TANGO Kernel News

Tango-9 is used at ESRF in production since October 2014

- C++ 9.0.2 to 9.0.7 (9.1.0 soon)
- Java client 9.0.2
- Tango-9 is nearly ready for C++ and Java
- Python is still under development.

Tango-9 new features (already presented at ESRF last Tango meeting)

- Enum attributes
- Forwarded attributes
- Pipes
- Polling thread improvement
- Miscellaneous
  - Dynamic commands
  - New event types
  - New PyTango programming style

Tango-9 official release

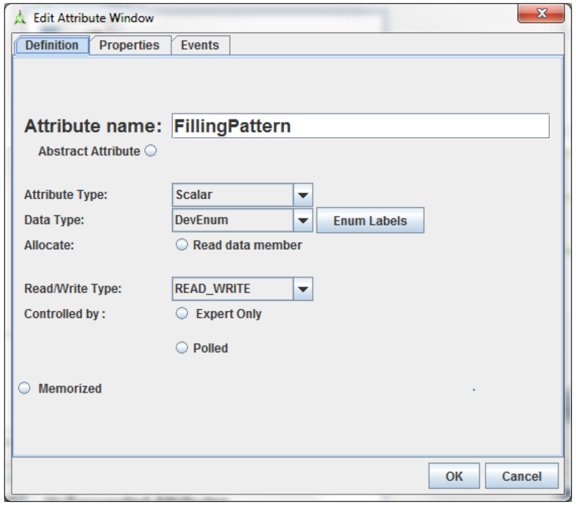
TANGO Kernel News
Enum Attributes

New IDL Tango type: DevEnum (29):

- It is defined as other attributes (name, properties, polling,.....)
- And
  - The Enum definition (***enum _MyEnum { XX, YY, ..}***)
  - Labels for enum fields (***vector<string>***)

TANGO Kernel News
Enum Attributes

Example using Pogo:



Attribute name: FillingPattern

Abstract Attribute ○

Attribute Type: Scalar

Data Type: DevEnum

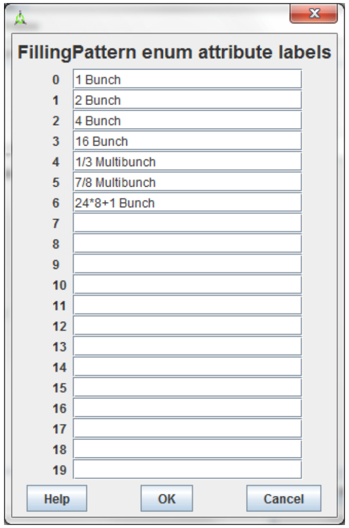
Allocate: ○ Read data member

Read/Write Type: READ_WRITE

Controlled by : ○ Expert Only

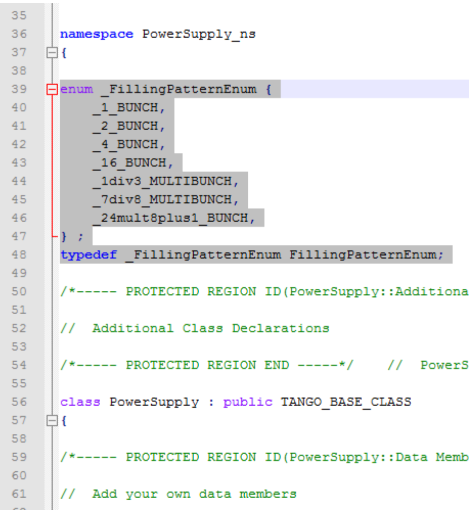
○ Polled

○ Memorized

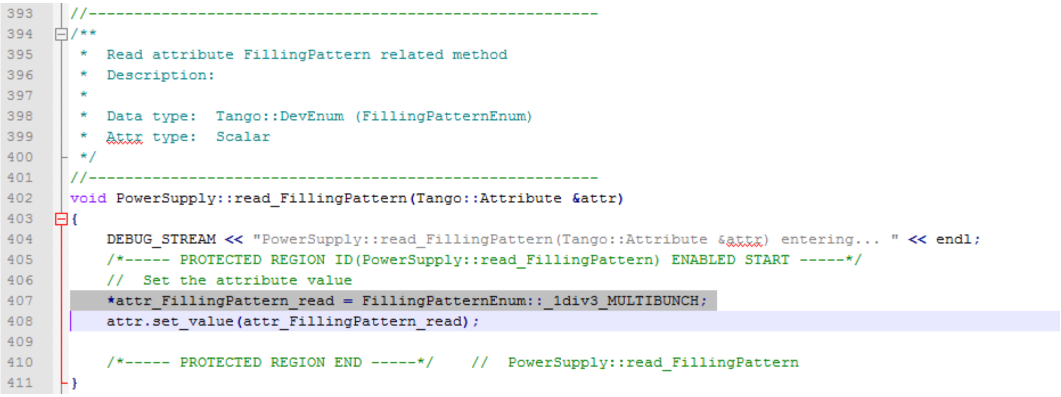


FillingPattern enum attribute labels

0	1 Bunch
1	2 Bunch
2	4 Bunch
3	16 Bunch
4	1/3 Multibunch
5	7/8 Multibunch
6	24*8+1 Bunch
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	

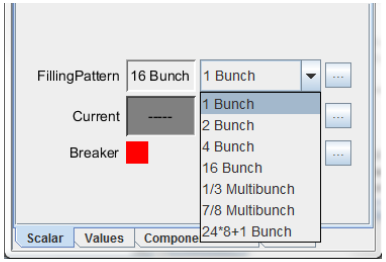


```

namespace PowerSupply_ns
{

enum _FillingPatternEnum {
    _1_BUNCH,
    _2_BUNCH,
    _4_BUNCH,
    _16_BUNCH,
    _1div3_MULTIBUNCH,
    _7div8_MULTIBUNCH,
    _24mult8plus1_BUNCH,
} ;
typedef _FillingPatternEnum FillingPatternEnum;

/*----- PROTECTED REGION ID(PowerSupply::Additiona

//  Additional Class Declarations

/*----- PROTECTED REGION END -----*/    //  PowerS

class PowerSupply : public TANGO_BASE_CLASS
{

/*----- PROTECTED REGION ID(PowerSupply::Data Memb

//  Add your own data members
```



```
//--------------------------------------------------------
/**
 *  Read attribute FillingPattern related method
 *  Description:
 *
 *  Data type:  Tango::DevEnum (FillingPatternEnum)
 *  Attr type:  Scalar
 */
//--------------------------------------------------------
void PowerSupply::read_FillingPattern(Tango::Attribute &attr)
{
    DEBUG_STREAM << "PowerSupply::read_FillingPattern(Tango::Attribute &attr) entering... " << endl;
    /*----- PROTECTED REGION ID(PowerSupply::read_FillingPattern) ENABLED START -----*/
    //  Set the attribute value
    *attr_FillingPattern_read = FillingPatternEnum::_1div3_MULTIBUNCH;
    attr.set_value(attr_FillingPattern_read);

    /*----- PROTECTED REGION END -----*/    //  PowerSupply::read_FillingPattern
}
```

ATKpanel:



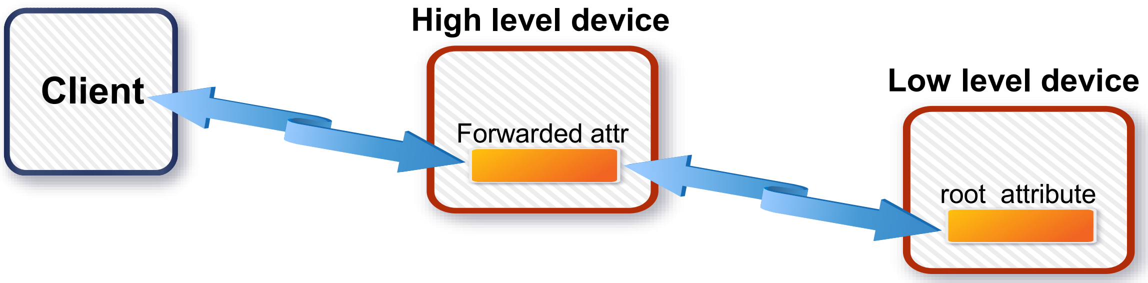
FillingPattern 16 Bunch 1 Bunch

Current -----

Breaker 🟥

TANGO Kernel News
Forwarded Attributes

A forwarded attribute is a device attribute mapping an attribute of a root device.



A forwarded attribute is defined by:

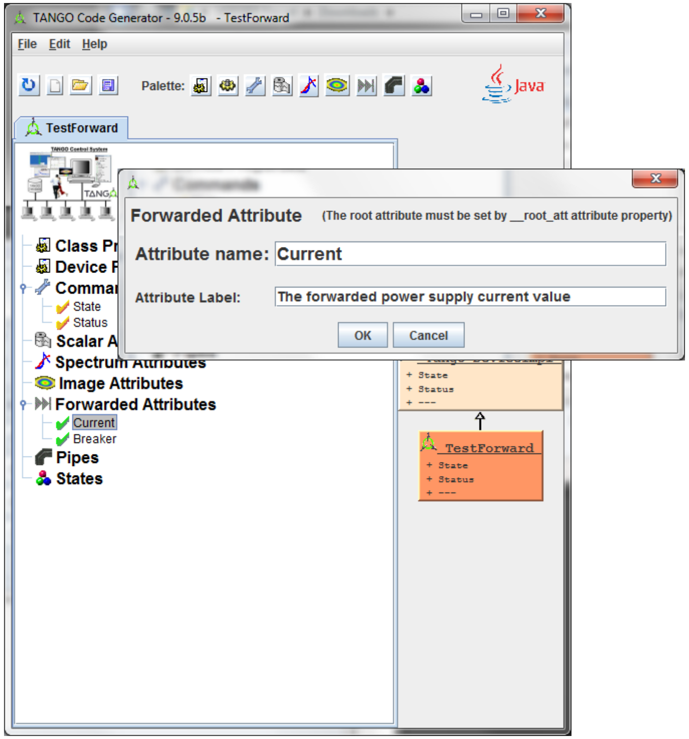
- A name
- A label to overload the root attribute label
- An attribute property ***__root_att*** in database to set root attribute name.

Note:

- If the ***__root_att*** is not defined, the attribute will not be created
  - Error message in Status: ***"Error: Missing root attribute definition"***
- If the root device is not running, the device will in ALARM
  - Error message in Status: ***"Forwarded attribute xxx not reachable"***
- The attribute is really forwarded. No access to the value by code !

TANGO Kernel News
Forwarded Attributes

A forwarded attribute is a device attribute mapping an attribute of a root device.
If you use Pogo, there is no code to write.

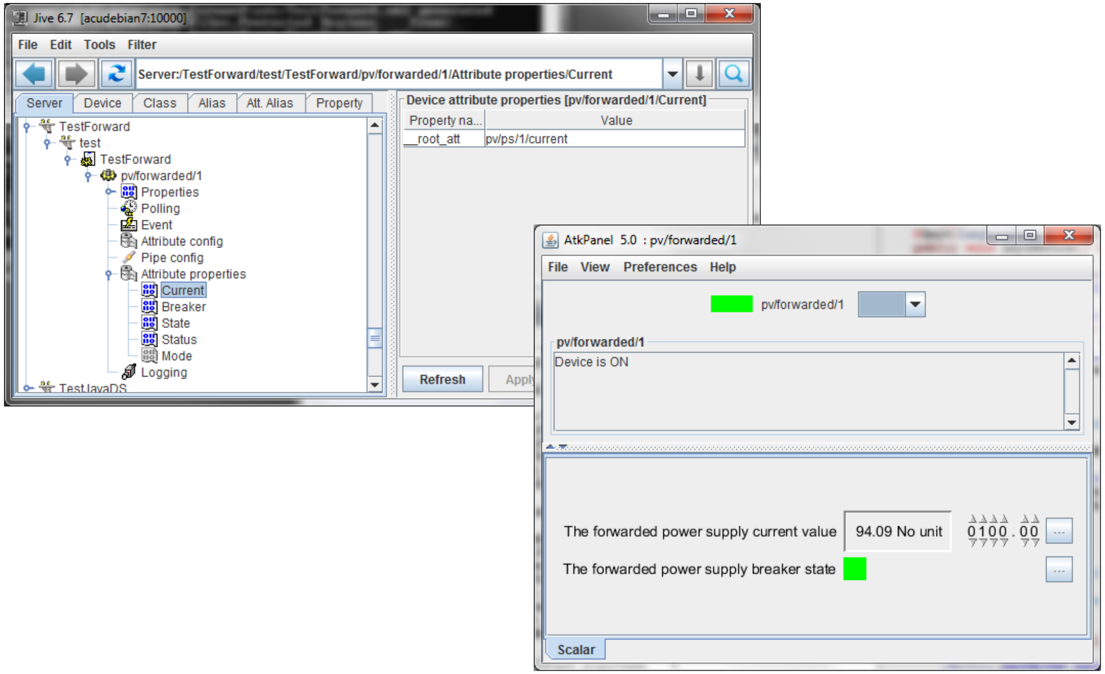


TANGO Code Generator - 9.0.5b - TestForward

Forwarded Attribute (The root attribute must be set by __root_att attribute property)

Attribute name: Current

Attribute Label: The forwarded power supply current value



Jive 6.7 [acudebian7:10000]

Server:/TestForward/test/TestForward/pv/forwarded/1/Attribute properties/Current

Device attribute properties [pv/forwarded/1/Current]

Property na...	Value
__root_att	pv/ps/1/current

AtkPanel 5.0 : pv/forwarded/1

pv/forwarded/1

Device is ON

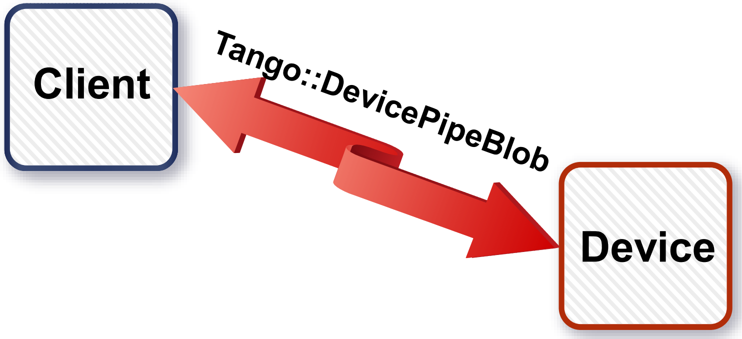
The forwarded power supply current value 94.09 No unit

The forwarded power supply breaker state 🟩

TANGO Kernel News
Pipes

Pipes are a new way to communicate between client and device.
It could be useful to transport in one call, data with different types.

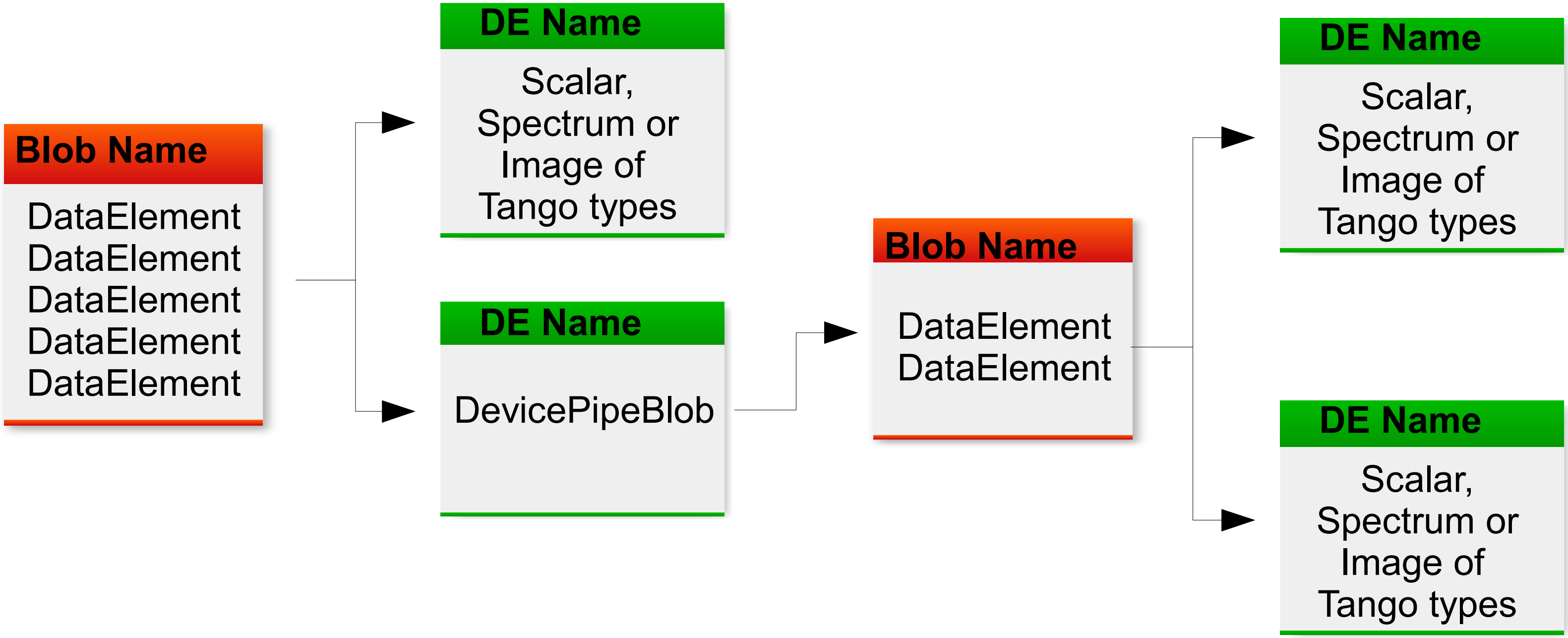
The pipe transports a ***blobs (Tango::DevicePipeBlob).***



TANGO Kernel News
Pipes

The pipe transports a ***blobs (Tango::DevicePipeBlob).***
The transported blob structure could change at each call.

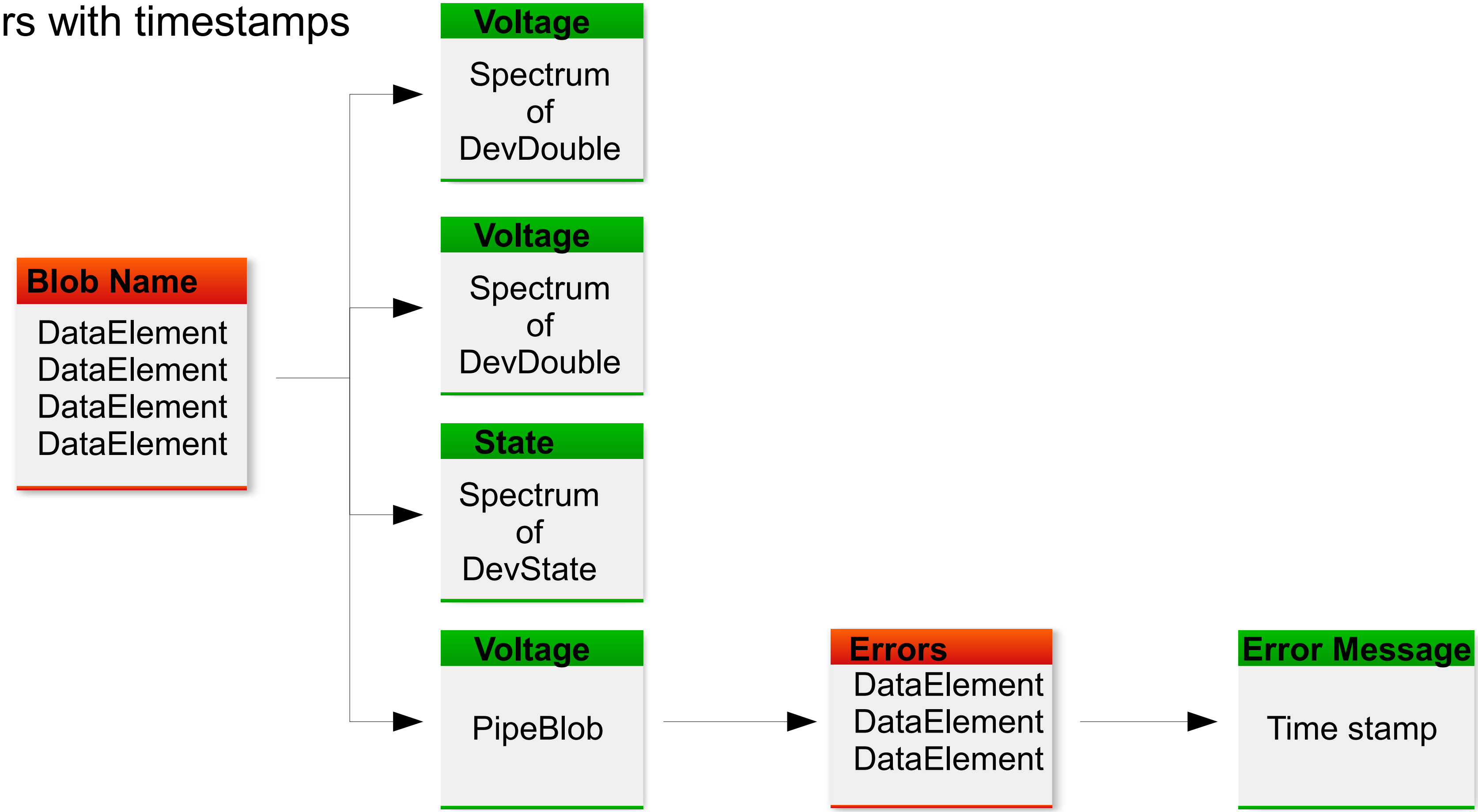
A ***blob*** is defined by a name and a sequence of **Tango::DataElement**
A **DataElement** is defined by a name and a data.
A data could be aTango data type or a ***blob***.



TANGO Kernel News
Pipes

Simple example: create a Pipe to read

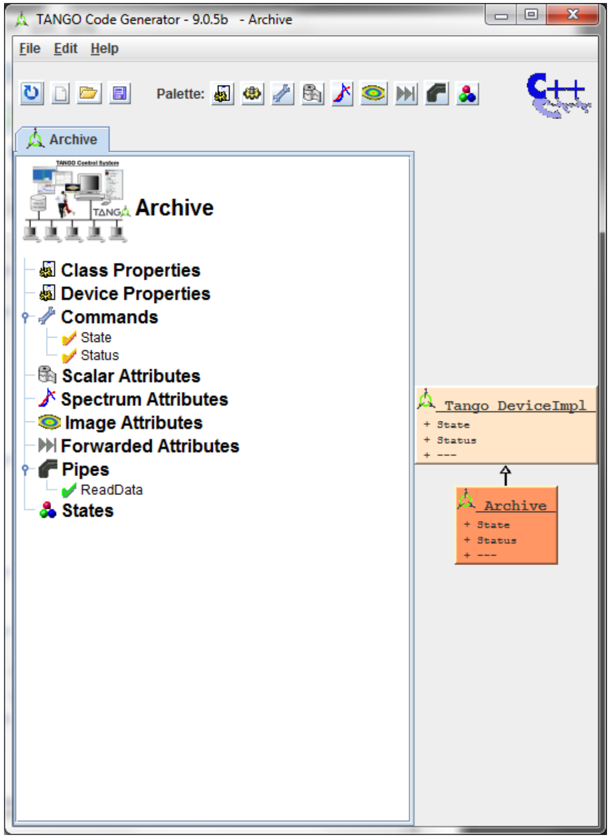
- attribute history values
- errors with timestamps

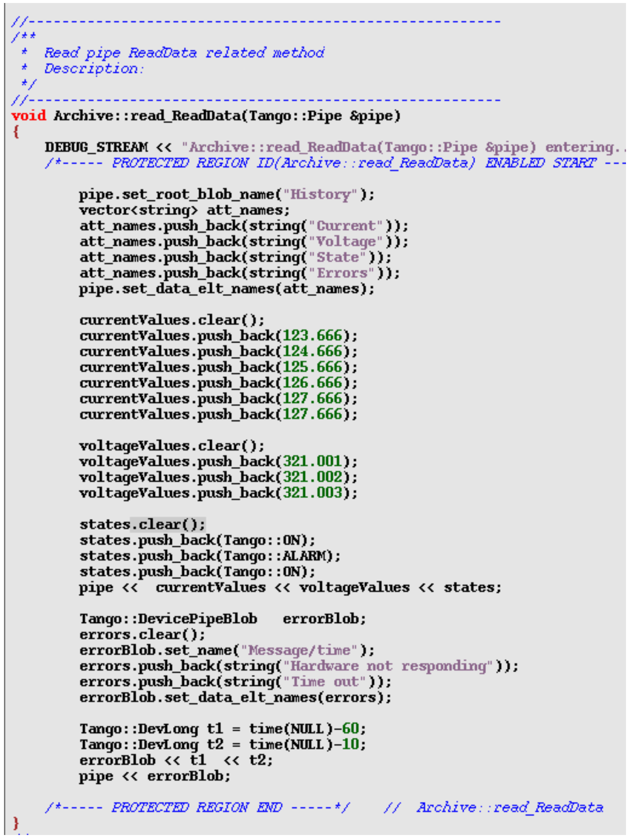


TANGO Kernel News
Pipes

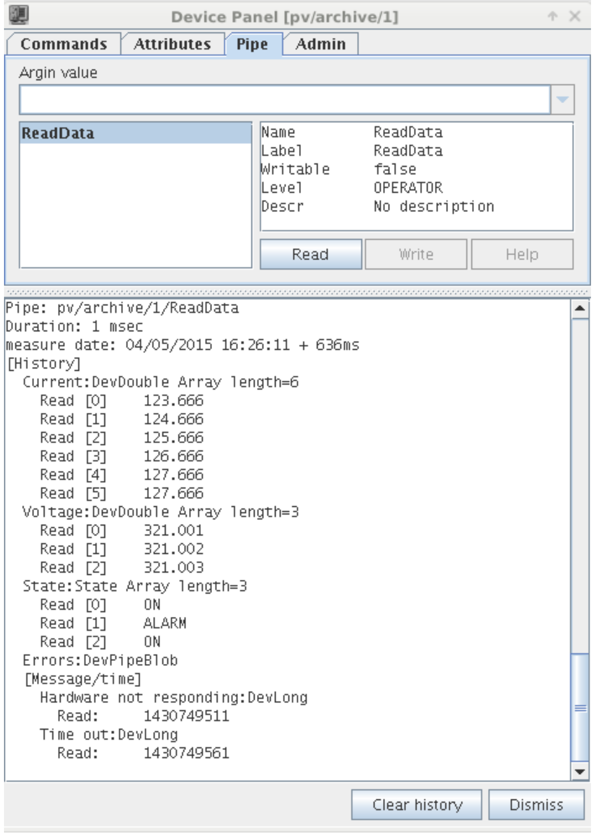
Simple example: create a Pipe to read

- attribute history values
- errors with timestamps





```
//------------------------------------------------------------
/**
 *  Read pipe ReadData related method
 *  Description:
 */
//------------------------------------------------------------
void Archive::read_ReadData(Tango::Pipe &pipe)
{
    DEBUG_STREAM << "Archive::read_ReadData(Tango::Pipe &pipe) entering..
    /*----- PROTECTED REGION ID(Archive::read_ReadData) ENABLED START ---

        pipe.set_root_blob_name("History");
        vector<string> att_names;
        att_names.push_back(string("Current"));
        att_names.push_back(string("Voltage"));
        att_names.push_back(string("State"));
        att_names.push_back(string("Errors"));
        pipe.set_data_elt_names(att_names);

        currentValues.clear();
        currentValues.push_back(123.666);
        currentValues.push_back(124.666);
        currentValues.push_back(125.666);
        currentValues.push_back(126.666);
        currentValues.push_back(127.666);
        currentValues.push_back(127.666);

        voltageValues.clear();
        voltageValues.push_back(321.001);
        voltageValues.push_back(321.002);
        voltageValues.push_back(321.003);

        states.clear();
        states.push_back(Tango::ON);
        states.push_back(Tango::ALARM);
        states.push_back(Tango::ON);
        pipe <<  currentValues << voltageValues << states;

        Tango::DevicePipeBlob   errorBlob;
        errors.clear();
        errorBlob.set_name("Message/time");
        errors.push_back(string("Hardware not responding"));
        errors.push_back(string("Time out"));
        errorBlob.set_data_elt_names(errors);

        Tango::DevLong t1 = time(NULL)-60;
        Tango::DevLong t2 = time(NULL)-10;
        errorBlob << t1  << t2;
        pipe << errorBlob;

    /*----- PROTECTED REGION END -----*/    //  Archive::read_ReadData
}
```



```
Name        ReadData
Label       ReadData
Writable    false
Level       OPERATOR
Descr       No description
```

```
Pipe: pv/archive/1/ReadData
Duration: 1 msec
measure date: 04/05/2015 16:26:11 + 636ms
[History]
  Current:DevDouble Array length=6
    Read [0]    123.666
    Read [1]    124.666
    Read [2]    125.666
    Read [3]    126.666
    Read [4]    127.666
    Read [5]    127.666
  Voltage:DevDouble Array length=3
    Read [0]    321.001
    Read [1]    321.002
    Read [2]    321.003
  State:State Array length=3
    Read [0]    ON
    Read [1]    ALARM
    Read [2]    ON
  Errors:DevPipeBlob
  [Message/time]
    Hardware not responding:DevLong
      Read:     1430749511
    Time out:DevLong
      Read:     1430749561
```

TANGO Kernel News
Polling thread improvement

Not already installed.
A polling thread improvement is under development to avoid the famous error "Polling thread is late" when event system used.

Two ways:

- When attributes belong to same device and are polled at same frequency,
  Replace ***read_attribute*** by ***read_attributes***

- Pro: Improve the efficiency.
- Con: Time spent by each attribute is lost.

- Do not send error if thread is late just once.

TANGO Kernel News
Miscellaneous

Dynamic commands:

- It is now possible to add and remove a dynamic command.
- Programmation and code generated by Pogo is similar to the code for dynamic attributes.

Pipe event: an event is fired when a pipe content is set.

Interface change event: an event is fired when a device interface change.

- Add/Remove dynamically an attribute
- Add/Remove dynamically a command
  - Note: Different to (attribute) config event !

New PyTango programming style:

- Called PythonHL (Python High Level) developped by Tiago on top of PyTango
- Pogo templates written by Sébastien Gara
- Available only for Tango-8

TANGO Kernel News
Official Release

Tango-9 release will use:

- OmniORB-4.2
- ZMQ-4.0.5

Before official release:

- Polling thread improvement validation
- Few bugs to be fixed in Java server API
  - Java API client and server will be merged (JTango.jar)
  - Java tools and ATK will be distributed in Java-7.
- Windows tests
  - VC12 (VS 2013) for development
  - VC9 and VC10 only for libraries (compatibility with PyTango)
- Packaging

Expected release dates:

C++ and Java:	end of June
Python and PythonHL:	end of summer
Python Pogo templates:	few weeks after PyTango release

Thank you for your attention.