Tango core

- News from kernel
  - Tango library 8.1
  - Source distribution 8.1
- Tango 8 @ ESRF
- Tango logo
- Tango survey
- Executive committee

- Continuous Integration @ ESRF
- Tango kernel test system

Team

- After 18 months, Tomasz has left Alba for a RTW trip!

Tango library 8.1.2

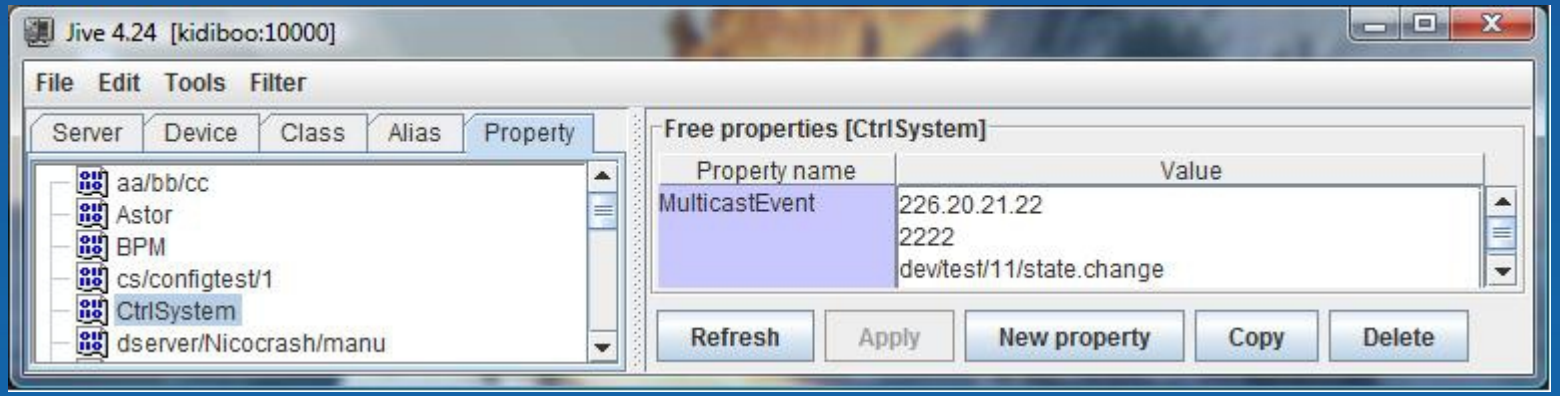
- Mostly a bug fixes release nevertheless
  - Multicast event propagation available (C++ only)
  - DeviceImpl::write_attr_hardware()
  - Move servers from one CS to another
  - Miscellaneous small changes
- Bug fixes
  - 22 (+) SourceForge recorded bug fixes
  - 5 not recorded in SourceForge
  - Detailed list in TANGO_CHANGES file (in distrib)

Tango library 8.1.2 - Multicast

- Event unicast propagation is STILL the default
  - Multicast requires specific configuration
- Multicasting using the Pragmatic General Multicast (PGM) protocol – RFC 3208
  - OpenPGM implementation
  - Supported by ZMQ (--with-pgm during ZMQ compilation)
  - PGM is a reliable multicast protocol

Tango library 8.1.2 - Multicast

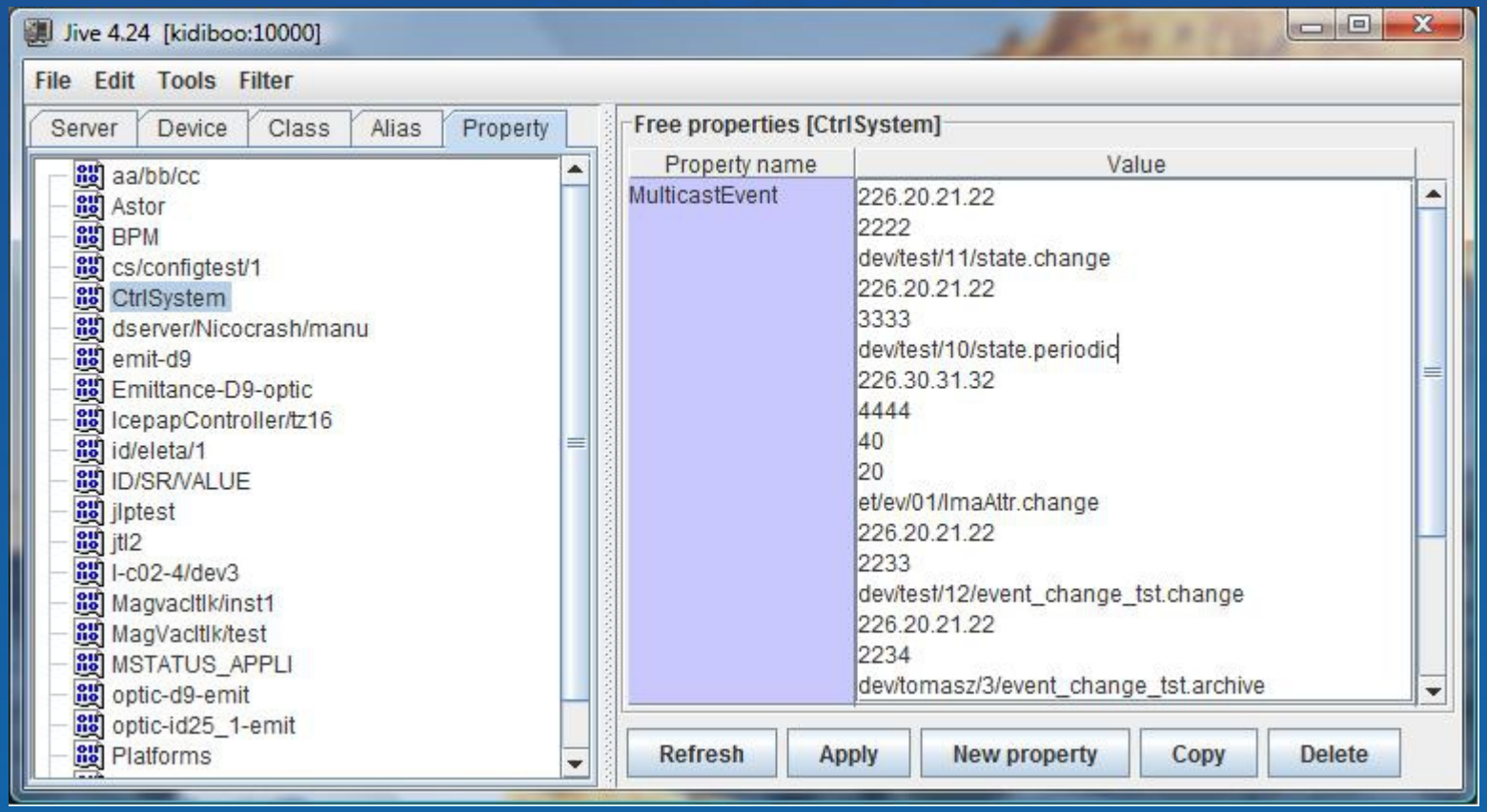
- Configuring one multicast event
  - Choose the IP address and port used for transmission
    - IP between 224.0.1.0 and 238.255.255.255
  - Define the **MulticastEvent** free property belonging to the **CtrlSystem** object
  - The event name



Tango library 8.1.2 - Multicast

- The complete definition of properties for multicast event is:
  - Multicast address
  - Port number
  - [Rate (Mbit/sec)]
  - [IVL (sec)]
  - Event name
- Rate and IVL are optionals
- PGM is rate limited. **Rate** is the max bandwidth used by PGM
- **IVL** is the max time one receiver could be absent from network before data loss happens
  - Requires memory buffer
    - IVL = 60 for Rate = 1024 means 7 Gbytes buffer !! (on DS side)

Tango library 8.1.2 - Multicast



Tango library 8.1.2 - Multicast

- Multicast rate and ivl properties default value defined by properties
  - CtrlSystem → MulticastRate
    - Rate: 80 Mbit/sec hard coded in library if not defined
  - CtrlSystem → MulticastIvl
    - IVL: 20 sec hard coded in library if not defined

- Router numbers to cross for multicast packets
  - CtrlSystem → MulticastHops
    - 5 hard coded in library if not defined

Tango library 8.1.2 - write_attr_hardware()

- A new method in Tango device classes called by kernel
  - Feature Request 68
- For hardware which support writing several parameters (data) in one go
  - Called **after** the write_xxx() method
- Virtual method in DeviceImpl with default implementation doing nothing
  - Don't need it → Don't code it

```
void MyTangoClass::write_attr_hardware(vector<long> &att_idx);
```

Tango library 8.1.2 - write_attr_hardware()

- Pb : Throwing exception!
  - Classical Tango::Except::throw_exception()
    - Error reported to client for all concerned attributes
  - New Tango::Except::throw_named_exception()
    - Error reported to client for one attribute
    - Attribute in error specified by
      - Its name
      - Its index (from the input arg)
  - On client side
    - Idem classical DeviceProxy::write_attributes()

Tango library 8.1.2 - Miscellaneous

- New class DbServerData
  - Methods to easily move a complete DS from one CS to another one
- Memorization of error reported during device server startup sequence for memorized attributes
- New methods in Database class for device/attribute alias management
- Change default attribute format according to attribute data types
  - Bug 570 and FR 22

Tango 8.1 - ZMQ

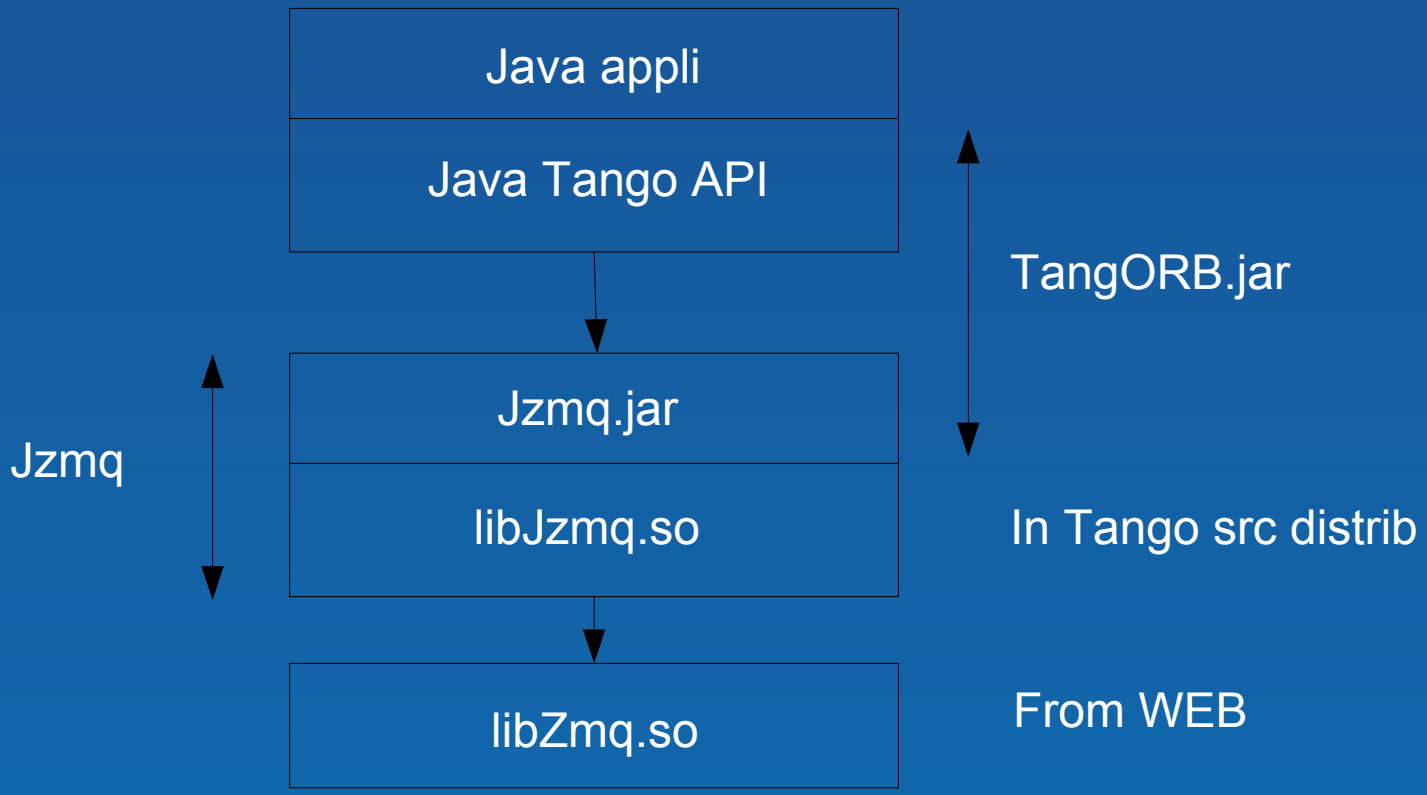
- ZMQ is now at release 3.2.3
  - Much better release than 3.1
  - But wire **INCOMPATIBLE** with 3.1 !!!
- Tango 8.1 checks that both side using event are 3.1 or 3.2
  - No mix allowed
- Tango 8.1 compiled with ZMQ 3.1 can be used with 3.1 or 3.2
- Tango 8.1 compiled with ZMQ 3.2 has to be used with 3.2

Tango 8.1 - ZMQ

- ZMQ is a “fashion” product
  - Selected by CERN
  - Used by PSI
  - Used by Ipython....
- Pieter Hintjens (ZMQ guru) at ESRF the 24/25 of June
  - Discuss about we are using ZMQ today
  - Is it possible to replace CORBA by ZMQ + XX ?
    - Which improvement for which work?
- No news from the Nano project

Tango distribution 8.1

- New TangORB release **including** ZMQ event (client side – No multicast)
  - Using Jzmq (2.1.2) which is a JNI layer above C++ ZMQ



Tango distribution 8.1

- LD_LIBRARY_PATH required for Jzmq to find libJzmq.so
- A message printed on console to inform you which event system (notifd or zmq) will be used for event propagation

- TangORB now uses Jacorb 3.1

- TangORB on Android for native application development
  - Still problems with
    - Jacorb 3.1 and RMI classes
    - Tango Access Control
    - ZMQ events
    - Specify Fully Qualified Device Name (tango://host:port/dom/fam/memb)

Tango distribution 8.1

- New TAC server release with improved device name access specification (sr/v-*/*)

- New database server release
- New ATK release

Available: Mid June

- Generate libJzmq.so

- Added a “make pdf” entry to generate Tango book
  - Requires a new log4tango release (4.0.8)

Tango 8@ESRF

- Tango 8.1.1 (8.1.2 without write_attr_hardware()) in use for the machine control system since 2 months but using ZMQ 3.1
  - Including ZMQ events for Java appli (TangORB 8.2.3)
    - Used for all Java applis receiving events from devices (using Tango 8)
  - For Tango HDB
    - 25 % of events uses ZMQ
  - But still all notifd processes running
- Some beam lines use Tango 8.0.5

Tango 8@ESRF

- Due to ZMQ wire incompatibility, update to ZMQ 3.2 touchy
  - For machine control system
    - Stop all control system
    - Update lib
    - Restart everything
  - For beam lines
    - Care has to be taken for beam lines processes accessing machine control system (Insertion device control, front-end control,...)

Tango 9

- Main added features (reminder):
  - Enumeration as attribute data type
  - Pipes
  - DeviceImpl::write_attr_hardware()
- Not started yet but
  - Point 3 already implemented in Tango 8.1
  - Updated reference doc for C++
    - Tango book chapter 6 and on-line doc about classes/methods available for Tango classes development merged
    - Available from pink site
    - Doxygen generated
  - Jzmq replaced by Jeromq ?

SourceForge - Forum

- Tango-cs and tango-ds SourceForge projects moved to Allura
- As requested by Executive Committee Tango-ds wiki opened
  - Not used!

- Krister (MaxLab) did a study about forum
  - PhpBB or Google group (requires Google account)
  - What about bugs report sent to mailing list?
    - Do we need forum + mailing list?

Tango logo

- Thank's to all those who sent one (11 proposals!)
- The last word is for the EC
- Community choice is proposal 9
  - Was the question well asked?

Tango logo

- 30 (21/05) answers from community members

47 %

20 %

17 %

7 %

Tango survey

- Thank's to all those who took the few necessary minutes to answer
  - 38 answers
- Aims:
  - How Tango is used?
  - Is it possible to remove some unused features?
- First sent to only one contact per institute then to the whole Tango list
- Not easy to conclude (some answers are strange)
- Every single features seems to be used
  - Can't remove anything!!!

Tango survey

- Tango 7 is still the most widely used release (53% - 41%)
- C++ is language of choice to write Tango classes
  - Python just behind
  - Some Java classes
- Inheritance in Tango classes is sometimes used
- Half of the answers uses Tango logging for their own messages
- 30 % of answers never use attribute alarms
  - Even RDS is used!
- Polling is used but not always
  - 22 % don't use it

Tango survey

- Polling threads pool, externally triggered polling and filling polling buffer by code marginally used
  - 1 user for externally triggered polling!
- Serialization (device or attribute) rarely changed
- Memorized attributes widely used (67% + 11%)
- DevEncoded data type also used
- Your own main loop in DS, DS with database in a file or without database marginally used
  - 80 % don't use it but 15-20 % use it sometimes and even 1 user uses its own main loop!

Tango survey

- Python is the winner to write clients (60 %)
  - C++ and Java equally used
- Asynchronous calls, write_read_attribute() AttributeProxy and Group features are used
- attribute_history() and command_history() marginally used
  - Requires polling
- Device locking also marginally used
- Device alias more used than attribute alias

Tango survey

- Events used but not by everybody (8 % don't use them)
  - Change and periodic events are the most appreciated
- ATK and Taurus are the favorite GUI layers
- Matlab then LabView are the main bindings
  - Not very surprising

- Starter DS widely used but not everywhere
- All provided apps used
  - Big success for Jive (96 % use it)
  - Astor, AtkPanel and Pogo used in 60 % of answers
    - Pogo mostly for C++ classes and Pogo 7 (Still some Pogo 6 use)
  - Less use for other apps (LogViewer, Jdraw, AtkMoni, DeviceTree, AtkTuning)

Tango survey

- TangoTest DS well appreciated
- TAC marginally used (80 % don't use it)
  - Only one site (ESRF)
- Half of the user get Tango from the source distribution
  - 8 % windows binary distribution
- Only 20 % of users update their Tango installation at each minor releases

- File with all answers will be sent to the mailing list

Executive committee

- Tango foundation
- Tango logo
- Icalepcs 2013
- Next Tango meeting (ESRF - 03/04 2014)
- Tango collaboration coordinator