

---

# ERAS test doc Documentation

*Release 0.1*

**IMS**

May 02, 2013



# CONTENTS



Contents:



# TANGO SETUP

```
sudo apt-get install python-pytango
```

```
...
```

The following NEW packages will be installed:

```
ipython libblas3 libblas3gf libboost-python1.49.0 libcos4-1  
libgfortran3 liblapack3 liblapack3gf liblog4tango4 libomniorb4-1  
libomnithread3c2 libtango7 python-configobj python-decorator  
python-numpy python-pytango python-simplegeneric tango-common
```

TANGO Host: pcname:10000





# TANGO SERVERS

## 2.1 Hunveyor

Test page

## 2.2 GOES

### 2.2.1 API

```
class goes.Entry(values)
    this is some kind of entry

class goes.PyDevice(name)
    this is some kind of device

    attr_list = {'mev10': [], {'polling period': 20000}}
    cmd_list = {}

class goes.PyDeviceExp(cl, name)
    this is some other kind of device

    get_data()
        this is what you use to get the data

    init_device()
        this is used by Tango

    is_get_data_allowed()
        some method that checks if the data are allowed

    is_mev10_allowed(req_type)
        you can get the mev10 only if you are allowed

    read_mev10(the_att)
        with this you get the mev10!

class goes.PyDeviceExp(cl, name)
    this is some other kind of device

    get_data()
        this is what you use to get the data
```

**init\_device()**  
this is used by Tango

**is\_get\_data\_allowed()**  
some method that checks if the data are allowed

**is\_mev10\_allowed(*req\_type*)**  
you can get the mev10 only if you are allowed

**read\_mev10(*the\_att*)**  
with this you get the mev10!

# TEMPLATES

## 3.1 SWRS Template

### 3.1.1 Change Record

### 3.1.2 Introduction

#### Purpose

Describes the purpose of the document, and the intended audience.

#### Scope

Describes the scope of this requirements specification.

#### Applicable Documents

#### Reference Documents

#### Glossary

See the *Glossary* page for a list of terms, like *IMS* or *ERAS*.

#### Overview

Provides a brief overview of the package defined as a result of the requirements elicitation process.

### 3.1.3 General Description

#### Problem Statement

This section describes the essential problem(s) currently confronted by the user community. In other words, this section should discuss what purpose this software package fulfills.

## Functional Description

Describes the general functionality of the software, which will be discussed in more detail below.

## Environment

Describes the environment in which this software will function.

## User objectives

### User1

Describe all the users and there expectations for this package

## Constraints

Describe any constraints that are placed on this software.

## 3.1.4 Functional Requirements

This section lists the functional requirements in ranked order. Functional requirements describe the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements. Each functional requirement should be specified in a format similar to the following:

### Requirement

#### Description

#### Criticality

- High | Normal | Low

#### Dependency

Indicate if this requirement is dependant on another.

## 3.1.5 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

## User Interfaces

Describes how this product interfaces with the user.

### **GUI (Graphical User Interface)**

Describes the graphical user interface if present. This section should include a set of screen dumps or mockups to illustrate user interface features. If the system is menu-driven, a description of all menus and their components should be provided.

### **CLI (Command Line Interface)**

Describes the command-line interface if present. For each command, a description of all arguments and example values and invocations should be provided.

### **API (Application Programming Interface)**

Describes the application programming interface, if present. For each public interface function, the name, arguments, return values, examples of invocation, and interactions with other functions should be provided. If this package is a library, the functions that the library provides should be described here together with the parameters.

### **Diagnostics**

Describes how to obtain debugging information or other diagnostic data.

### **Hardware Interfaces**

A high level description (from a software point of view) of the hardware interface if one exists. This section can refer to an ICD (Interface Control Document) that will contain the detail description of this interface.

### **Software Interfaces**

A high level description (from a software point of view) of the software interface if one exists. This section can refer to an ICD (Interface Control Document) that will contain the detail description of this interface.

### **Communication Interfaces**

Describe any communication interfaces that will be required.

## **3.1.6 Performance Requirements**

Specifies speed and memory requirements.

## **3.1.7 Development and Test Factors**

### **Standards Compliance**

Mention to what standards this software must adhere to.

### **Hardware Limitations**

Describe any hardware limitations if any exist.

### **Software validation and verification**

Give a detail requirements plan for the how the software will be tested and verified.

### **Planning**

Describe the planning of the whole process mentioning major milestones and deliverables at these milestones.

### **3.1.8 Use-Case Models**

If UML Use-Case notation is used in capturing the requirements, these models can be inserted and described in this section. Also providing references in paragraphs 5, 6 and 7 where applicable.

### **3.1.9 Notes**

### **3.1.10 Appendix A**

# GLOSSARY

**ERAS** European Mars Analog Station

**IMS** Italian Mars Society





# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*



# PYTHON MODULE INDEX

**g**

goes, ??