator.

All client applications, the DataBase device server and the SQL database server are still running directly inside the virtual machine. There are plans to move the database-related services into dedicated containers.

Build Process

Before the 9.3 release, the build process of the virtual machine was fully manual. Starting with the 9.3 release, the virtual machine can be provisioned using a set of Bash scripts. The scripts can be run on a fresh virtual machine instance in order to install all the software and configure all the services. Others can use these scripts as a reference on how to install Tango Controls and related software on a clean system. Still, some device servers require manual configuration so unattended installation is not possible. The VM provisioning scripts are available online [7] under LGPL license.

The dockerized part of the virtual machine is built automatically using the GitLab continuous integration services. The images are hosted at the GitLab container registry and are pulled into the virtual machine by the provisioning scripts. In order to improve maintainability, a common image with basic libraries and tools was created. Other images build on top of this common image. Relations between the images are shown in Fig. 1. The Dockerfiles are available online [8].

Use Cases

The TangoBox can be useful in a wide range of scenarios. Not only it provides a demonstration of Tango Controls capabilities and features but also can be used by developers for building and testing new software. A pre-configured virtual machine also facilitates organization of trainings and workshops.

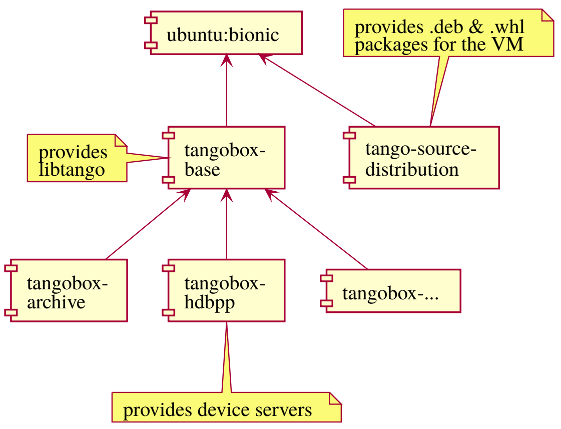

Figure 1: Docker images used by the TangoBox.

DEVICE CLASSES CATALOGUE

Goal

New device classes for accessing all kinds of hardware are developed by the Tango Controls Community all the time. A web application called *Device Classes Catalogue* [9] has been developed to allow for effective search if a particular device is already supported. This application is meant to facilitate code sharing and prevent duplication of work.

Catalogue

The catalogue stores the references to the repositories or the institutes (URL or contact information) where a specific device class can be found. Each device class is associated with meta-data which includes, among others, information on supported hardware, class family, programming language, operating system and license. The catalogue stores also information on device classes interfaces (attributes, commands, pipes and properties). This indicates the completeness of support for a particular hardware.

The application is developed within Django framework and integrated with Tango Controls webpage.

Catalogue Contents

Automatic update is provided by so-called *import script* [10]. The *import script* can search an SVN repository or use a CSV file or take the information provided from environment variables. The import script can also parse java or python source code or HTML documentation to generate metadata for the catalogue.

Previously, most of Tango Controls institutes use a shared SVN repository on Sourceforge, *tango-ds* [11]. Nowadays, most of the institutes keep their development in their repositories. However, the shared SVN is still in use by some of the institutes, and it stores a lot of useful source code. There is a weekly run job which checks for changes in the repository and updates the catalogue (Fig. 2).

As it is now, the catalogue lists more than 750 device classes. The catalogue is about to be updated with device classes developed at more institutes.

New development has been made in order to increase the number of device visible in the catalogue. By using the

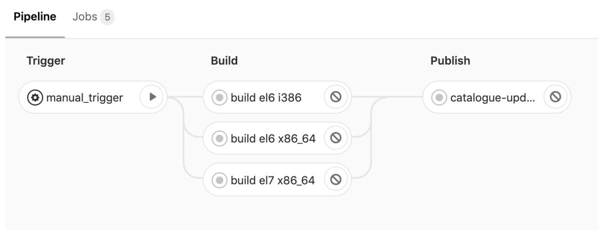

Figure 2: Example of CI/CD pipeline triggering an update of the Tango Catalogue for a particular Tango device.

Continuous Integration pipeline of the different institutes the Tango server can be registered or updated to the Tango Catalogue more systematically. The main advantage is to be agnostic to the type of repository e.g svn or git based, private or public, github or gitlab. The first prototype has been developed with the MAX-IV Gitlab Continuous Integration/Continuous Delivery (CI/CD) pipeline [12]. A script is triggered automatically collecting the information on device classes and uploading to the Tango Catalogue. Some information can be missing since not all Tango server code comes with an xml descriptive file (usually generated by Pogo). So the script can takes environment variable to set custom information about the equipment such as the manufacturer, the class type, the model or the communication bus.

CONCLUSION

The new Tango Controls Documentation collects all relevant information in a common place. Having a single, unified source of information is especially useful for people new to Tango. Storing documentation in a GitHub repository makes it easier for the Community to collaborate on improving it.

The TangoBox virtual machine allows newcomers to quickly start working with Tango Controls. It also supports experienced users, developers and system administrators in their day-to-day work.

The Device Classes Catalogue promotes open-source software, endorses software reuse and prevents duplication of work. The developers of Tango device servers can quickly verify if someone already implemented a device class for their hardware. The import script and Continuous Integration services facilitate the process of publication new classes to the catalogue.

The development, usage and maintenance of the documentation, TangoBox and Device Classes catalogue facilitate collaboration within the Community.

ACKNOWLEDGEMENTS

All the work related to the documentation, TangoBox and Device Classes Catalogue was done with direct or indirect contribution of many Tango Controls Community members. The list of all whom authors would like to thank could be too long to fit into this paper. However, authors would like to mention here the following persons with their significant input:

- Thomas Braun, for doing the first conversion of the Book to sphinx,
- Olga Merkulova, for re-organising the documentation and writing getting started,
- Lukasz Zytniak, for converting many documents to Sphinx,
- Igor Khorniakov and Łukasz Dudek, for setting up CI for documentation,
- All others who participate in the documentation. They are listed on the related section [13],
- All developers of Tango Controls tools and libraries, for support in the preparation of TangoBox,
- All developers and users of device servers, for the spirit of sharing and defining requirements for the Device Classes Catalogue.

REFERENCES

[1] A. Gentle, *Documentation like code*, https://www.docslikecode.com

[2] Sphinx, https://www.sphinx-doc.org/en/1.5/index.html

[3] Tango-Controls documentation, https://tango-controls.readthedocs.io

[4] http://docutils.sourceforge.net/docs/user/rst/quickstart.html

[5] Travis, https://travis-ci.org

[6] Docker, https://www.docker.com

[7] https://github.com/tango-controls/tangobox

[8] https://gitlab.com/s2innovation/tangobox-docker

[9] https://github.com/tango-controls/dsc

[10] https://github.com/tango-controls/dsc-import

[11] https://sourceforge.net/projects/tango-ds/

[12] https://docs.gitlab.com/ee/ci/

[13] https://tango-controls.readthedocs.io/en/latest/authors.html