An Open Source Software Suite for Air and Ocean Vehicles

NTNU 2014-11-17

Ricardo Martins <rasm@lsts.pt>
Paulo Dias <pdias@lsts.pt>

Underwater Systems and Technology Laboratory (LSTS)

Tutorial Overview

- **Background**
- **GLUED**
  - Minimal GNU/Linux distribution
- **IMC**
  - Inter-Module Communication API
- **DUNE**
  - On-board Software
- **Neptus**
  - Command and Control Unit

Background

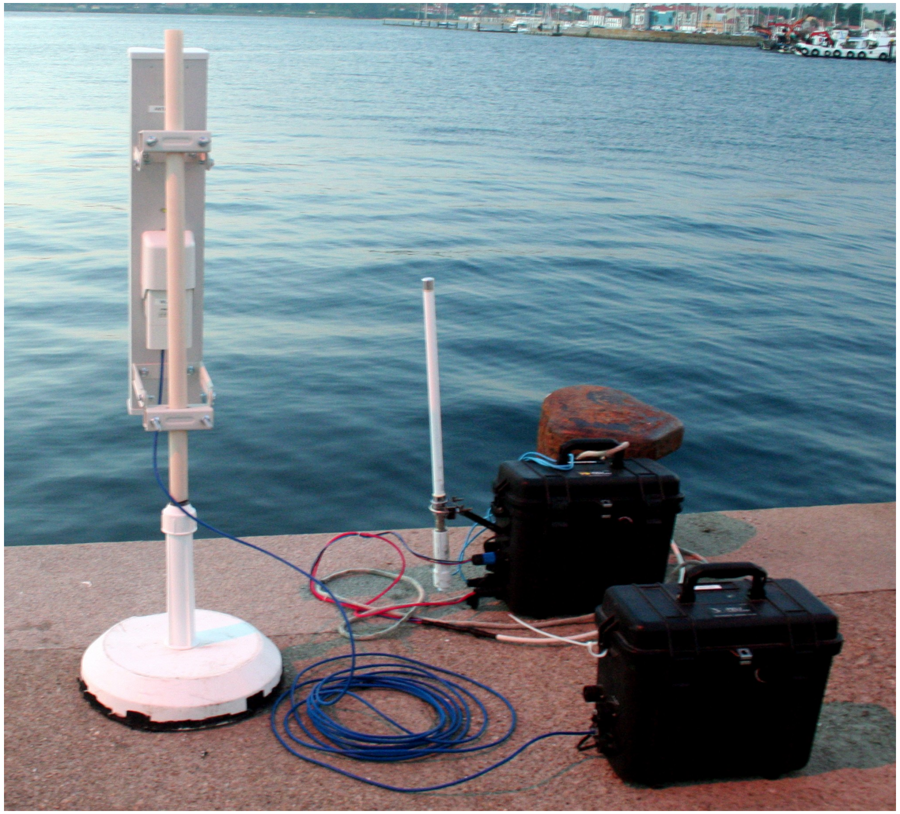
What is It ?

- Complete software solution for autonomous vehicles
- Operating system (Linux distribution)
- API for interaction between software modules
- On-board software for sensor interaction, Control, Guidance, Navigation
- GUI Command & Control Unit
- Mission Review and Analysis (Log Analysis)

Brief Timeline

- 1997 – LSTS was created
- 1997 – First AUV was purchased (WHOI REMUS AUV)
- 2004 – Neptus was created
- 2005 – First ROV built from scratch
- 2006 – DUNE and IMC were created
- 2006 – First AUV built from scratch (LAUV)
- 2007 – Projects and MoU with the PO Ministry of Defence
- ...
- 2014 – 7 operational LAUVs, 1 ROV, 1 ASV, 20 UAVs, 10 LAUVs sold, toolchain used by external entities (NTNU, HTWG, EvoLogics GmbH, ...)

Systems



Why start from scratch ?

- The REMUS on-board software was starting to show its age (based on QNX4 and Pre ISO C++ 98)
- The code-base was developed and maintained poorly
- Freely available software had a few shortcomings:
  - Highly experimental
  - Cumbersome or impossible to use in embedded systems
  - Command & Control software was primitive
  - We disagreed with the architecture/design choices

Further motivation

To avoid things like this:

// Created by John Doe (1996)

// Updated by John Smith (1999)

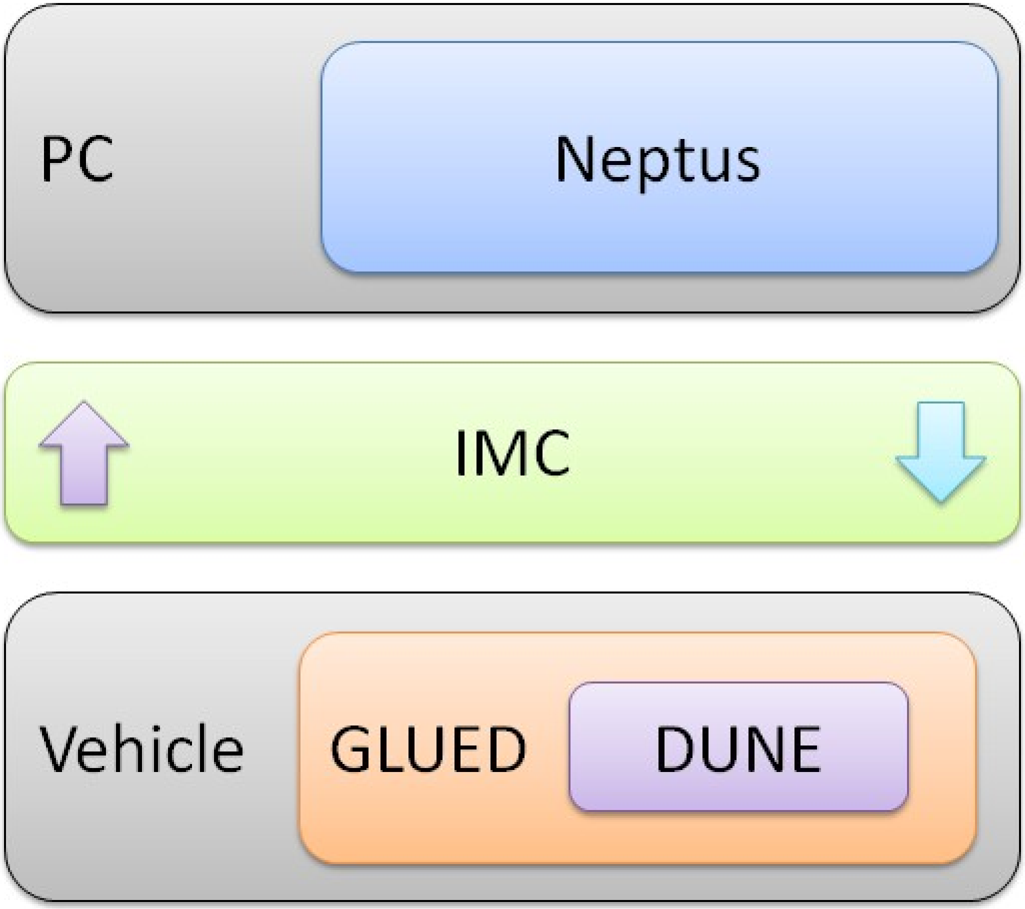
// Updated by Tommy Toe (2001)

// Vandalized by Joe Bloggs (2005)

From *common/abstraction/shared/whoi_mboard.h*

(Fictitious names)

Toolchain Overview



GLUED

GNU/Linux Uniform Environment Distribution

https://github.com/LSTS/glued

Overview

- Minimal GNU/Linux distribution focused on embedded systems
- Small footprint
  - around 10 MiB
- Fast boot time
  - 2 to 5 seconds depending on target machine and peripherals
- Target machine binaries are cross compiled (i.e., built for a platform other than the one on which the compiler is running)
- Creates a reproducible root filesystem for a given target
- Supports several x86, ARM, and MIPS targets

Motivation

- Time to build large software projects in embedded systems is almost unbearable
  - several hours vs a few minutes in modern PCs
- Embedded systems usually require bootloader and kernel customization
- The longer the system takes to boot the longer it is uncontrollable
- Upgrading the operating system should be an unattended process with a predictable outcome
- The root filesystem and target binaries should be easily replicated and traceable

Supported Hardware

- ARM
  - BeagleBone White & Black (TI AM3359 @ 1 GHz)
  - ISEE IGEPv2 (TI DM3730 @ 1 GHz)
- x86
  - IEI PM-LX 800 (AMD Geode LX @ 500 MHz)
  - IEI PM-LX2 800 (AMD Geode LX @ 500 MHz)
  - Kontron pITX (Intel Atom Z510 @ 1.6 GHz)
- MIPS
  - Ubiquiti RouterStation (Atheros AR7161 MIPS 24K @ 680 MHz)

IMC: Inter-Module Communication API

Overview

- Message Oriented Protocol
- One XML document defines all messages
- Generators for documentation, C++ and Java code
- Serialization/deserialization to/from:
  - JSON
  - XML
  - **Binary**
- Serialized messages are used for logging and communication
- Binary serialization format can be translated to human-readable format (LLF)

Interaction Layers

- Plan control
- Vehicle control
- Maneuvering
- Guidance
- Navigation
- Sensing
- Actuation
- Networking
- Storage

Addressing

- Addresses are partitioned in classes (AUV, UAV, ROV, CCU, etc)
- Each system has a unique address (i.e., unique number)
- Subsystems/submodules of a system are called ***entities***
- Each entity has a unique local number used to further qualify a message (e.g., disambiguate messages of the same type but different sources, temperature from a CTD vs CPU temperature)

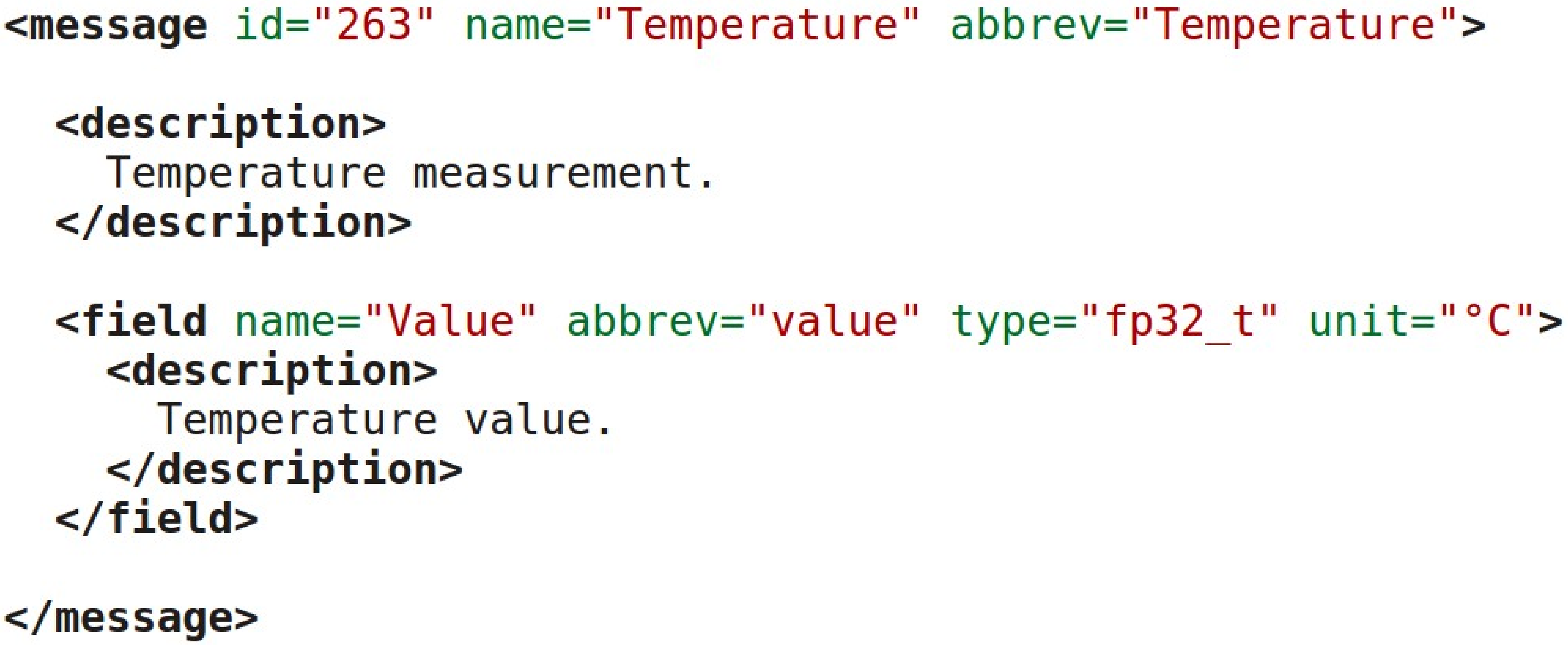
Anatomy of a message

- Synchronization Number
  - Marks the beginning of a message
  - Identifies protocol version
  - Allows for endianess detection
- Message Identification Number
  - Uniquely identifies a message type
- Message size
- Timestamp

Anatomy of a message

- Source Address
- Source Entity
- Destination Address
- Destination Entity
- *Message Specific Fields*
- CRC16

Example



```
<message id="263" name="Temperature" abbrev="Temperature">

  <description>
    Temperature measurement.
  </description>

  <field name="Value" abbrev="value" type="fp32_t" unit="°C">
    <description>
      Temperature value.
    </description>
  </field>

</message>
```

DUNE: Uniform Navigational Environment

Overview

- Designed for embedded systems
- Written in C++
- Used in AUVs, UAVs, ROVs, ASVs, data-loggers and communication gateways
- Related logical operations are isolated from each other in tasks, usually running in a separate thread of execution
- Communication between tasks and communication with external software is performed exclusively by using the set of messages described in the IMC API

Overview

- Communication
  - TCP, UDP, Acoustic modem, Iridium, GSM
- Logging
- Interaction with sensors, actuators, and power devices
- Controllers for attitude, speed, manual operation, etc
- Guidance algorithms
- Maneuvers (way-point following, area coverage, follow reference, loiter, station keeping, etc)

Supported Platforms

- Architectures
  - x86, ARM, PowerPC, SPARC, MIPS, AVR32
- Operating Systems
  - Linux v2.6+/Android, QNX v6.x, Oracle Solaris, Mac OS X, eCos, RTEMS, OpenBSD, FreeBSD, NetBSD, Microsoft Windows
- Hardware Interfaces
  - Serial Port, I^{2}C, I/O port, CAN

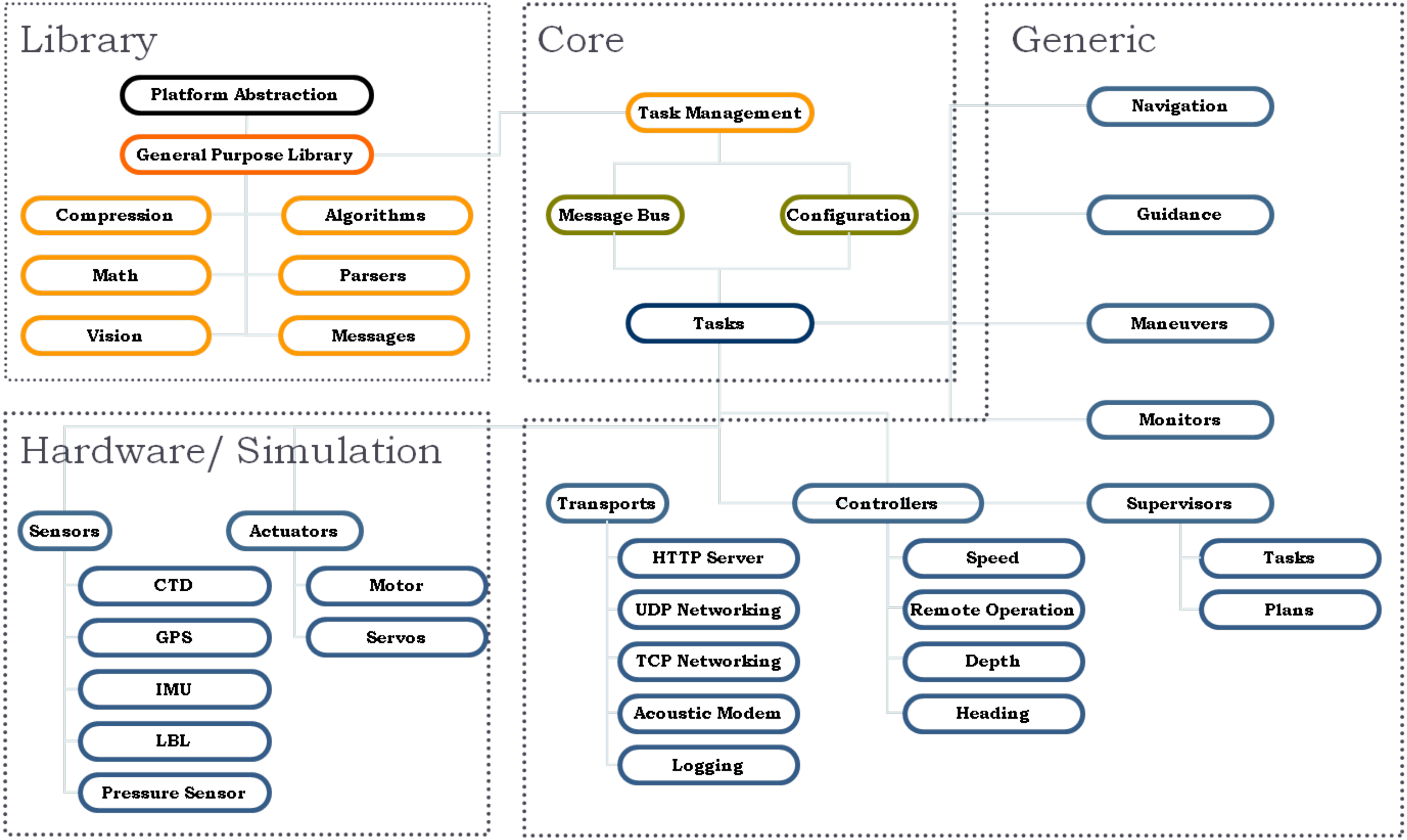
Required Software

- **Mandatory**
  - Git
  - CMake
  - C/C++ Compiler
  - Python Interpreter
- **Optional**
  - Eclipse
  - Microsoft Visual Studio

Required Software

- **Ubuntu/Debian**
  - sudo apt-get install cmake git g++ make python
- **Microsoft Windows**
  - http://www.cmake.org/download/
  - http://git-scm.com/downloads/
  - http://sf.net/projects/mingw/files/Installer/mingw-get-inst/
  - http://www.microsoft.com/express
- **Apple Mac OS X**
  - http://www.cmake.org/download/
  - https://developer.apple.com/xcode/

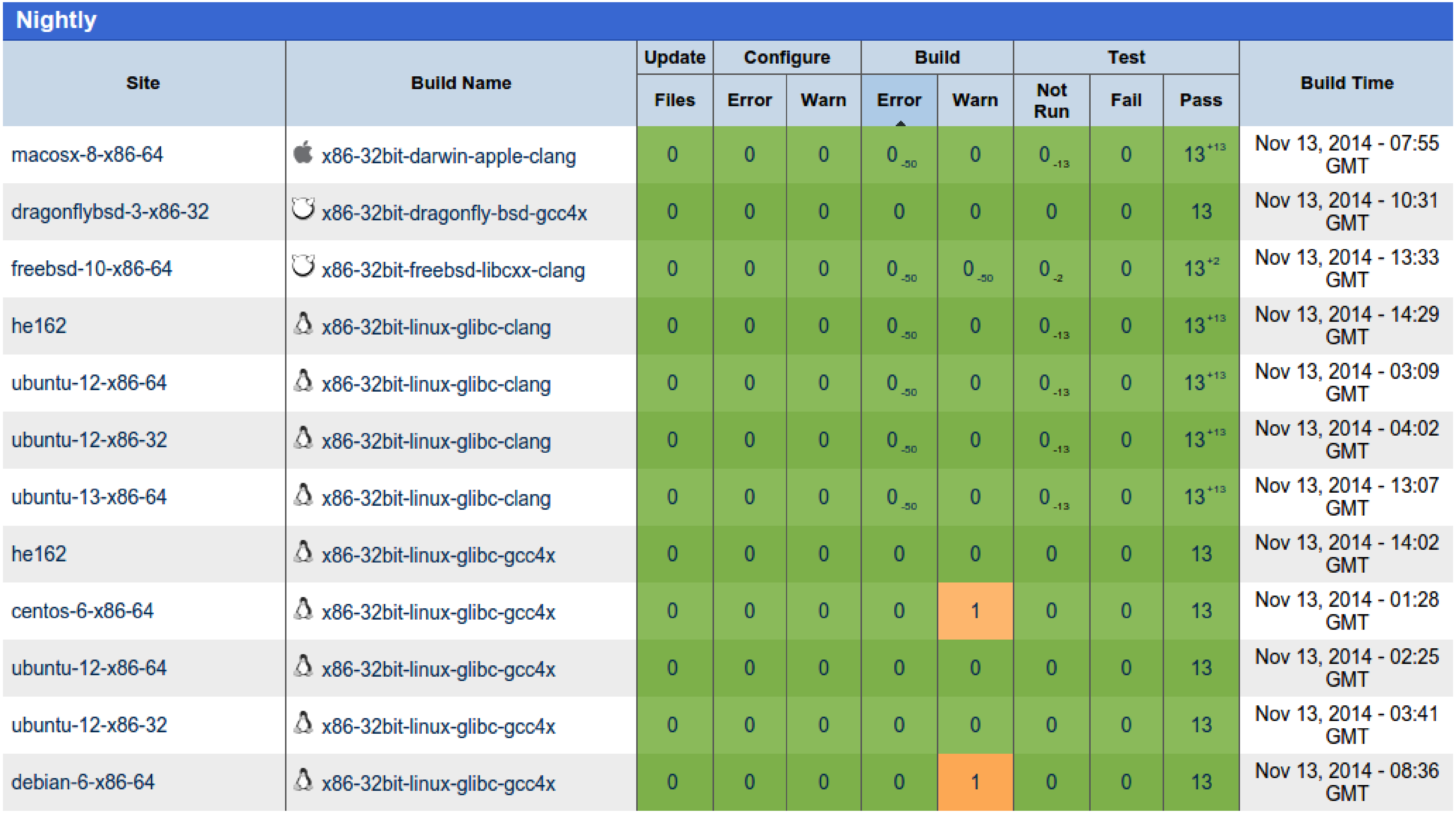
Example System



Resources

- **Source Code**
  - https://github.com/LSTS/dune
- **Documentation**
  - http://lsts.pt/docs
  - https://github.com/LSTS/dune/wiki
- **Mailing List**
  - https://groups.google.com/forum/#!forum/lsts-toolchain
  - lsts-toolchain@googlegroups.com
- **Nightly Builds**
  - http://www.lsts.pt/cdash/index.php?project=DUNE

Nightly Builds



Nightly

Site	Build Name	Update	Configure		Build		Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass	
macosx-8-x86-64	x86-32bit-darwin-apple-clang	0	0	0	0$_{-50}$	0	0$_{-13}$	0	13^{+13}	Nov 13, 2014 - 07:55 GMT
dragonflybsd-3-x86-32	x86-32bit-dragonfly-bsd-gcc4x	0	0	0	0	0	0	0	13	Nov 13, 2014 - 10:31 GMT
freebsd-10-x86-64	x86-32bit-freebsd-libcxx-clang	0	0	0	0$_{-50}$	0$_{-50}$	0$_{-2}$	0	13^{+2}	Nov 13, 2014 - 13:33 GMT
he162	x86-32bit-linux-glibc-clang	0	0	0	0$_{-50}$	0	0$_{-13}$	0	13^{+13}	Nov 13, 2014 - 14:29 GMT
ubuntu-12-x86-64	x86-32bit-linux-glibc-clang	0	0	0	0$_{-50}$	0	0$_{-13}$	0	13^{+13}	Nov 13, 2014 - 03:09 GMT
ubuntu-12-x86-32	x86-32bit-linux-glibc-clang	0	0	0	0$_{-50}$	0	0$_{-13}$	0	13^{+13}	Nov 13, 2014 - 04:02 GMT
ubuntu-13-x86-64	x86-32bit-linux-glibc-clang	0	0	0	0$_{-50}$	0	0$_{-13}$	0	13^{+13}	Nov 13, 2014 - 13:07 GMT
he162	x86-32bit-linux-glibc-gcc4x	0	0	0	0	0	0	0	13	Nov 13, 2014 - 14:02 GMT
centos-6-x86-64	x86-32bit-linux-glibc-gcc4x	0	0	0	0	1	0	0	13	Nov 13, 2014 - 01:28 GMT
ubuntu-12-x86-64	x86-32bit-linux-glibc-gcc4x	0	0	0	0	0	0	0	13	Nov 13, 2014 - 02:25 GMT
ubuntu-12-x86-32	x86-32bit-linux-glibc-gcc4x	0	0	0	0	0	0	0	13	Nov 13, 2014 - 03:41 GMT
debian-6-x86-64	x86-32bit-linux-glibc-gcc4x	0	0	0	0	1	0	0	13	Nov 13, 2014 - 08:36 GMT

Source Code Organization

- **cmake**
  - CMake related files
- **vendor**
  - 3rd party libraries
- **firmware**
  - Microcontroller firmware
- **www**
  - HTTP server files

Source Code Organization

- **programs**
  - Standalone programs, utilities and scripts
- **src/Main**
  - Daemon/launcher main functions (executable entry point)
- **src/DUNE**
  - Core library
- **src/Actuators**
  - Device driver tasks for actuator or actuator-like devices

Source Code Organization

- **src/Maneuver**
  - Maneuvering related tasks. Waypoint following and more complex compound maneuvers
- **src/Monitors**
  - Safety monitors (CPU, Clock, Fuel, Operational Limits, etc)
- **src/Navigation**
  - Position estimators, dead reckoning, etc
- **src/Plan**
  - Plan execution and storage

Source Code Organization

- **src/Power**
  - Device driver tasks for power supplies and related devices
- **src/Sensors**
  - Device driver tasks for sensors (IMUs, Sonars, GPS, ADCs)
- **src/Simulators**
  - Simulation engines and simulation based tasks
- **src/Supervisors**
  - Tasks responsible for supervising global states
- **src/Transports**
  - Communication and logging tasks (UDP, TCP, HTTP, GSM, etc)

Source Code Organization

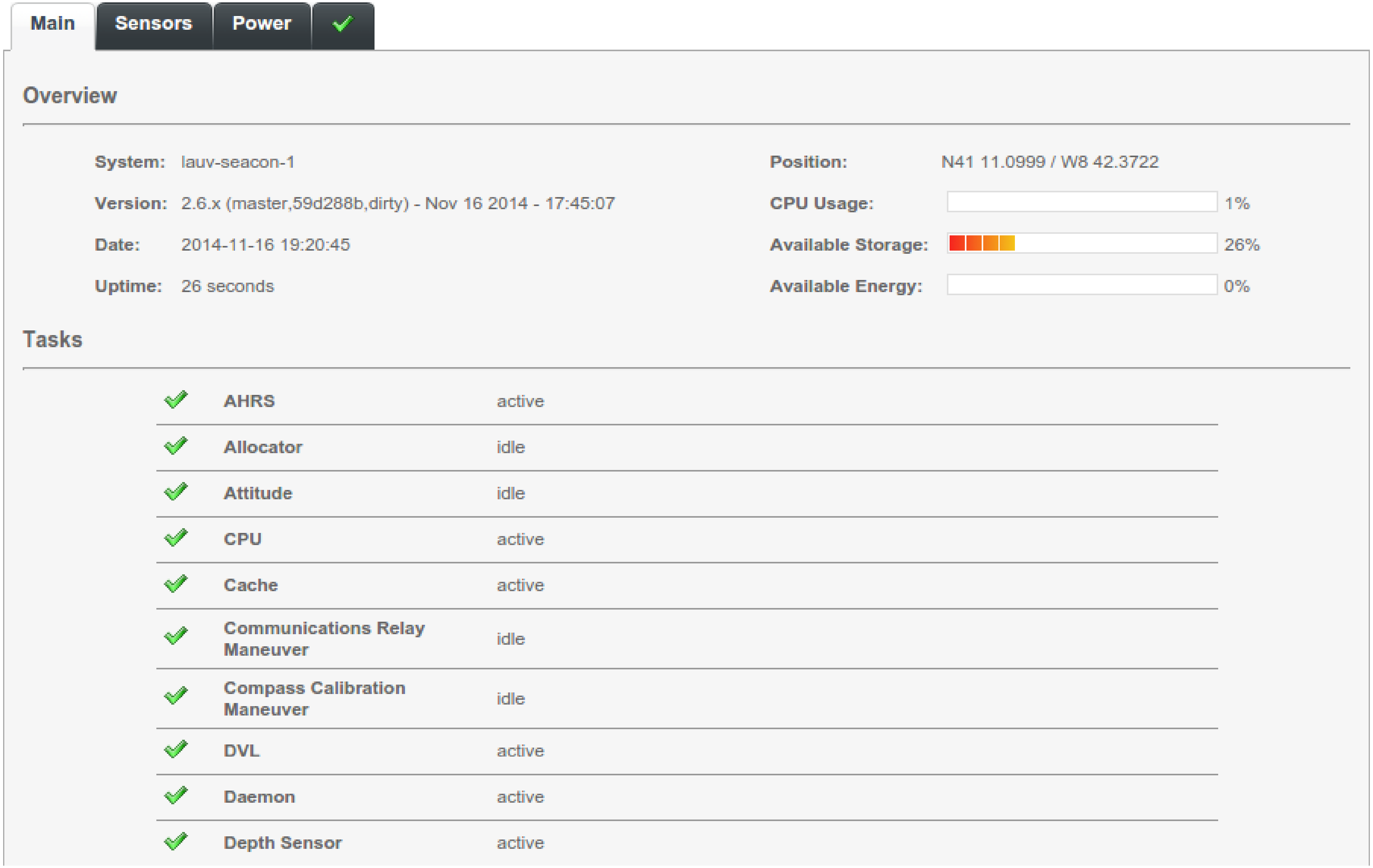
- **src/UserInterfaces**
  - Tasks that control LEDs, LCDs, buttons and instrument panels
- **src/Vision**
  - Video acquisition and processing
- **etc**
  - Configuration files

Bootstrapping

```
mkdir $HOME/tutorial && cd $HOME/tutorial
git clone https://github.com/LSTS/dune.git dune
mkdir build && cd build
cmake ../dune
make
./dune -c lauv-seacon-1 -p Simulation
```

Web Interface

- http://127.0.0.1:8080



Overview

System: lauv-seacon-1

Version: 2.6.x (master,59d288b,dirty) - Nov 16 2014 - 17:45:07

Date: 2014-11-16 19:20:45

Uptime: 26 seconds

Position: N41 11.0999 / W8 42.3722

CPU Usage: 1%

Available Storage: 26%

Available Energy: 0%

Tasks

✓	AHRS	active
✓	Allocator	idle
✓	Attitude	idle
✓	CPU	active
✓	Cache	active
✓	Communications Relay Maneuver	idle
✓	Compass Calibration Maneuver	idle
✓	DVL	active
✓	Daemon	active
✓	Depth Sensor	active

Anatomy of a Task

- Runs concurrently with other tasks
- Communicates with other tasks using IMC messages
- Does one job (and does it right)
- Can be event-driven or periodic

Basic Functions

- Task(const std::string& name, Tasks::Context& ctx)
  - Task constructor
  - Never fails, doesn't throw exceptions
  - Declares configuration parameters
  - Allocates resources that do not depend on configuration parameters
- void onUpdateParameters(void)
  - Called when configuration parameters change
- void onEntityReservation(void)
  - Called when the task can reserve entities

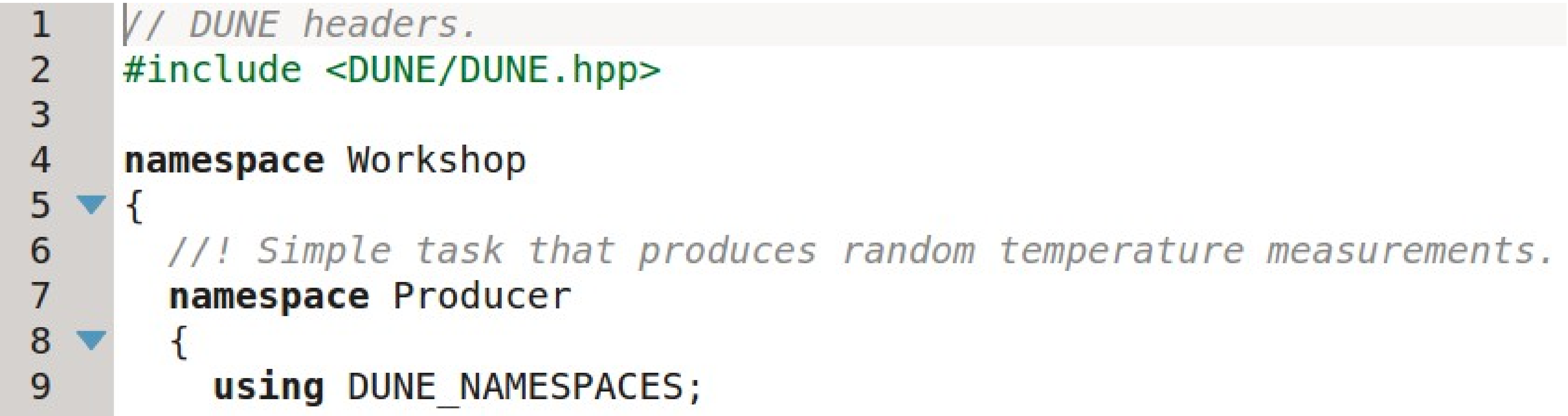
Basic Functions

- void onEntityResolution(void)
  - Called when the task can resolve entities
- void onResourceAcquisition(void)
  - Called when the task can acquire resources (open serial ports, sockets, etc)
- void onResourceInitialization(void)
  - Called when the task can initialize previously acquired resources
- void onResourceRelease(void)
  - Releases all acquired resources
- void onMain(void) / void task(void)
  - Main task loop

Example Task: Producer

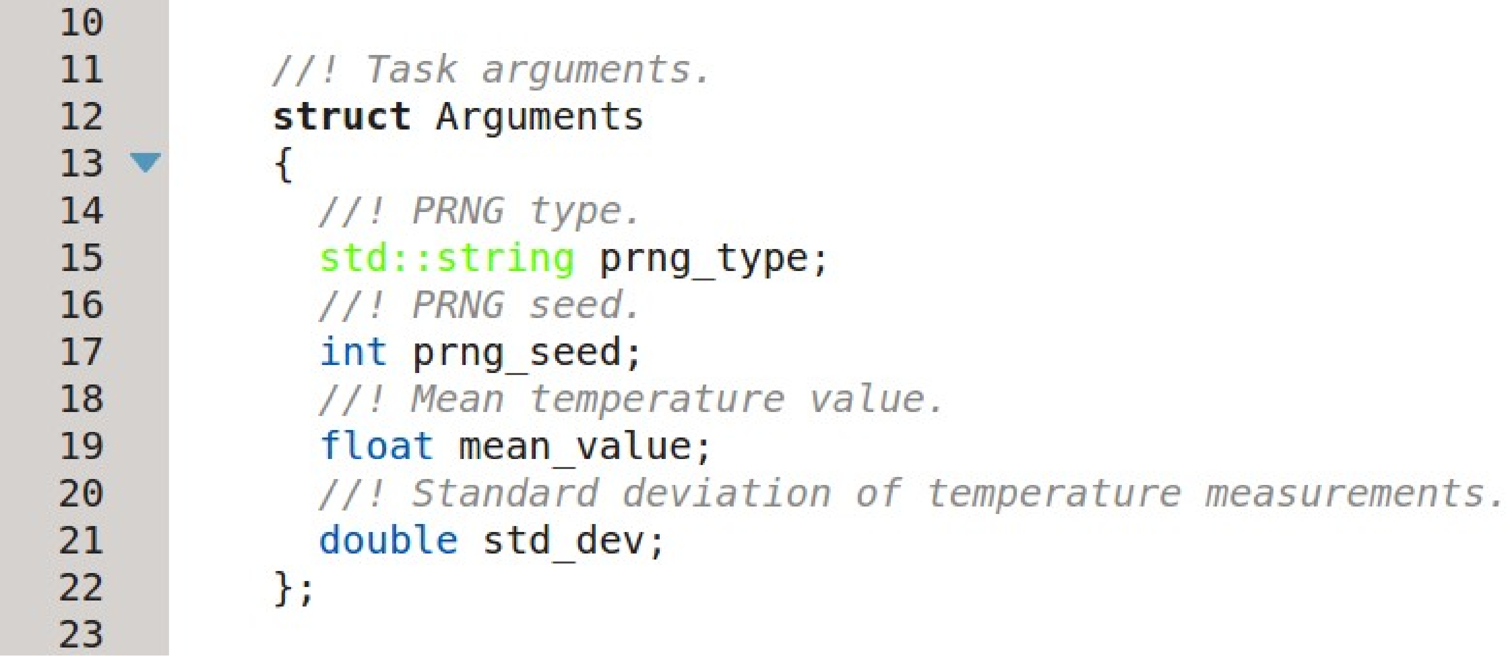
- http://goo.gl/FUezwX
- Task produces random temperature values and dispatches them to the message bus
- Scaffold created using the command:
  - python ../dune/programs/scripts/dune-create-task.py ../dune 'Ricardo Martins' 'Workshop/Producer'
  - make rebuild_cache
- Source code resides in ***src/Workshop/Producer***
- Task entry point is ***src/Workshop/Producer/Task.cpp***
- Build: make

Example Task: Producer



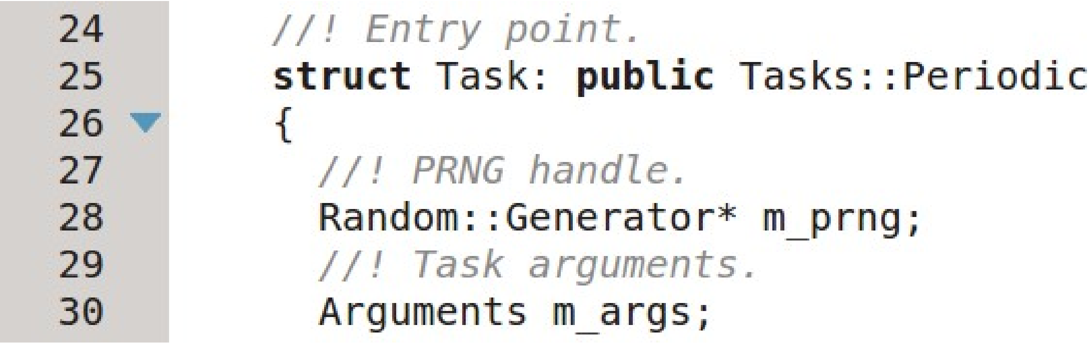
```
// DUNE headers.
#include <DUNE/DUNE.hpp>

namespace Workshop
{
  //! Simple task that produces random temperature measurements.
  namespace Producer
  {
    using DUNE_NAMESPACES;
```

Example Task: Producer



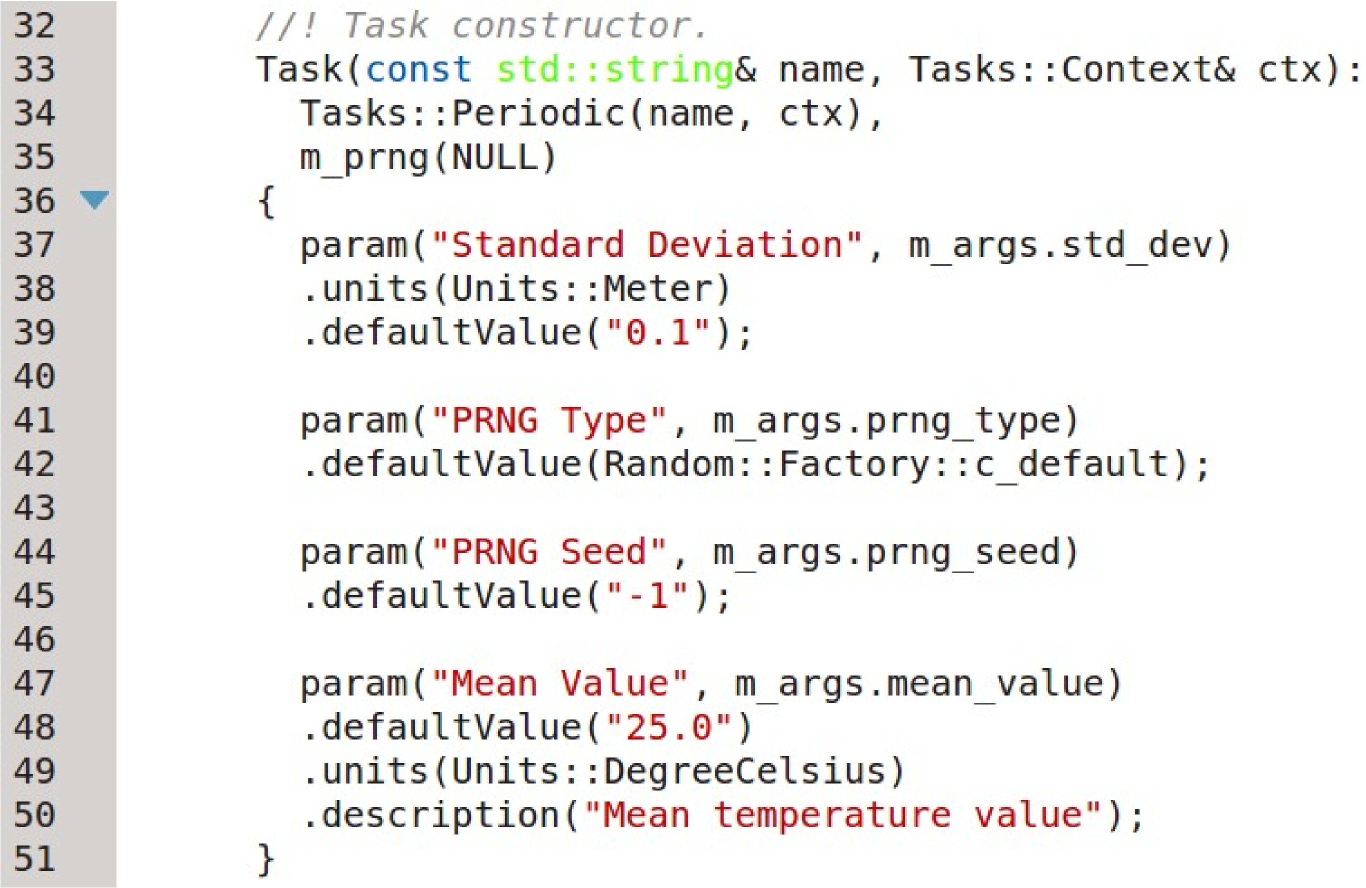
```

//! Task arguments.
struct Arguments
{
  //! PRNG type.
  std::string prng_type;
  //! PRNG seed.
  int prng_seed;
  //! Mean temperature value.
  float mean_value;
  //! Standard deviation of temperature measurements.
  double std_dev;
};

```

Example Task: Producer



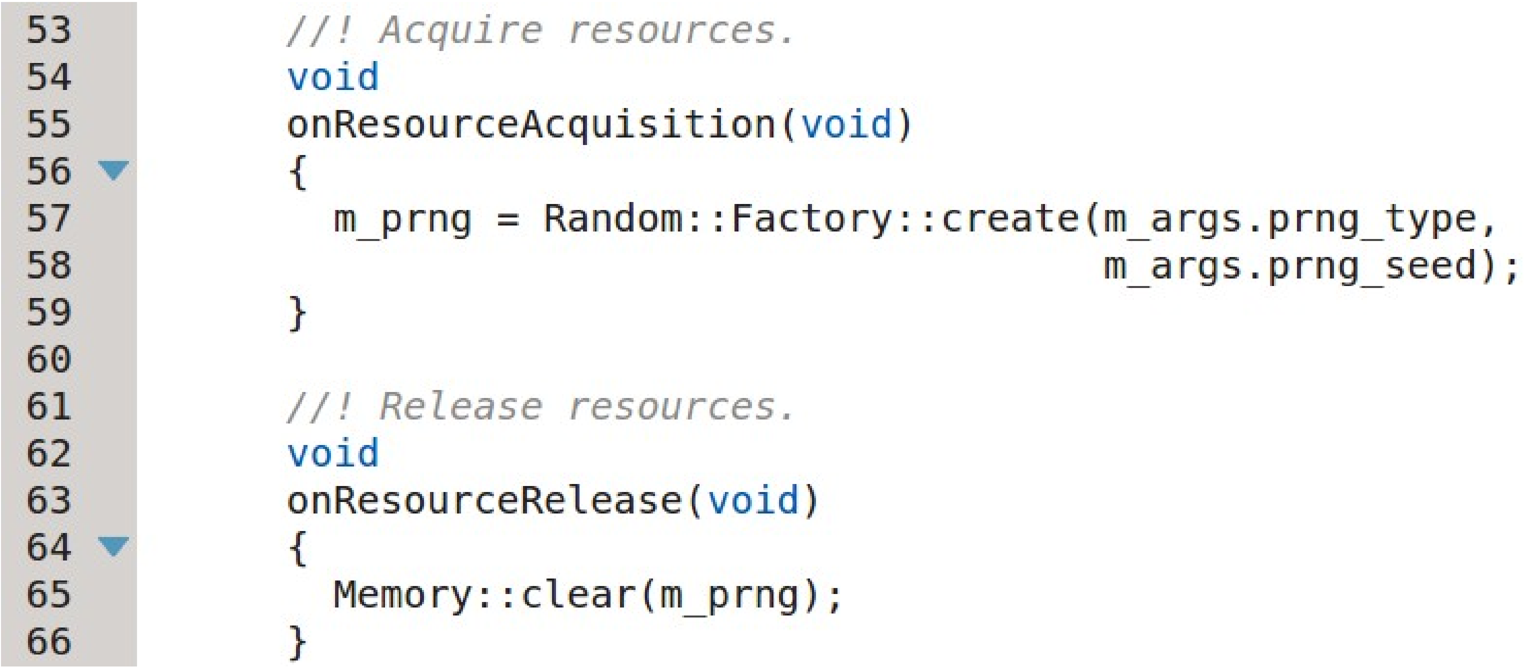
```
//! Entry point.
struct Task: public Tasks::Periodic
{
  //! PRNG handle.
  Random::Generator* m_prng;
  //! Task arguments.
  Arguments m_args;
```

Example Task: Producer



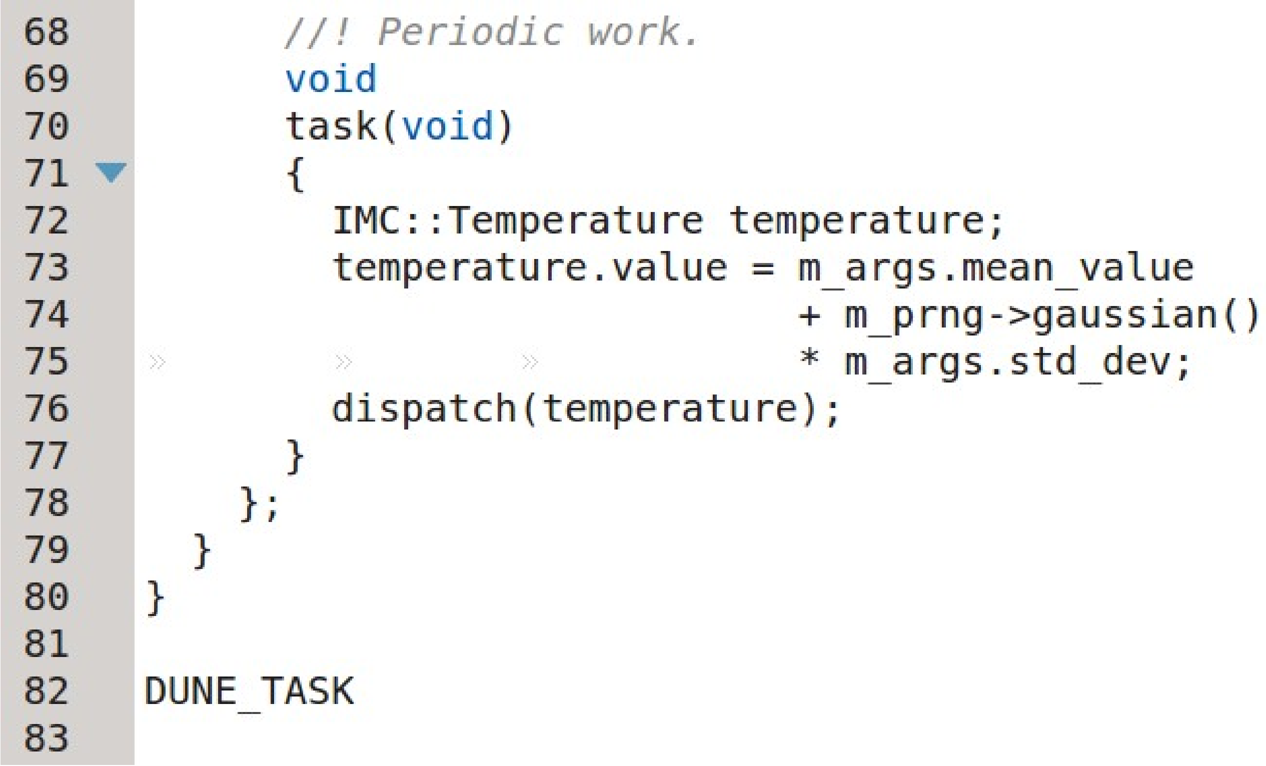
```
//! Task constructor.
Task(const std::string& name, Tasks::Context& ctx):
  Tasks::Periodic(name, ctx),
  m_prng(NULL)
{
  param("Standard Deviation", m_args.std_dev)
  .units(Units::Meter)
  .defaultValue("0.1");

  param("PRNG Type", m_args.prng_type)
  .defaultValue(Random::Factory::c_default);

  param("PRNG Seed", m_args.prng_seed)
  .defaultValue("-1");

  param("Mean Value", m_args.mean_value)
  .defaultValue("25.0")
  .units(Units::DegreeCelsius)
  .description("Mean temperature value");
}
```

Example Task: Producer



```
//! Acquire resources.
void
onResourceAcquisition(void)
{
  m_prng = Random::Factory::create(m_args.prng_type,
                                   m_args.prng_seed);
}

//! Release resources.
void
onResourceRelease(void)
{
  Memory::clear(m_prng);
}
```

Example Task: Producer

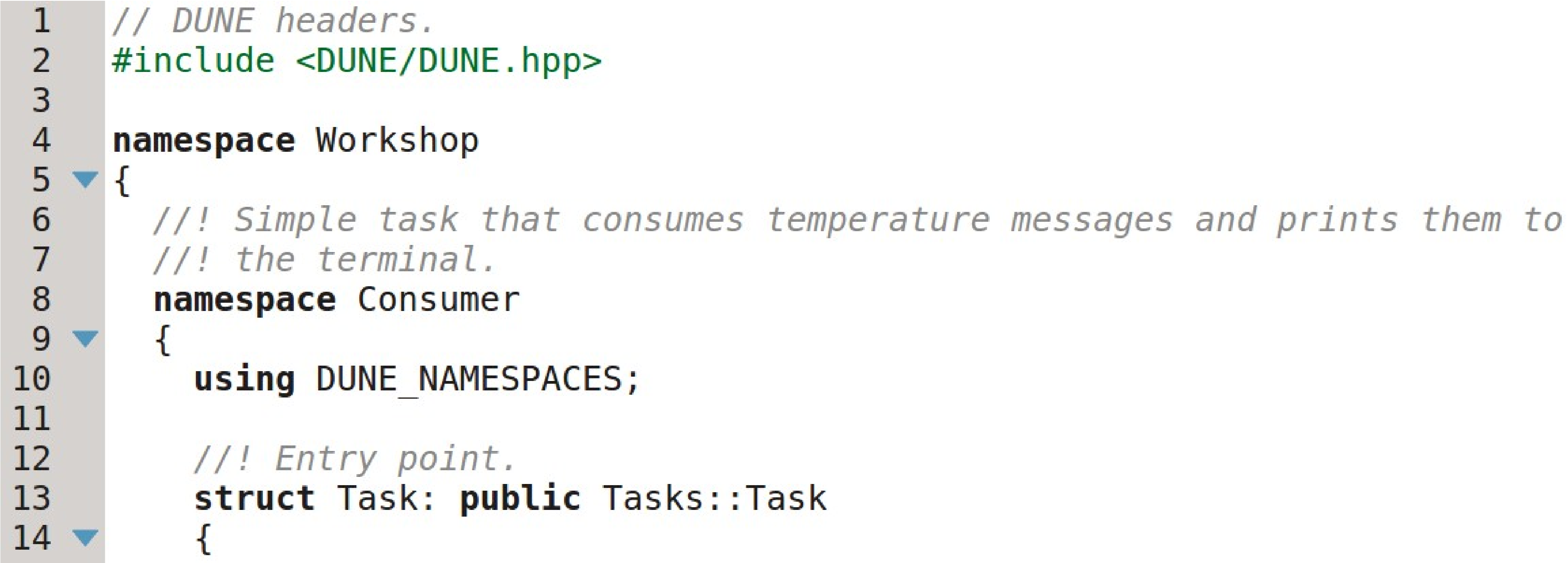


```
      //! Periodic work.
      void
      task(void)
      {
        IMC::Temperature temperature;
        temperature.value = m_args.mean_value
                            + m_prng->gaussian()
                            * m_args.std_dev;
        dispatch(temperature);
      }
    };
  }
}

DUNE_TASK

```

Example Task: Consumer

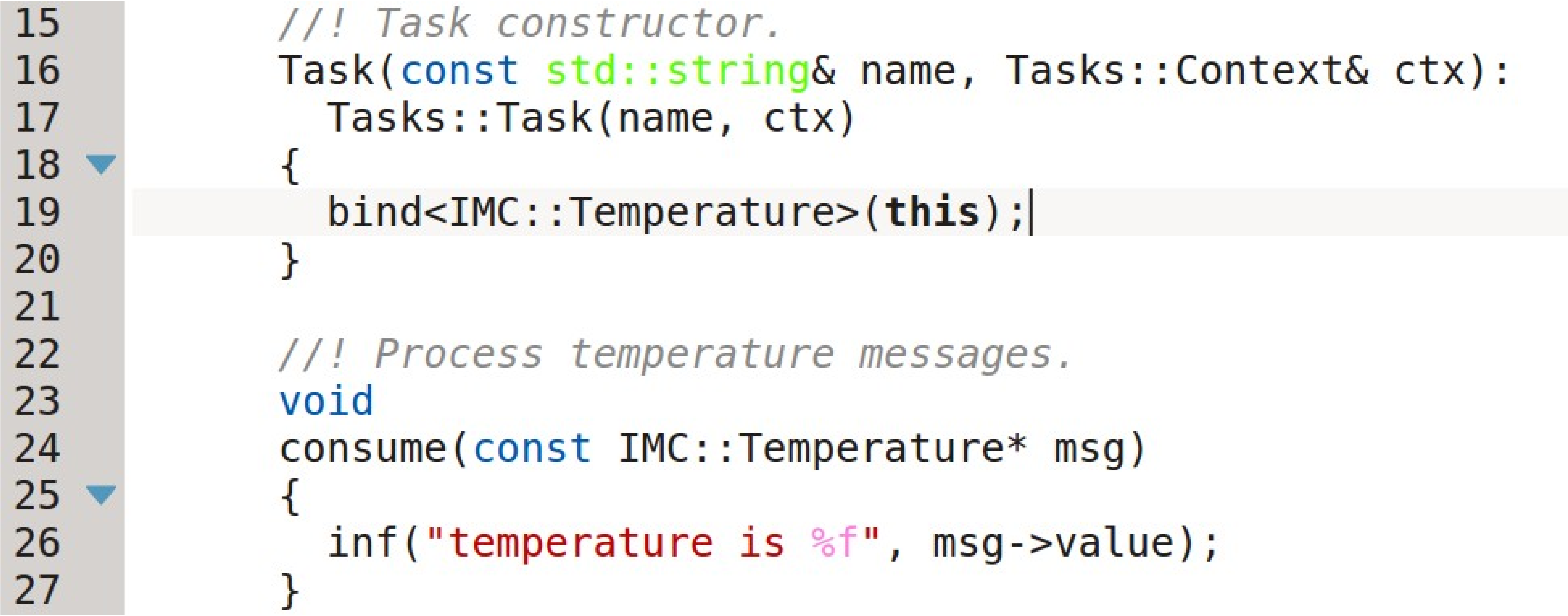
- http://goo.gl/1n2Cpk
- Task consumes temperature messages and prints them to the output (console)
- Scaffold created using the command:
  - python ../dune/programs/scripts/dune-create-task.py ../dune 'Ricardo Martins' 'Workshop/Consumer'
  - make rebuild_cache
- Source code resides in ***src/Workshop/Consumer***
- Task entry point is ***src/Workshop/Consumer/Task.cpp***
- Build: make

Example Task: Consumer



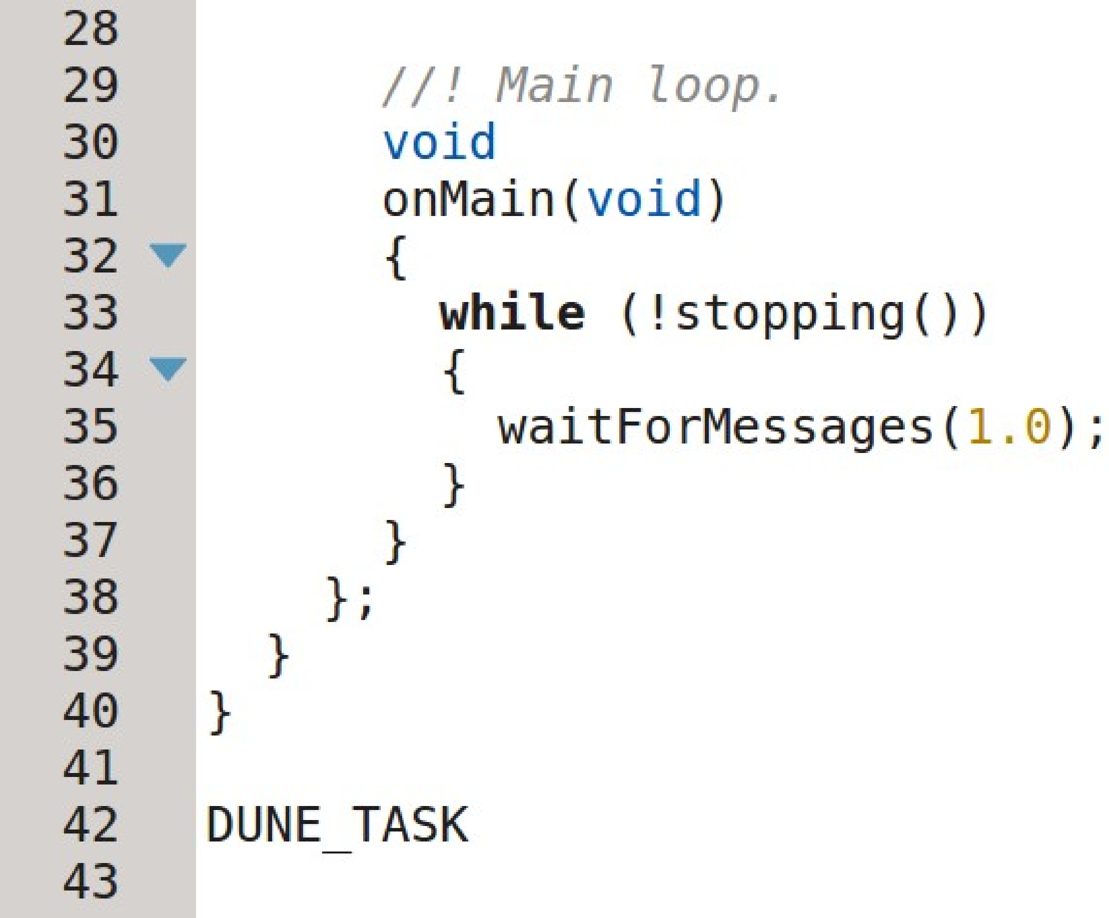
```cpp
// DUNE headers.
#include <DUNE/DUNE.hpp>

namespace Workshop
{
  //! Simple task that consumes temperature messages and prints them to
  //! the terminal.
  namespace Consumer
  {
    using DUNE_NAMESPACES;

    //! Entry point.
    struct Task: public Tasks::Task
    {
```

Example Task: Consumer



```
//! Task constructor.
Task(const std::string& name, Tasks::Context& ctx):
  Tasks::Task(name, ctx)
{
  bind<IMC::Temperature>(this);
}

//! Process temperature messages.
void
consume(const IMC::Temperature* msg)
{
  inf("temperature is %f", msg->value);
}
```

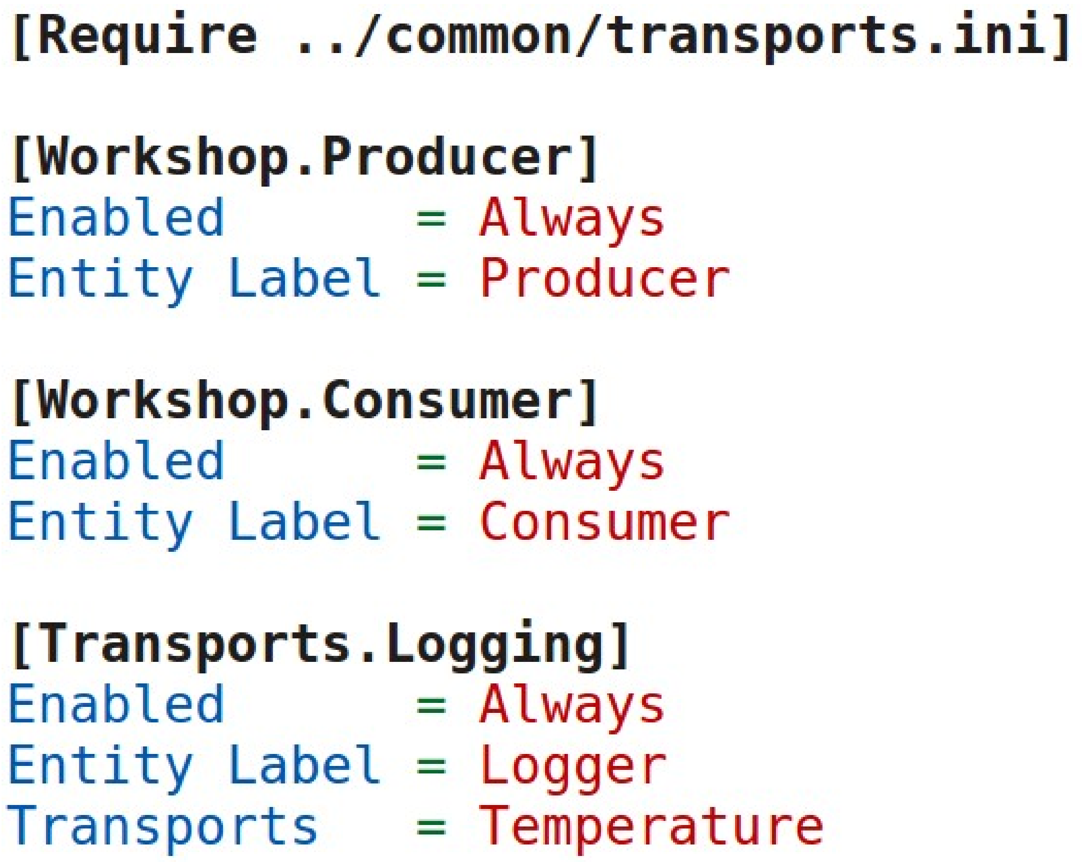
Example Task: Consumer



```

      //! Main loop.
      void
      onMain(void)
      {
        while (!stopping())
        {
          waitForMessages(1.0);
        }
      }
    };
  }
}

DUNE_TASK

```

Configuration File

- http://goo.gl/n4nli4
- Configuration file etc/development/workshop.ini:



```
[Require ../common/transports.ini]

[Workshop.Producer]
Enabled      = Always
Entity Label = Producer

[Workshop.Consumer]
Enabled      = Always
Entity Label = Consumer

[Transports.Logging]
Enabled      = Always
Entity Label = Logger
Transports   = Temperature
```

Runtime Output

Command: ./dune -c development/workshop

```
[2014/11/16 19:51:26] - MSG [Daemon] >> system name: 'unknown' (65535)
[2014/11/16 19:51:26] - MSG [Daemon] >> registered tasks: 160
[2014/11/16 19:51:26] - MSG [Daemon] >> base folder: '/home/rasm/tutorial/build'
[2014/11/16 19:51:26] - MSG [Daemon] >> configuration folder: '/home/rasm/tutorial/dune/etc'
[2014/11/16 19:51:26] - MSG [Daemon] >> web server folder: '/home/rasm/tutorial/dune/www'
[2014/11/16 19:51:26] - MSG [Daemon] >> log folder: '/home/rasm/tutorial/build/log/unknown'
[2014/11/16 19:51:26] - MSG [Daemon] >> library folder: '/home/rasm/tutorial/build'
[2014/11/16 19:51:26] - MSG [Daemon] >> firmware folder: '/home/rasm/tutorial/dune/firmware'
[2014/11/16 19:51:26] - MSG [Transports.Cache] >> starting
[2014/11/16 19:51:26] - MSG [Transports.FTP] >> starting
[2014/11/16 19:51:26] - MSG [Transports.Fragments] >> starting
[2014/11/16 19:51:26] - MSG [Transports.HTTP] >> starting
[2014/11/16 19:51:26] - MSG [Transports.LogBook] >> starting
[2014/11/16 19:51:26] - MSG [Transports.Logging] >> starting
[2014/11/16 19:51:26] - MSG [Workshop.Consumer] >> starting
[2014/11/16 19:51:26] - MSG [Workshop.Producer] >> starting
```

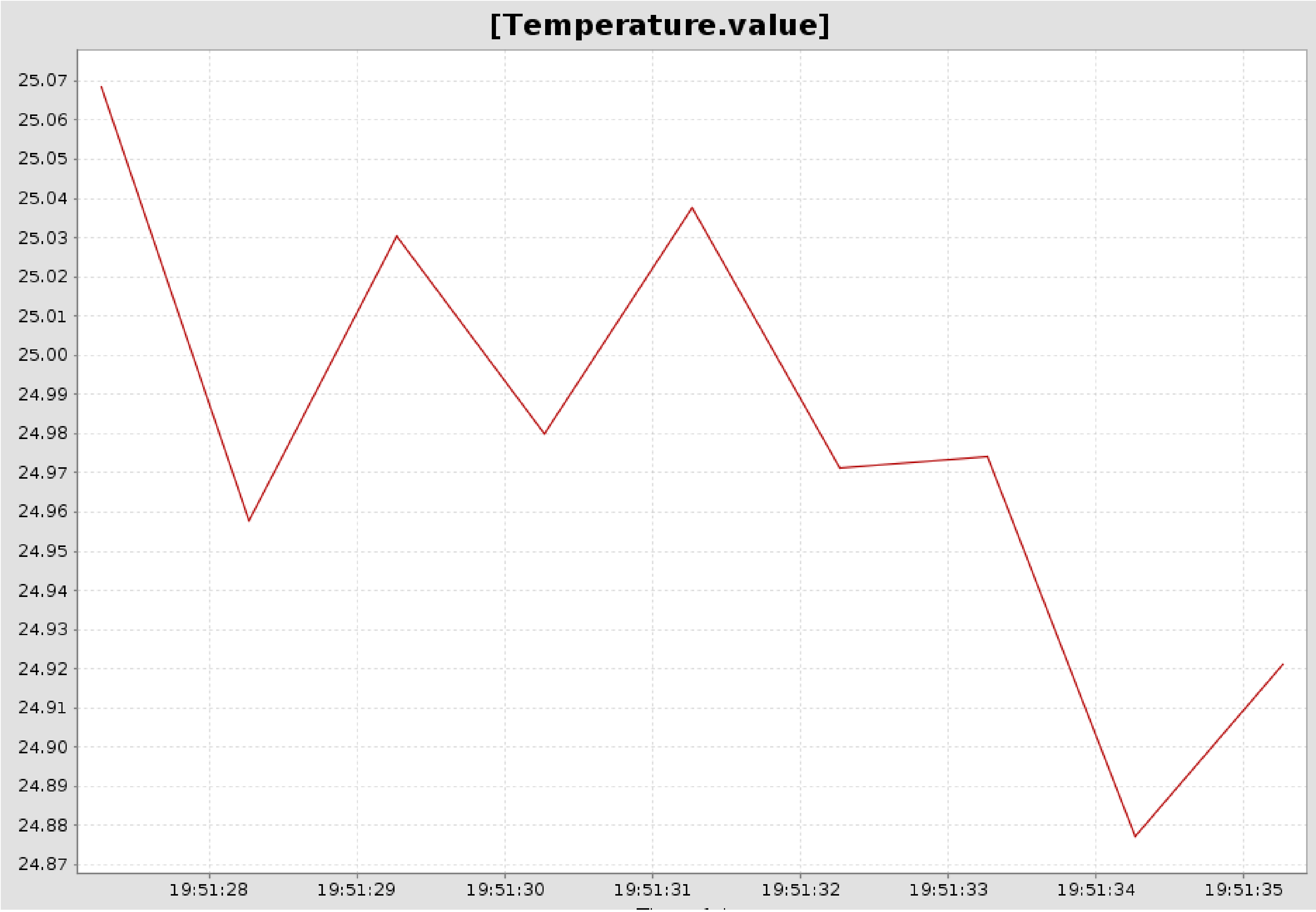
Runtime Output

```
[2014/11/16 19:51:26] - MSG [Transports.HTTP] >> listening on 0.0.0.0:8080
[2014/11/16 19:51:26] - MSG [Transports.Logging] >> log started '20141116/195126'
[2014/11/16 19:51:26] - MSG [Transports.FTP] >> listening on 127.0.0.1:30021
[2014/11/16 19:51:26] - MSG [Transports.FTP] >> listening on 192.168.1.178:30021
[2014/11/16 19:51:26] - MSG [Transports.FTP] >> listening on 10.0.254.1:30021
[2014/11/16 19:51:27] - MSG [Workshop.Consumer] >> temperature is 25.068323
[2014/11/16 19:51:28] - MSG [Workshop.Consumer] >> temperature is 24.957678
[2014/11/16 19:51:29] - MSG [Workshop.Consumer] >> temperature is 25.030371
[2014/11/16 19:51:30] - MSG [Workshop.Consumer] >> temperature is 24.979784
[2014/11/16 19:51:31] - MSG [Workshop.Consumer] >> temperature is 25.037634
[2014/11/16 19:51:32] - MSG [Workshop.Consumer] >> temperature is 24.971085
[2014/11/16 19:51:33] - MSG [Workshop.Consumer] >> temperature is 24.974072
[2014/11/16 19:51:34] - MSG [Workshop.Consumer] >> temperature is 24.877298
```

Log Files

- **DUNE stores log files in the IMC serialization format:**
  - Binary format
  - 1 file for all messages and message types (Data.lsf)
  - Messages are stored roughly in the same order as they were created
  - Supports Gzip and Bzip2 compression (Data.lsf.gz, Data.lsf.bz2)

Log File (Neptus MRA)



Neptus
Command & Control Unit

What's Neptus?

- Neptus allows planning, control and revision of missions performed by unmanned vehicles

- Neptus supports multiple heterogeneous vehicles
  - AUVs, UAVs, ROVs, ASVs, ...
  - Controlled individually or as a team

- Neptus supports multiple operators
  - Operators join in and access / control the network of vehicles

- Neptus can be extended through plug-ins
  - Map layers, Data visualizations, Console widgets, Maneuvers, Communication protocols, ...

Neptus mission concept

- In Neptus, a mission is specified as
  - A set of map features
  - A set of programmed plans
  - A set of vehicle configurations

- The mission is usually...
  - Created prior to execution (planning)
  - Changed during execution (monitoring / revision / re-planning)

LSTS Toolchain For Autonomous Systems

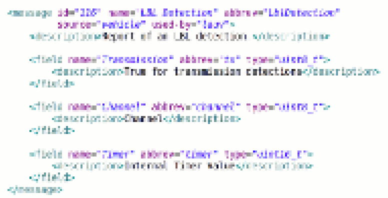
C4I – Command and Control Framework

Neptus

http://whale.fe.up.pt

Inter Module Communications

IMC



```
<message id="326" name="LBL Detection" abbrev="LblDetection"
         source="vehicle" used-by="lauv">
   <description>Report of an LBL detection</description>

   <field name="Transmission" abbrev="tx" type="uint8_t">
      <description>True for transmission detections</description>
   </field>

   <field name="Channel" abbrev="channel" type="uint8_t">
      <description>Channel</description>
   </field>

   <field name="Timer" abbrev="timer" type="uint16_t">
      <description>Internal Timer value</description>
   </field>
</message>
```

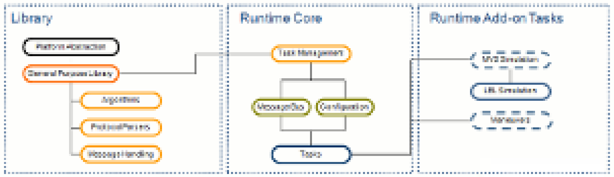
```
--------------------------------------------------------
Heartbeat [size=16]
--------------------------------------------------------
     0: 6c c6 00 00 00 00 60 e5  50 c1 ea 39 d2 41 9f 49
--------------------------------------------------------
```

Message Protocol

http://whale.fe.up.pt

DUNE: Uniform Navigational Environment
On-board software

DUNE



http://whale.fe.up.pt

Part 1: Using Neptus

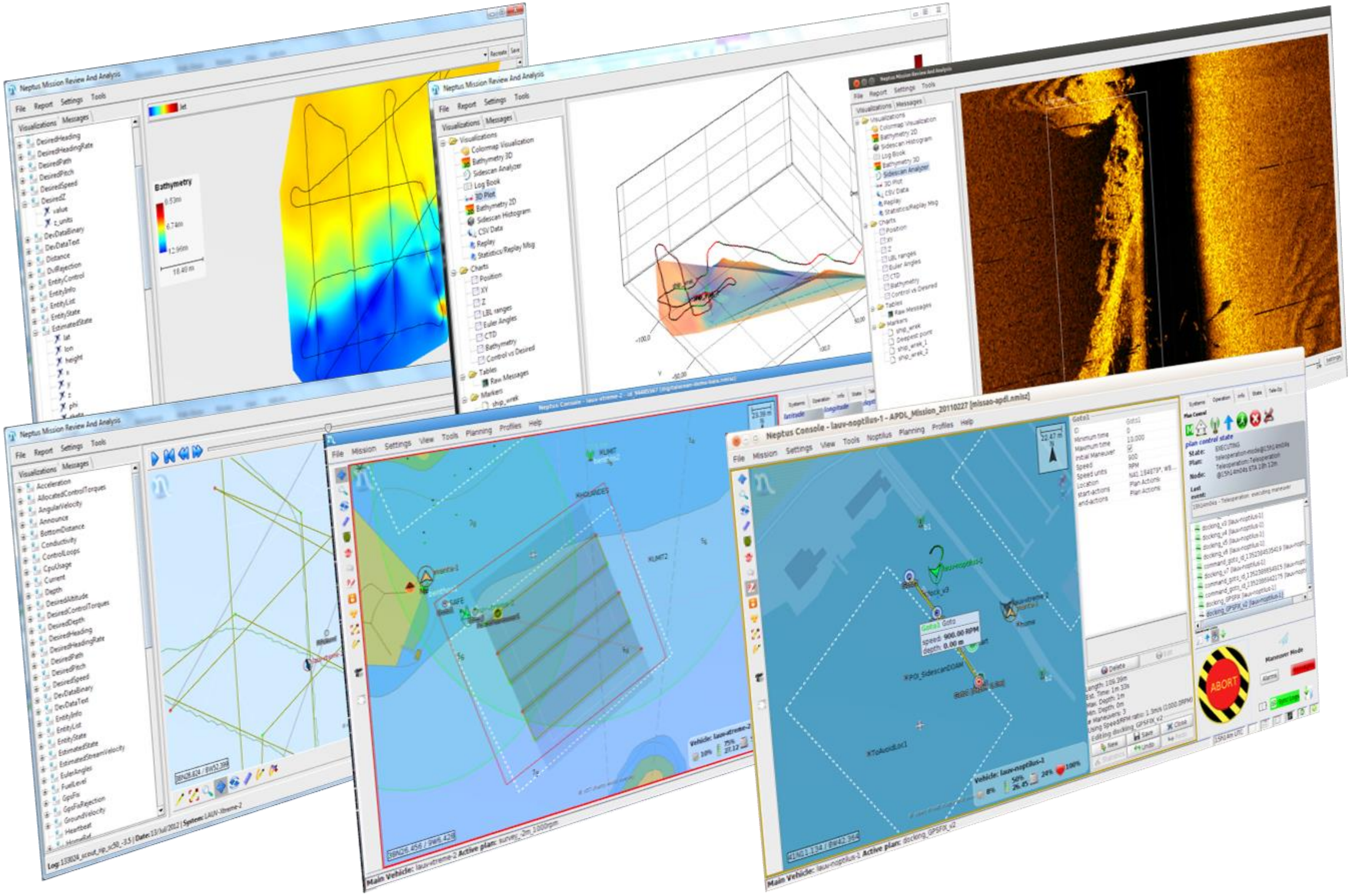
Neptus Requirements

- Neptus requires prior installation of Oracle’s Java Runtime Environment version 7 or newer
- For 3D widgets an OpenGL-compatible graphics adapter is recommended
- At least 1 GB of RAM (4 GB recommended)
- Compatible with Windows and Linux (known to work under OSX but rarely tested)

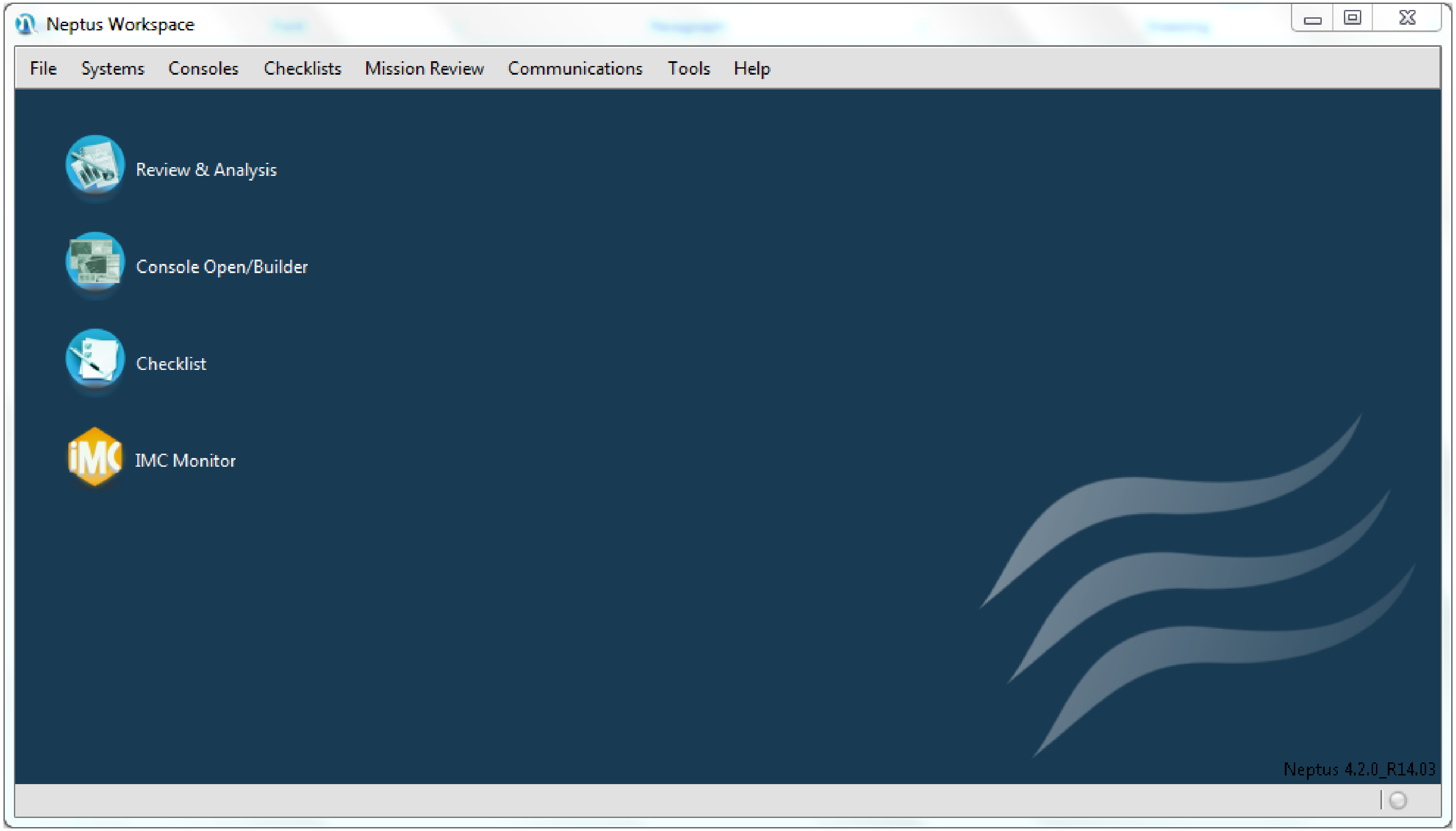
Installing and Running Neptus

- To install Neptus, just download the latest version to a directory of choice
  - Logs will be put under this directory so make sure you leave extra room for them
- Downloading Neptus
  - Use your favorite Git client to clone Neptus from https://github.com/LSTS/neptus
- Running Neptus
  - In Windows: run neptus.exe
  - In Linux: execute ./neptus.sh

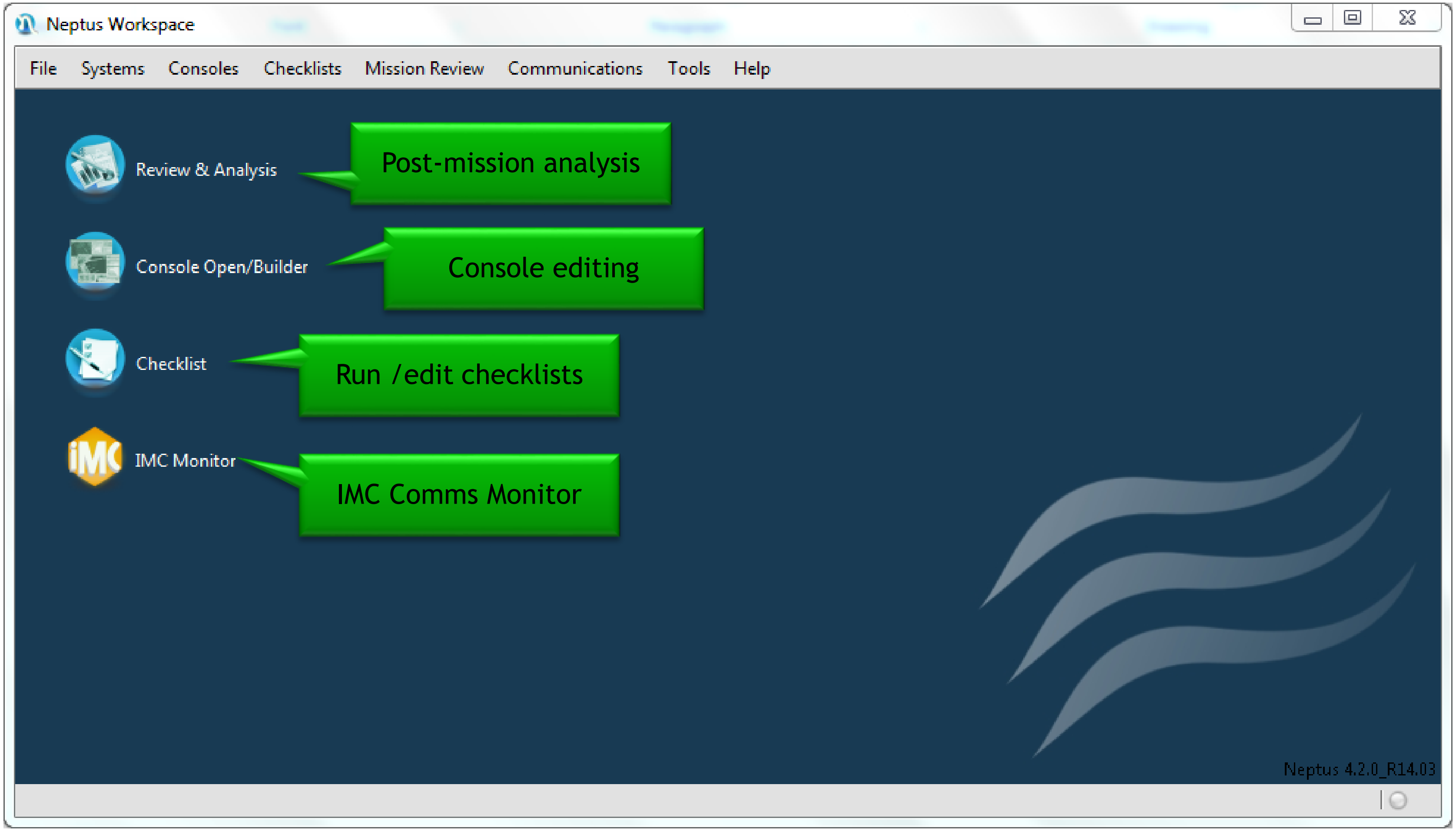
Interfaces Adjusted/Adjustable to Several Needs


Neptus Mission Review And Analysis
File Report Settings Tools
Visualizations Messages
Bathymetry
Colormap Visualization
Bathymetry 3D
Sidescan Analyzer
Log Book
3D Plot
Bathymetry 2D
Sidescan Histogram
CSV Data
Replay
Statistics/Replay Msg
Charts
Position
Tables
Raw Messages
Markers
Neptus Console - lauv-noptilus-1 - APDL_Mission_20110227 [missao-apdl.nmisz]
File Mission Settings View Tools Planning Profiles Help
Main Vehicle: lauv-xtreme-2 Active plan: survey_2m_1000rpm
Main Vehicle: lauv-noptilus-1 Active plan: docking_GPSFIX_v2
Vehicle: lauv-noptilus-1
ABORT
Maneuver Mode

The Neptus Workspace



The Neptus Workspace



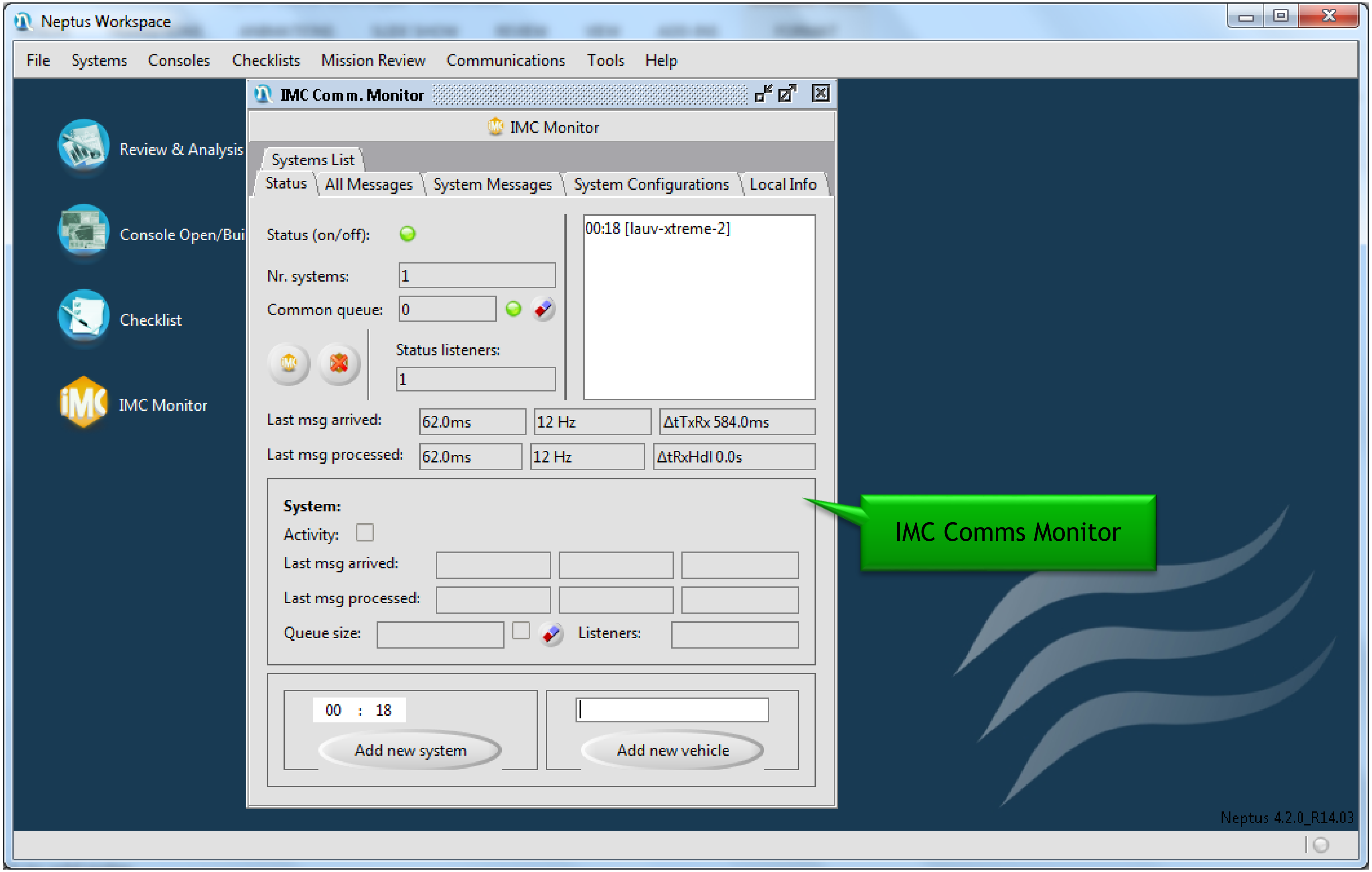
Post-mission analysis

Console editing

Run /edit checklists

IMC Comms Monitor

The Neptus Workspace



Status (on/off): ●

Nr. systems: 1

Common queue: 0 ●

Status listeners: 1

00:18 [lauv-xtreme-2]

Last msg arrived: 62.0ms | 12 Hz | ΔtTxRx 584.0ms

Last msg processed: 62.0ms | 12 Hz | ΔtRxHdl 0.0s

System:

Activity: □

Last msg arrived:

Last msg processed:

Queue size: □ Listeners:

00 : 18

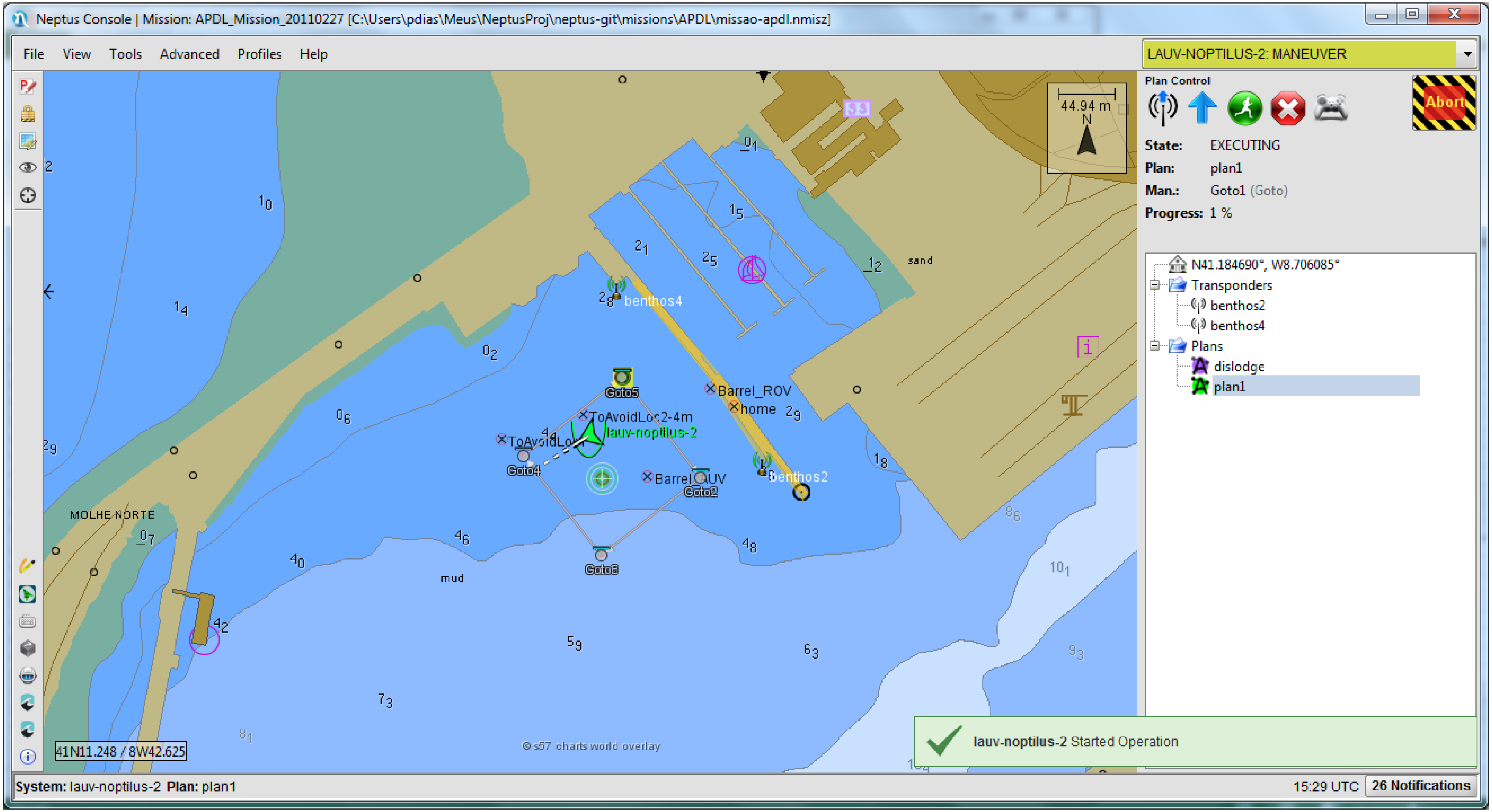
IMC Comms Monitor

Neptus Consoles

- Neptus allow end-users to create Operational Consoles
  - Based on existing widgets
  - Adapted to specific missions/vehicles

- Mission console definitions are stored as XML
  - .ncon file extension
  - A sort of consoles are already bundled

Neptus Consoles



Neptus Console | Mission: APDL_Mission_20110227 [C:\Users\pdias\Meus\NeptusProj\neptus-git\missions\APDL\missao-apdl.nmisz]

State: EXECUTING

Plan: plan1

Man.: Goto1 (Goto)

Progress: 1 %

- N41.184690°, W8.706085°
- Transponders
  - benthos2
  - benthos4
- Plans
  - dislodge
  - plan1

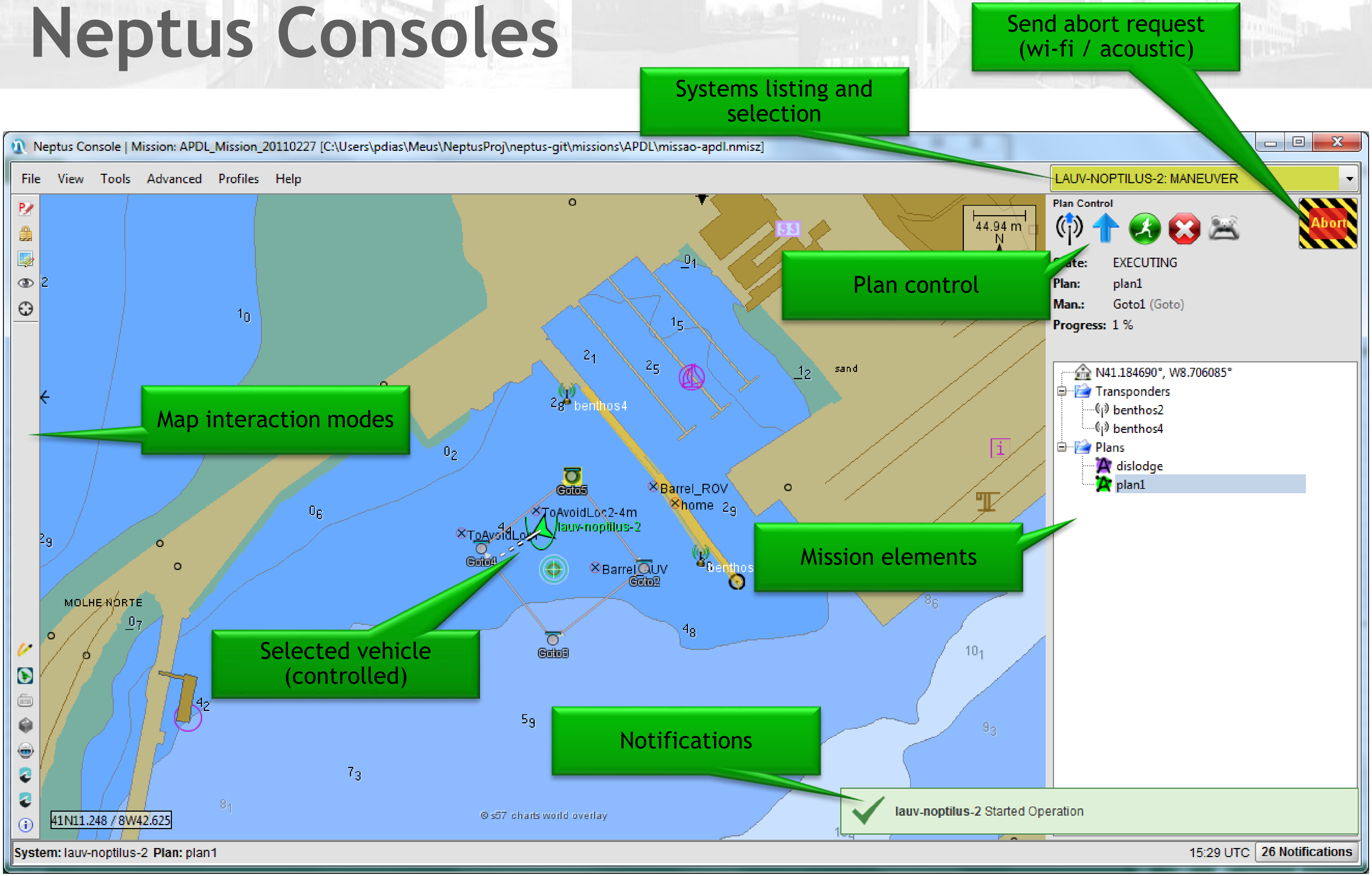
lauv-noptilus-2 Started Operation

System: lauv-noptilus-2 Plan: plan1

15:29 UTC 26 Notifications

FEUP Universidade do Porto Faculdade de Engenharia

Neptus Consoles



Send abort request (wi-fi / acoustic)

Systems listing and selection

Plan control

Map interaction modes

Mission elements

Selected vehicle (controlled)

Notifications

Neptus Console | Mission: APDL_Mission_20110227 [C:\Users\pdias\Meus\NeptusProj\neptus-git\missions\APDL\missao-apdl.nmisz]

LAUV-NOPTILUS-2: MANEUVER

Plan Control

State: EXECUTING

Plan: plan1

Man.: Goto1 (Goto)

Progress: 1 %

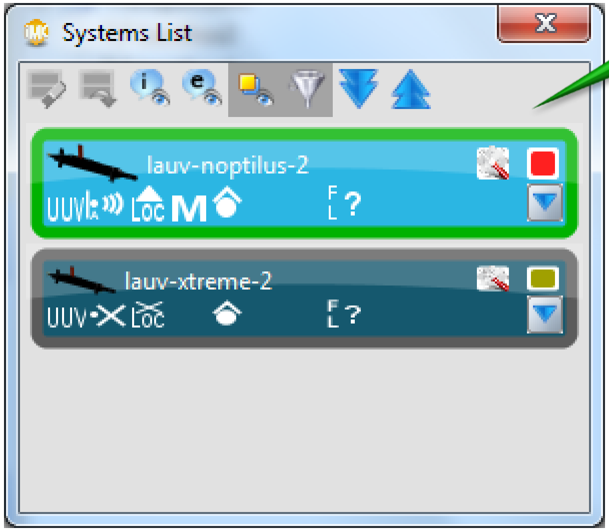
- N41.184690°, W8.706085°
- Transponders
  - benthos2
  - benthos4
- Plans
  - dislodge
  - plan1

lauv-noptilus-2 Started Operation

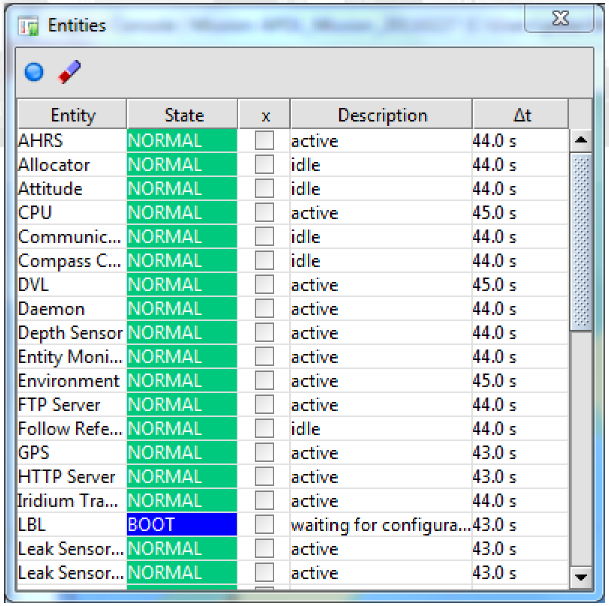
System: lauv-noptilus-2 Plan: plan1

15:29 UTC 26 Notifications

Neptus Consoles



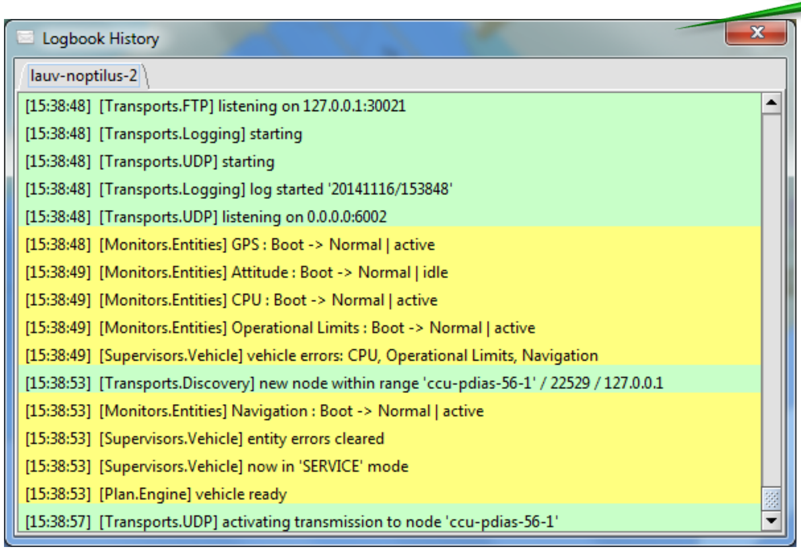
Systems listing and selection



Entity	State	x	Description	Δt
AHRS	NORMAL	□	active	44.0 s
Allocator	NORMAL	□	idle	44.0 s
Attitude	NORMAL	□	idle	44.0 s
CPU	NORMAL	□	active	45.0 s
Communic...	NORMAL	□	idle	44.0 s
Compass C...	NORMAL	□	idle	44.0 s
DVL	NORMAL	□	active	45.0 s
Daemon	NORMAL	□	active	44.0 s
Depth Sensor	NORMAL	□	active	44.0 s
Entity Moni...	NORMAL	□	active	44.0 s
Environment	NORMAL	□	active	45.0 s
FTP Server	NORMAL	□	active	44.0 s
Follow Refe...	NORMAL	□	idle	44.0 s
GPS	NORMAL	□	active	43.0 s
HTTP Server	NORMAL	□	active	43.0 s
Iridium Tra...	NORMAL	□	active	44.0 s
LBL	BOOT	□	waiting for configura...	43.0 s
Leak Sensor...	NORMAL	□	active	43.0 s
Leak Sensor...	NORMAL	□	active	43.0 s

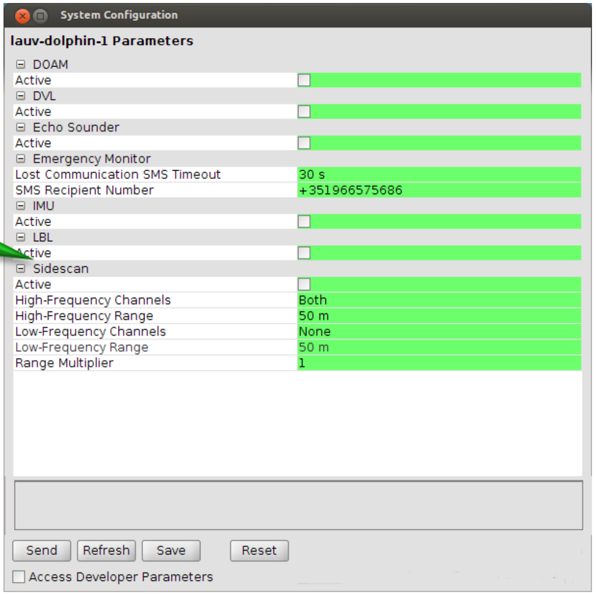
Vehicle subsystems state

Vehicle Log Book



```
[15:38:48]  [Transports.FTP] listening on 127.0.0.1:30021
[15:38:48]  [Transports.Logging] starting
[15:38:48]  [Transports.UDP] starting
[15:38:48]  [Transports.Logging] log started '20141116/153848'
[15:38:48]  [Transports.UDP] listening on 0.0.0.0:6002
[15:38:48]  [Monitors.Entities] GPS : Boot -> Normal | active
[15:38:49]  [Monitors.Entities] Attitude : Boot -> Normal | idle
[15:38:49]  [Monitors.Entities] CPU : Boot -> Normal | active
[15:38:49]  [Monitors.Entities] Operational Limits : Boot -> Normal | active
[15:38:49]  [Supervisors.Vehicle] vehicle errors: CPU, Operational Limits, Navigation
[15:38:53]  [Transports.Discovery] new node within range 'ccu-pdias-56-1' / 22529 / 127.0.0.1
[15:38:53]  [Monitors.Entities] Navigation : Boot -> Normal | active
[15:38:53]  [Supervisors.Vehicle] entity errors cleared
[15:38:53]  [Supervisors.Vehicle] now in 'SERVICE' mode
[15:38:53]  [Plan.Engine] vehicle ready
[15:38:57]  [Transports.UDP] activating transmission to node 'ccu-pdias-56-1'
```

Vehicles Configurations

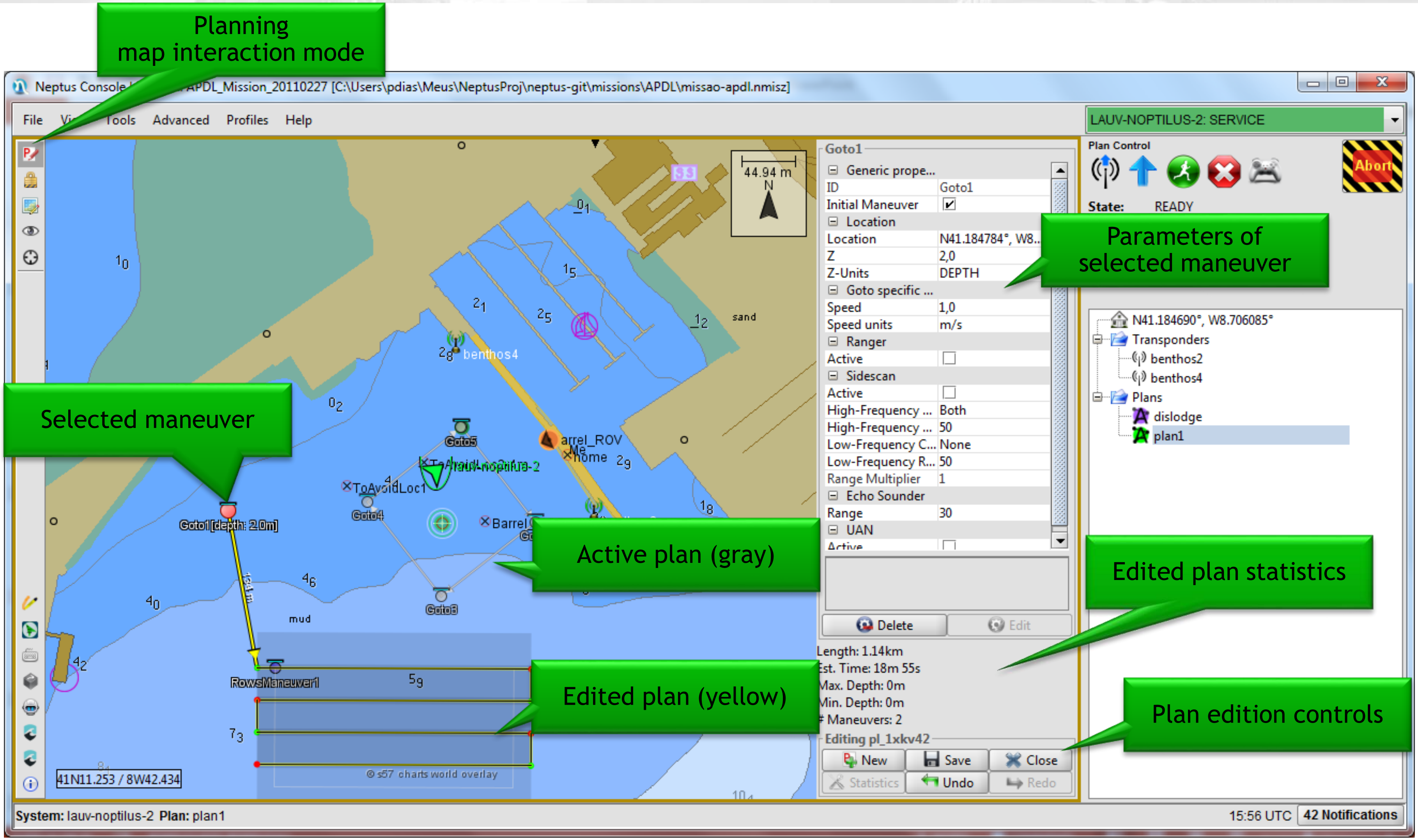


lauv-dolphin-1 Parameters

⊟ DOAM	
Active	□
⊟ DVL	
Active	□
⊟ Echo Sounder	
Active	□
⊟ Emergency Monitor	
Lost Communication SMS Timeout	30 s
SMS Recipient Number	+351966575686
⊟ IMU	
Active	□
⊟ LBL	
Active	□
⊟ Sidescan	
Active	□
High-Frequency Channels	Both
High-Frequency Range	50 m
Low-Frequency Channels	None
Low-Frequency Range	50 m
Range Multiplier	1

□ Access Developer Parameters

Neptus Consoles



Planning map interaction mode

Selected maneuver

Parameters of selected maneuver

Active plan (gray)

Edited plan (yellow)

Edited plan statistics

Plan edition controls

Neptus Console APDL_Mission_20110227 [C:\Users\pdias\Meus\NeptusProj\neptus-git\missions\APDL\missao-apdl.nmisz]

LAUV-NOPTILUS-2: SERVICE

Plan Control

State: READY

Goto1

⊟ Generic prope...	
ID	Goto1
Initial Maneuver	☑
⊟ Location	
Location	N41.184784°, W8..
Z	2,0
Z-Units	DEPTH
⊟ Goto specific ...	
Speed	1,0
Speed units	m/s
⊟ Ranger	
Active	□
⊟ Sidescan	
Active	□
High-Frequency ...	Both
High-Frequency ...	50
Low-Frequency C...	None
Low-Frequency R...	50
Range Multiplier	1
⊟ Echo Sounder	
Range	30
⊟ UAN	
Active	□

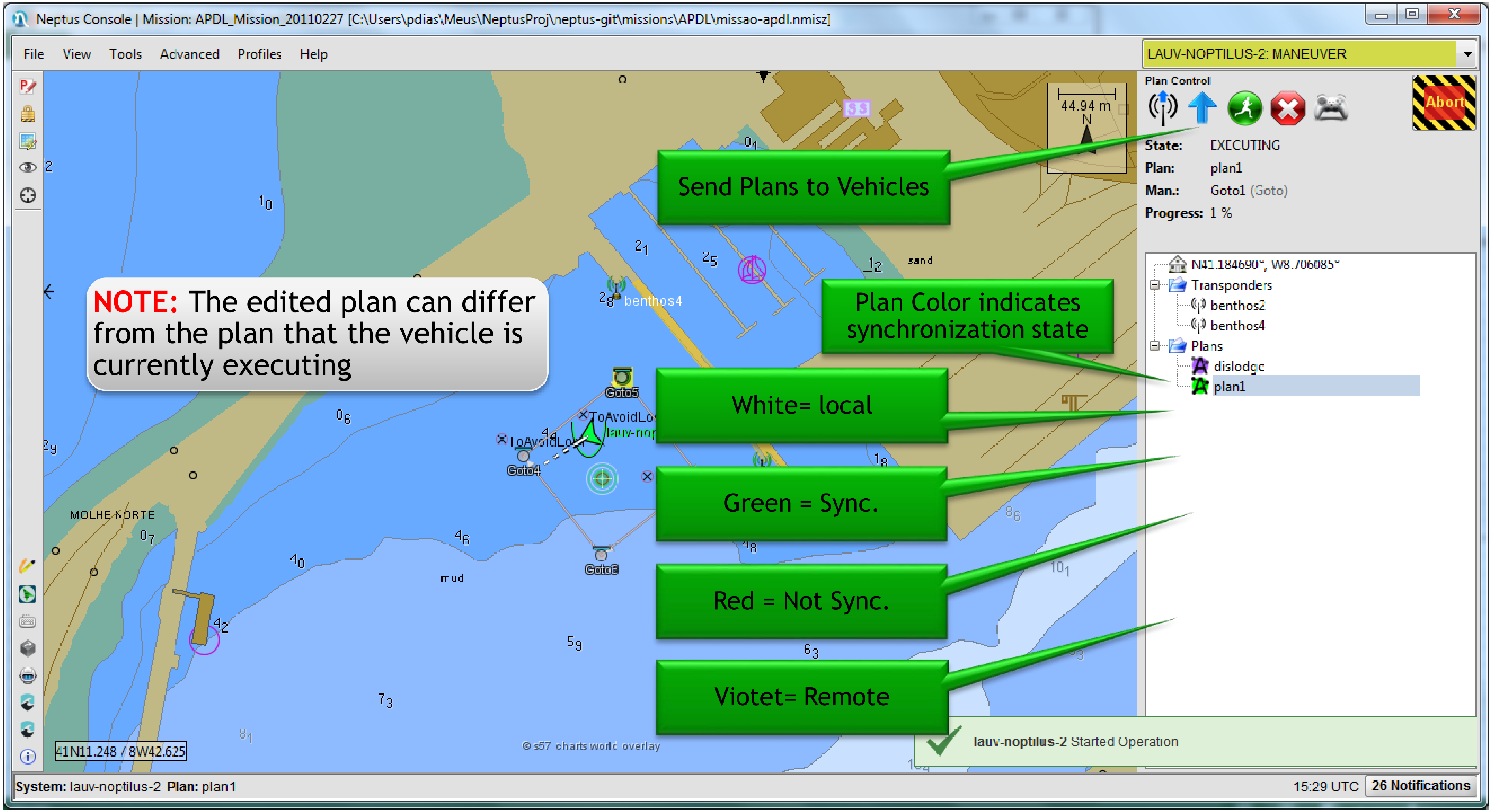
Length: 1.14km
Est. Time: 18m 55s
Max. Depth: 0m
Min. Depth: 0m
Maneuvers: 2

Editing pl_1xkv42

System: lauv-noptilus-2 Plan: plan1

15:56 UTC 42 Notifications

Neptus Consoles



Neptus Console | Mission: APDL_Mission_20110227 [C:\Users\pdias\Meus\NeptusProj\neptus-git\missions\APDL\missao-apdl.nmisz]

LAUV-NOPTILUS-2: MANEUVER

Plan Control

State: EXECUTING
Plan: plan1
Man.: Goto1 (Goto)
Progress: 1 %

N41.184690°, W8.706085°
Transponders
benthos2
benthos4
Plans
dislodge
plan1

lauv-noptilus-2 Started Operation

System: lauv-noptilus-2 Plan: plan1

15:29 UTC

26 Notifications

41N11.248 / 8W42.625

NOTE: The edited plan can differ from the plan that the vehicle is currently executing

Send Plans to Vehicles

Plan Color indicates synchronization state

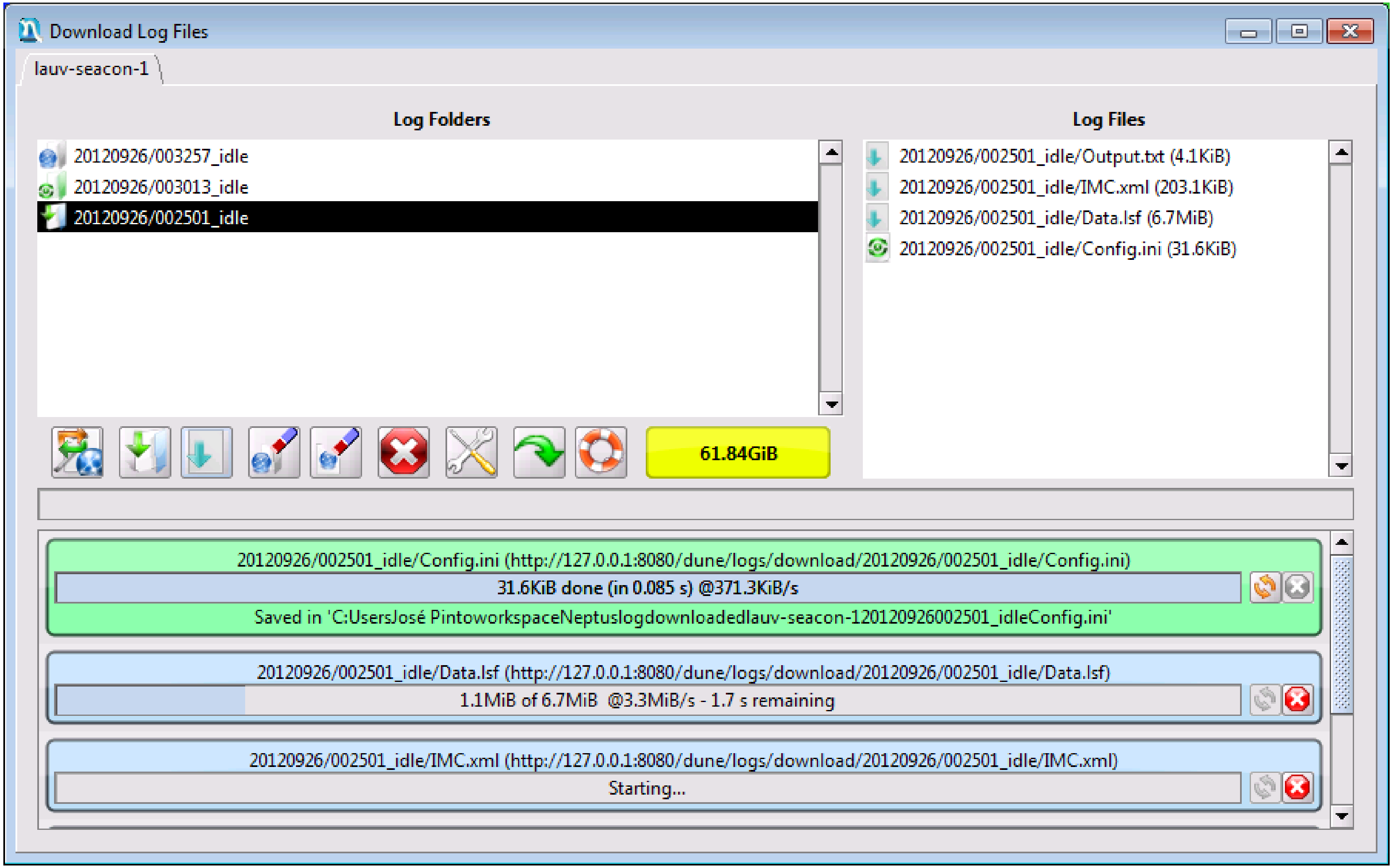
White= local

Green = Sync.

Red = Not Sync.

Viotet= Remote

Neptus Consoles – Log Download Dialog



Log Folders:

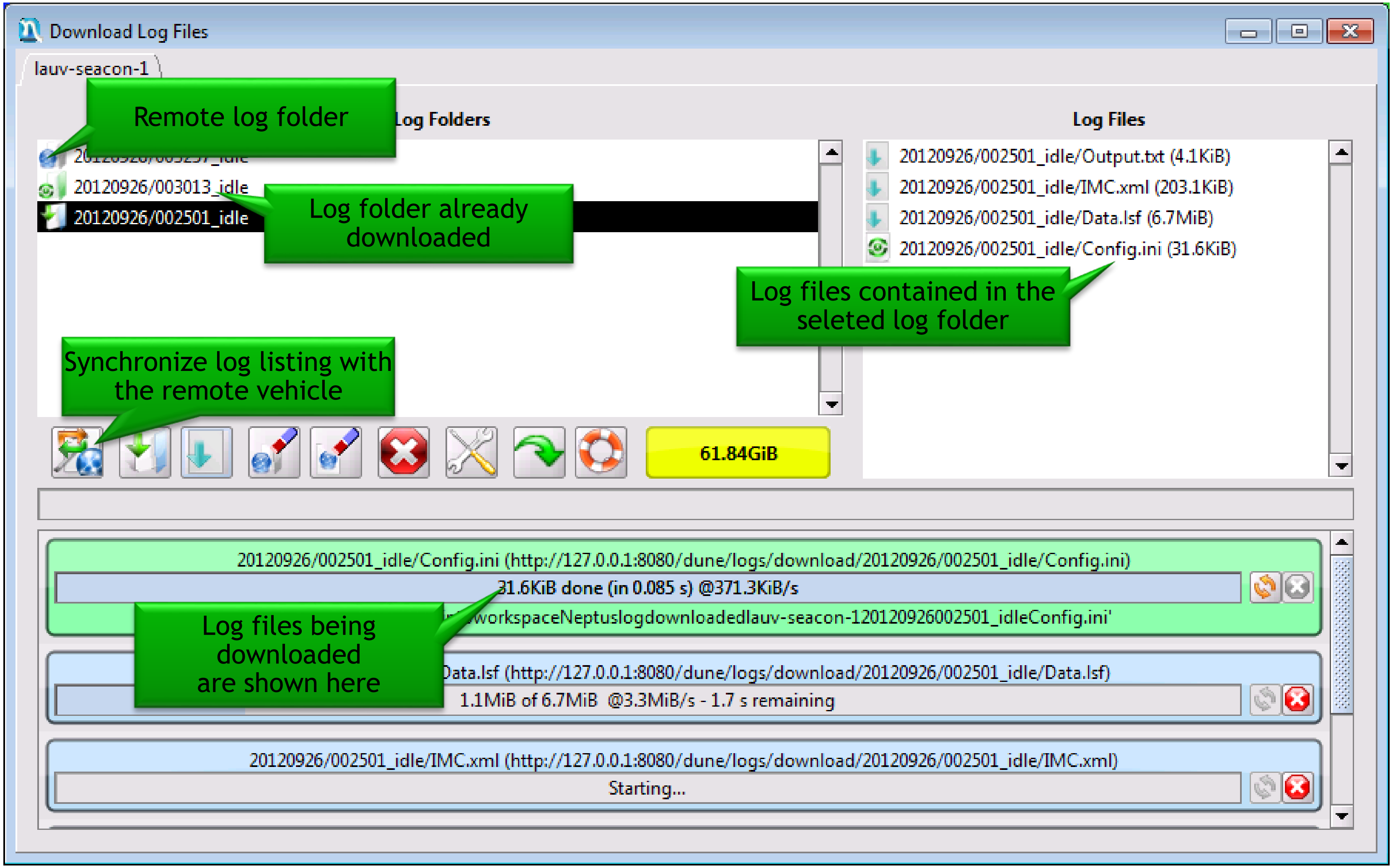
```
20120926/003257_idle
20120926/003013_idle
20120926/002501_idle
```

Log Files:

```
20120926/002501_idle/Output.txt (4.1KiB)
20120926/002501_idle/IMC.xml (203.1KiB)
20120926/002501_idle/Data.lsf (6.7MiB)
20120926/002501_idle/Config.ini (31.6KiB)
```

```
20120926/002501_idle/Config.ini (http://127.0.0.1:8080/dune/logs/download/20120926/002501_idle/Config.ini)
31.6KiB done (in 0.085 s) @371.3KiB/s
Saved in 'C:UsersJosé PintoworkspaceNeptuslogdownloadedlauv-seacon-120120926002501_idleConfig.ini'

20120926/002501_idle/Data.lsf (http://127.0.0.1:8080/dune/logs/download/20120926/002501_idle/Data.lsf)
1.1MiB of 6.7MiB  @3.3MiB/s - 1.7 s remaining

20120926/002501_idle/IMC.xml (http://127.0.0.1:8080/dune/logs/download/20120926/002501_idle/IMC.xml)
Starting...
```

Neptus Consoles – Log Download Dialog



20120926/003237_idle
20120926/003013_idle
20120926/002501_idle

20120926/002501_idle/Output.txt (4.1KiB)
20120926/002501_idle/IMC.xml (203.1KiB)
20120926/002501_idle/Data.lsf (6.7MiB)
20120926/002501_idle/Config.ini (31.6KiB)

61.84GiB

20120926/002501_idle/Config.ini (http://127.0.0.1:8080/dune/logs/download/20120926/002501_idle/Config.ini)
31.6KiB done (in 0.085 s) @371.3KiB/s
workspaceNeptuslogdownloadedlauv-seacon-120120926002501_idleConfig.ini'

Data.lsf (http://127.0.0.1:8080/dune/logs/download/20120926/002501_idle/Data.lsf)
1.1MiB of 6.7MiB @3.3MiB/s - 1.7 s remaining

20120926/002501_idle/IMC.xml (http://127.0.0.1:8080/dune/logs/download/20120926/002501_idle/IMC.xml)
Starting...

Remote log folder

Log folder already downloaded

Log files contained in the seleted log folder

Synchronize log listing with the remote vehicle

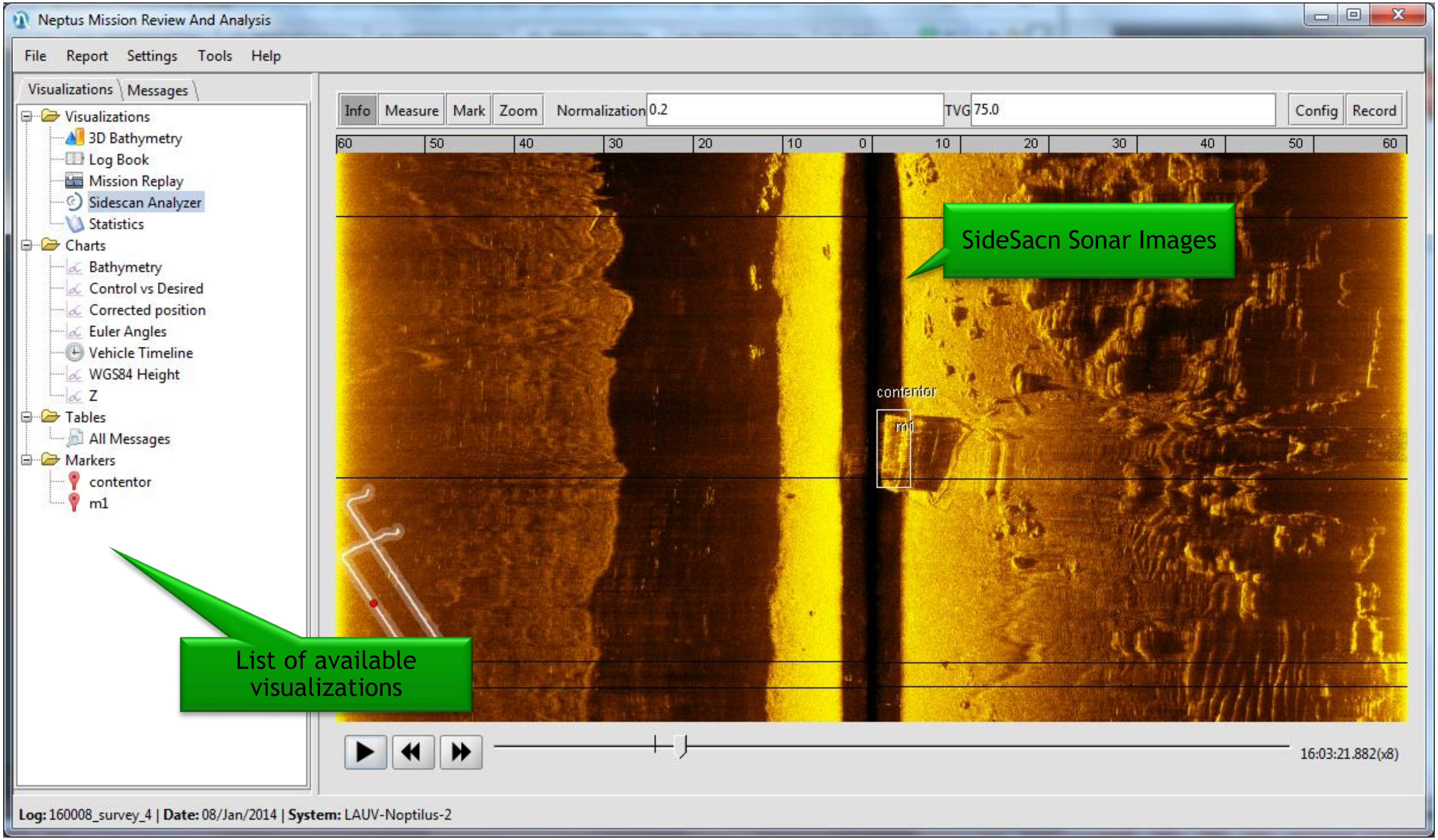
Log files being downloaded are shown here

Neptus Mission Review and Analysis

- Can be acessed
  - Directly by right-clicking a downloaded log
  - From the Neptus workspace

- Compatible with LSF log folders
  - Data.lsf (binary concatenation of IMC data)
  - IMC.xml (definition of the protocol used in the LSF)
  - config.ini (used vehicle configuration)

Neptus Mission Review and Analysis



Normalization 0.2

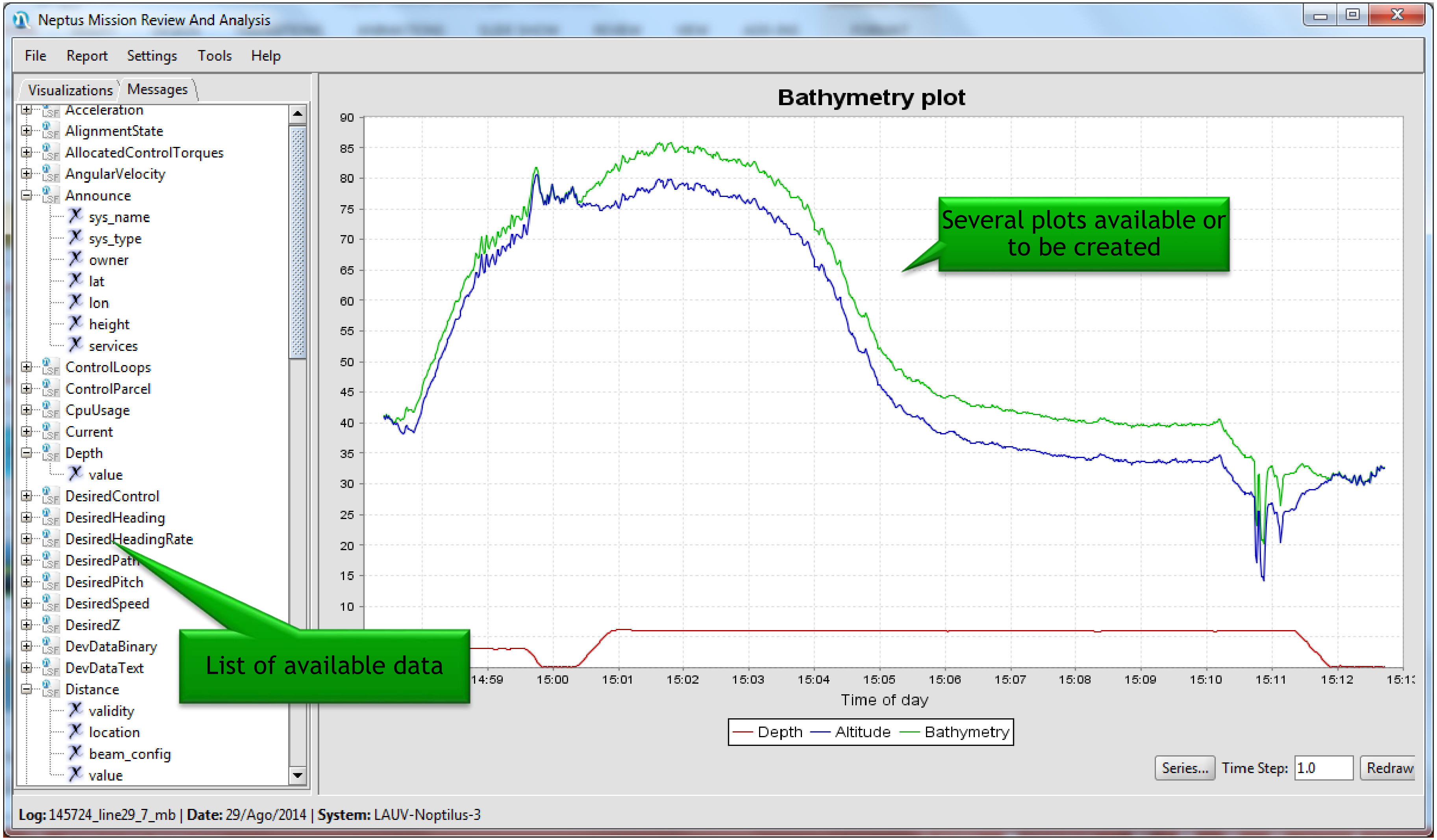
TVG 75.0

Log: 160008_survey_4 | Date: 08/Jan/2014 | System: LAUV-Noptilus-2

SideSacn Sonar Images

List of available visualizations

Neptus Mission Review and Analysis



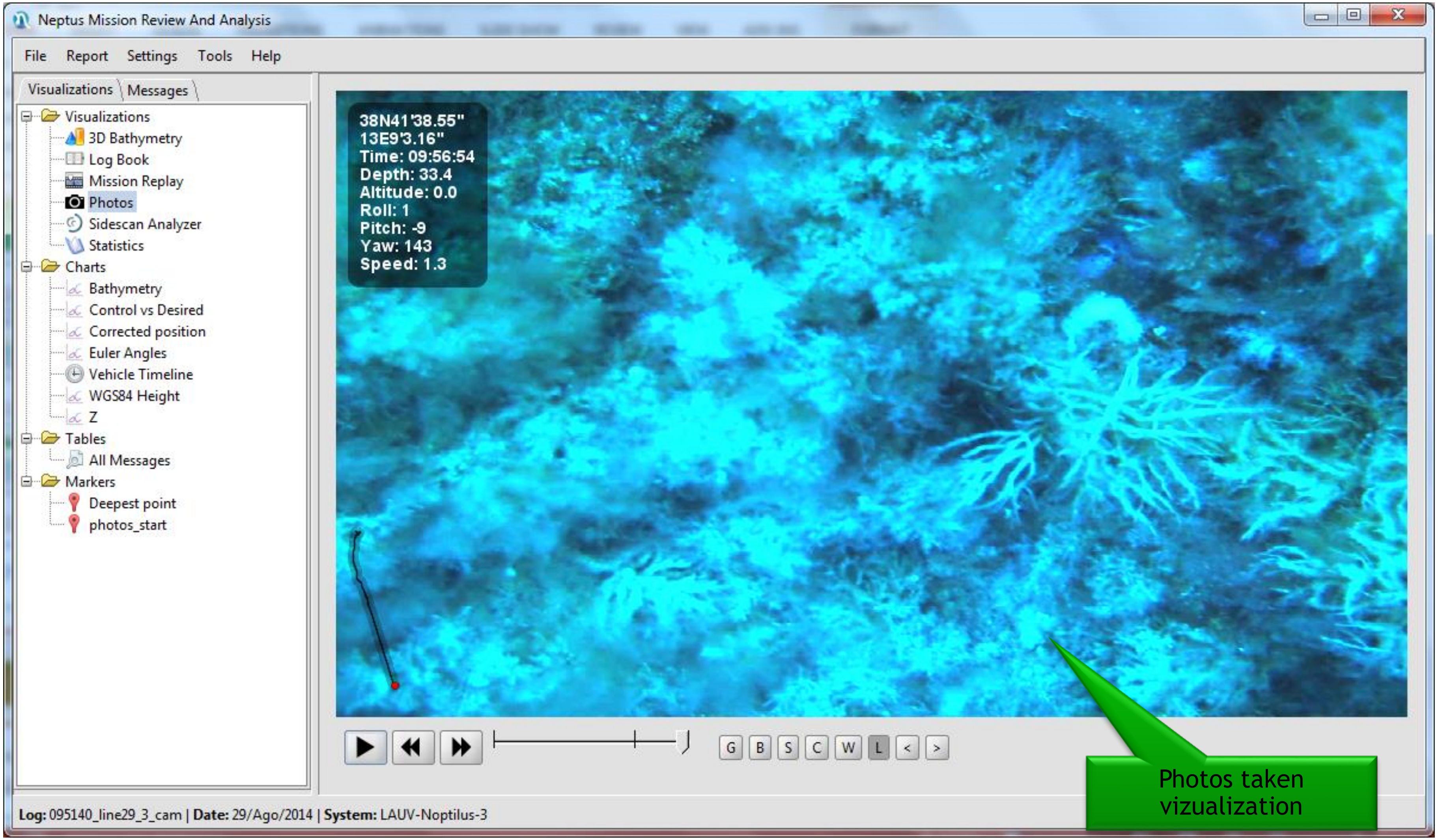
Time Step: 1.0

Log: 145724_line29_7_mb | **Date:** 29/Ago/2014 | **System:** LAUV-Noptilus-3

Several plots available or to be created

List of available data

Neptus Mission Review and Analysis

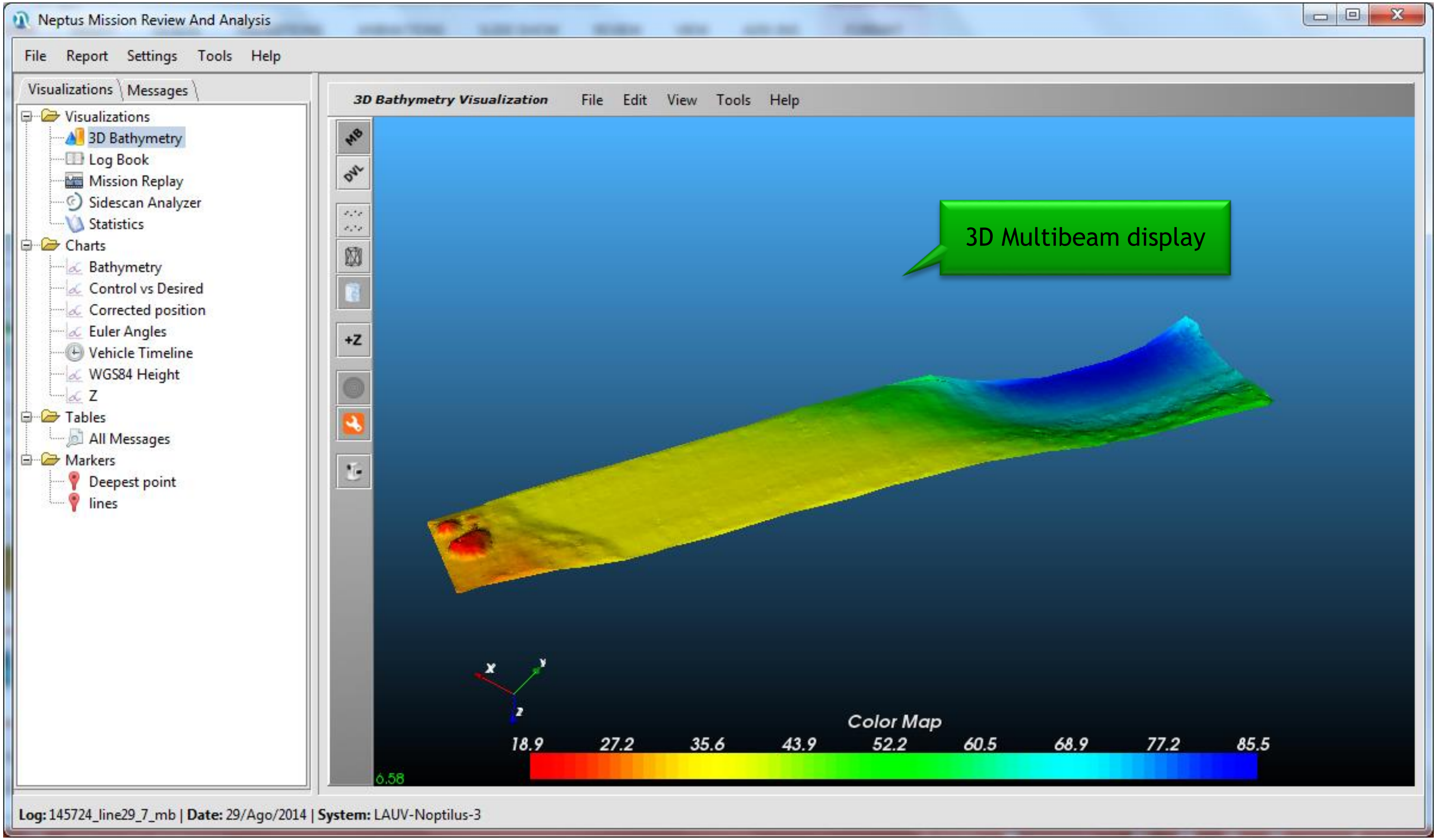


38N41'38.55"
13E9'3.16"
Time: 09:56:54
Depth: 33.4
Altitude: 0.0
Roll: 1
Pitch: -9
Yaw: 143
Speed: 1.3

Log: 095140_line29_3_cam | **Date:** 29/Ago/2014 | **System:** LAUV-Noptilus-3

Photos taken vizualization

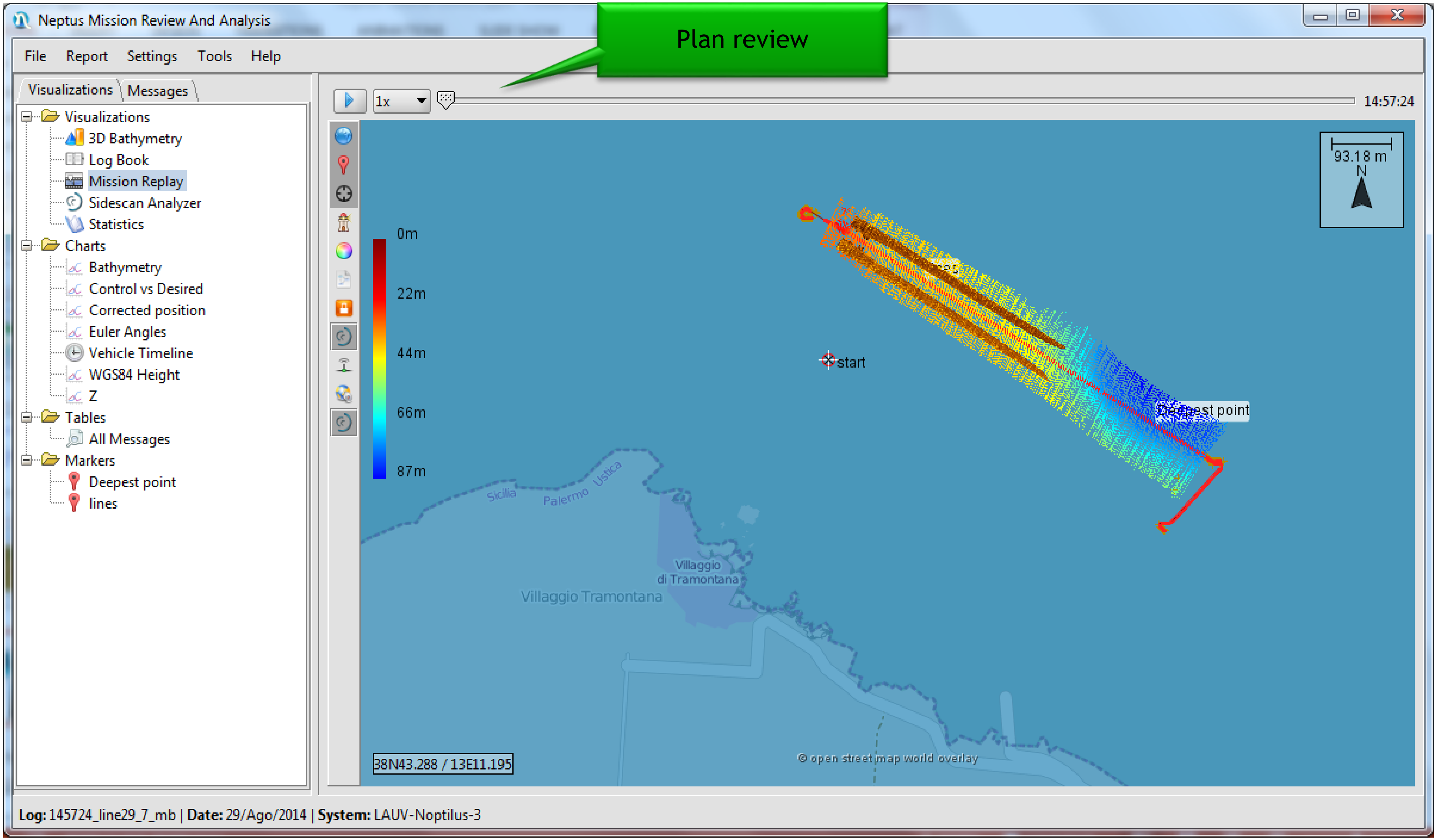
Neptus Mission Review and Analysis



3D Multibeam display

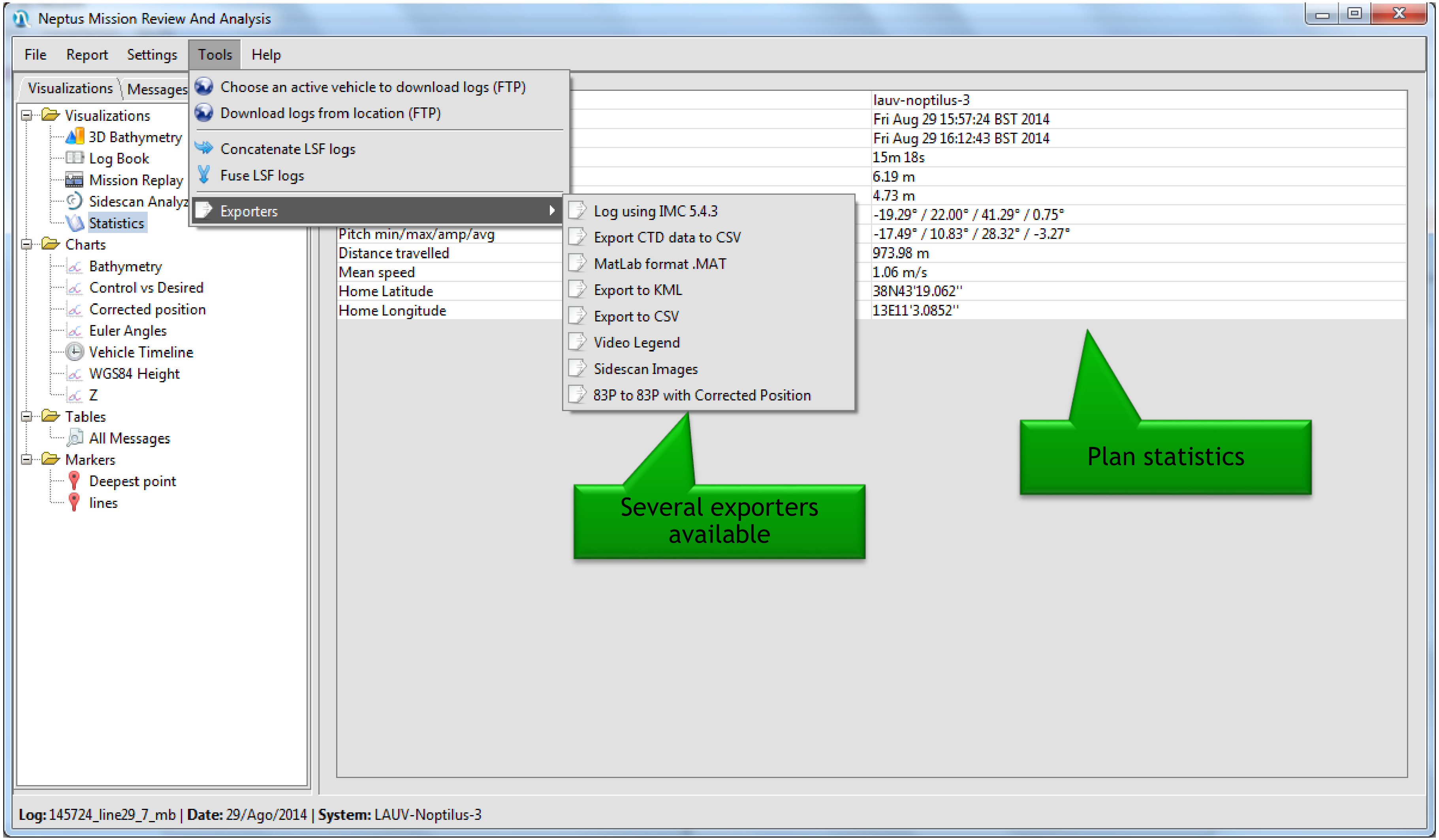
Log: 145724_line29_7_mb | **Date:** 29/Ago/2014 | **System:** LAUV-Noptilus-3

Neptus Mission Review and Analysis



Log: 145724_line29_7_mb | **Date:** 29/Ago/2014 | **System:** LAUV-Noptilus-3

Neptus Mission Review and Analysis



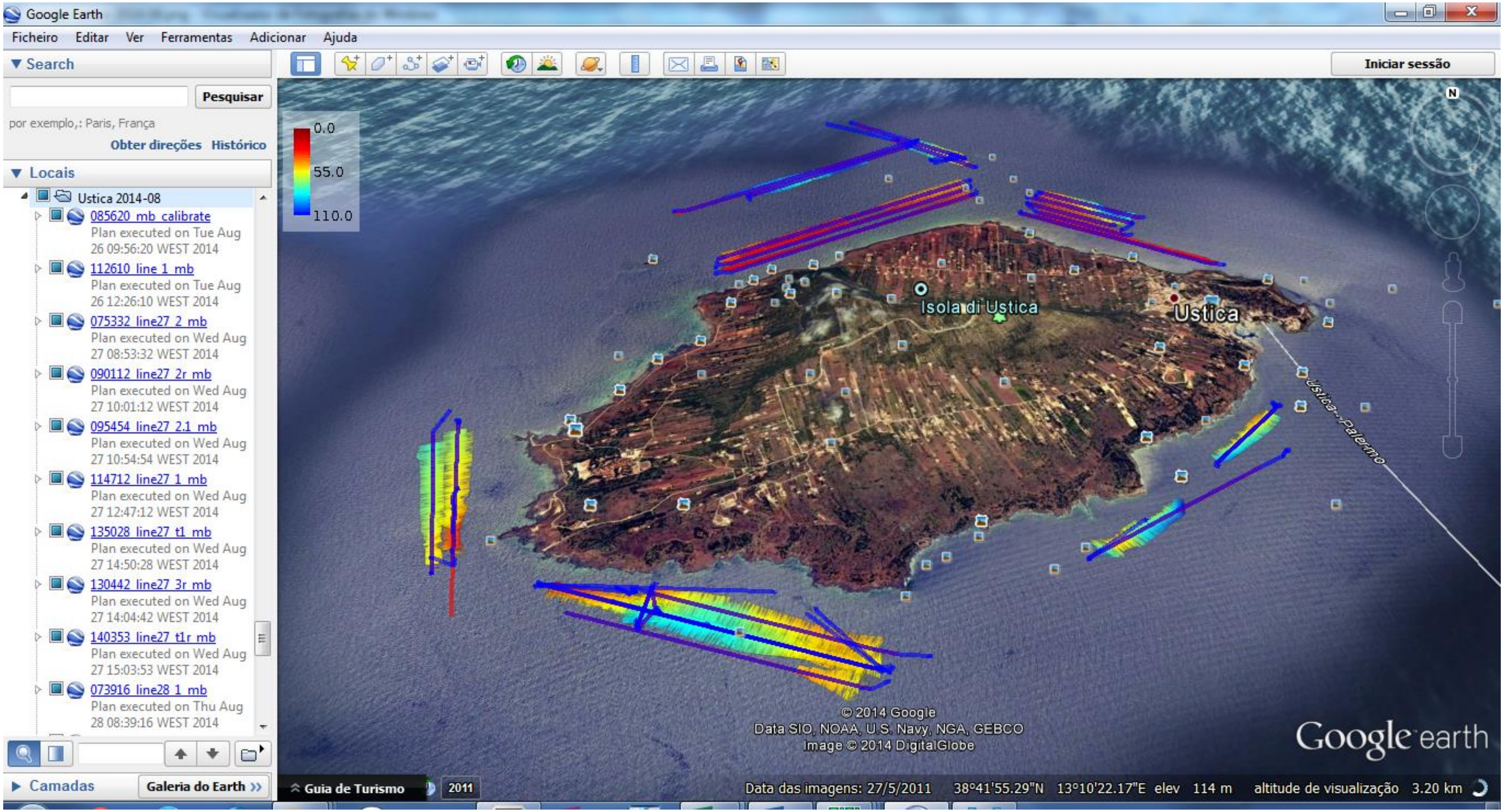
	lauv-noptilus-3
	Fri Aug 29 15:57:24 BST 2014
	Fri Aug 29 16:12:43 BST 2014
	15m 18s
	6.19 m
	4.73 m
	-19.29° / 22.00° / 41.29° / 0.75°
Pitch min/max/amp/avg	-17.49° / 10.83° / 28.32° / -3.27°
Distance travelled	973.98 m
Mean speed	1.06 m/s
Home Latitude	38N43'19.062''
Home Longitude	13E11'3.0852''

Log: 145724_line29_7_mb | **Date:** 29/Ago/2014 | **System:** LAUV-Noptilus-3

Several exporters available

Plan statistics

Neptus KML Export



Ustica 2014-08

085620_mb_calibrate
Plan executed on Tue Aug 26 09:56:20 WEST 2014

112610_line_1_mb
Plan executed on Tue Aug 26 12:26:10 WEST 2014

075332_line27_2_mb
Plan executed on Wed Aug 27 08:53:32 WEST 2014

090112_line27_2r_mb
Plan executed on Wed Aug 27 10:01:12 WEST 2014

095454_line27_2.1_mb
Plan executed on Wed Aug 27 10:54:54 WEST 2014

114712_line27_1_mb
Plan executed on Wed Aug 27 12:47:12 WEST 2014

135028_line27_t1_mb
Plan executed on Wed Aug 27 14:50:28 WEST 2014

130442_line27_3r_mb
Plan executed on Wed Aug 27 14:04:42 WEST 2014

140353_line27_t1r_mb
Plan executed on Wed Aug 27 15:03:53 WEST 2014

073916_line28_1_mb
Plan executed on Thu Aug 28 08:39:16 WEST 2014

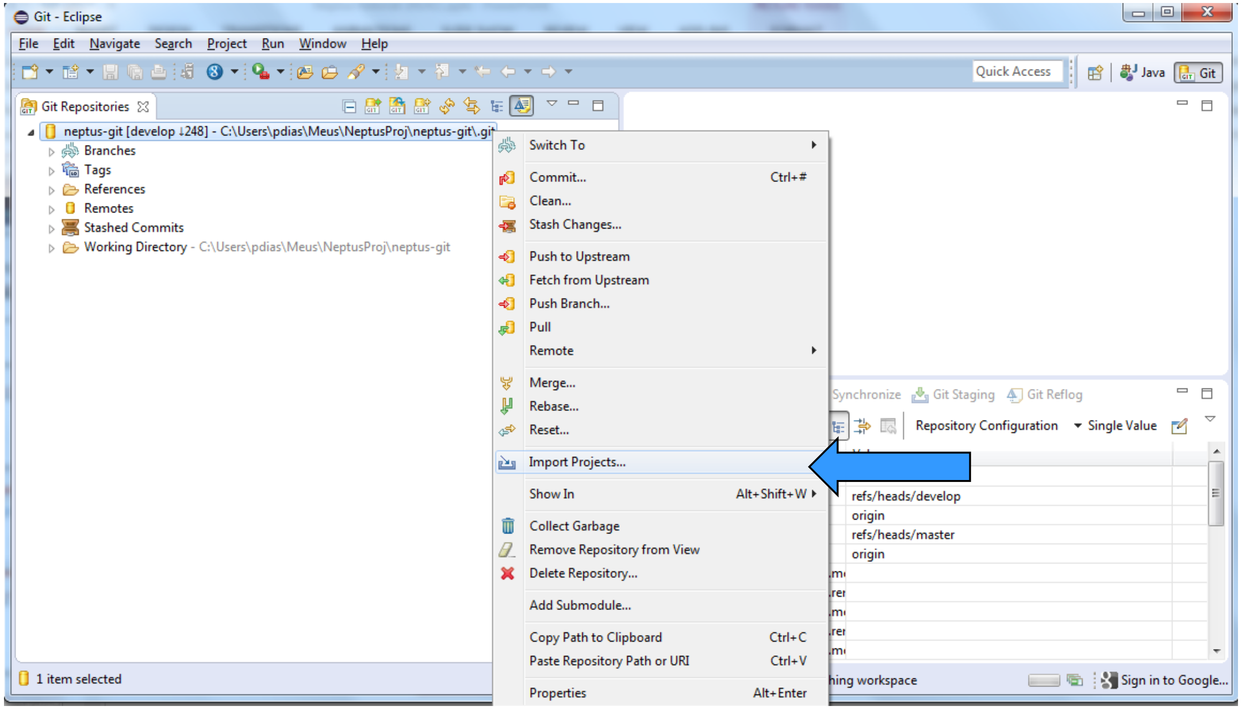
Part 2: Extending Neptus

Requirements for Extending Neptus

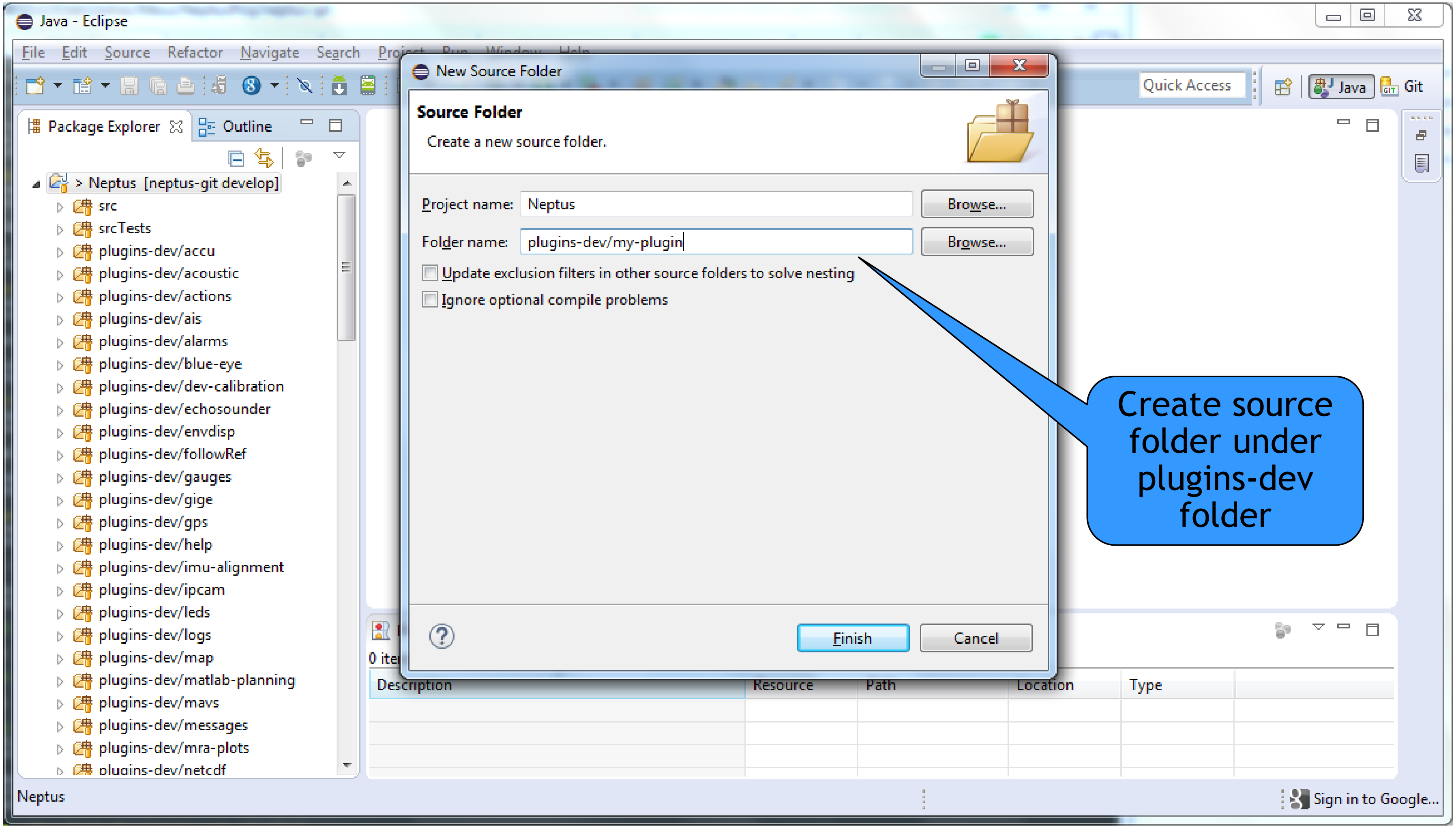
- Installation of Oracle’s Java JDK version 7 or newer
- Git (Source Control Management)
- Ant (Build System)
- Eclipse Luna for Java Developers

Setup Your Development Tool

- Clone Neptus
  - Use your favorite Git client to clone Neptus from https://github.com/LSTS/neptus
- Configure Eclipse



Creating a plug-in



Source Folder

Create a new source folder.

Project name: Neptus

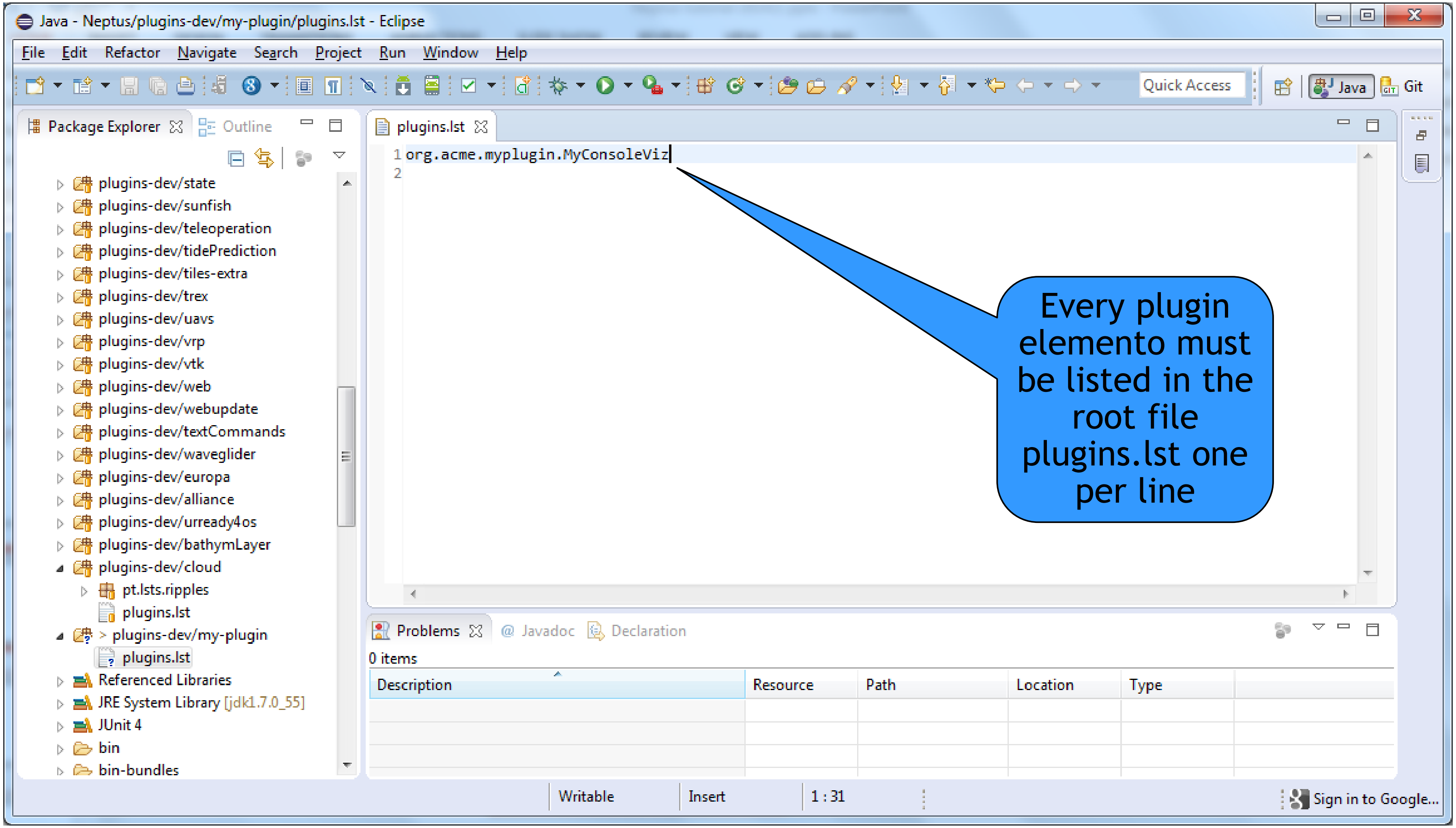
Folder name: plugins-dev/my-plugin

□ Update exclusion filters in other source folders to solve nesting

□ Ignore optional compile problems

Create source folder under plugins-dev folder

Plug-in properties



```
org.acme.myplugin.MyConsoleViz
```

Every plugin elemento must be listed in the root file plugins.lst one per line

Plug-in example - Console Widget

```
package org.acme.myplugin;
import pt.lsts.neptus.console.ConsoleLayout;
…
/**
 * @author You
 *
 */
@PluginDescription(name = "My Console Viz")
@Popup(pos = POSITION.RIGHT, width = 200, height = 200, accelerator = ‘Y')
@SuppressWarnings("serial")
public class MyConsoleViz extends ConsolePanel {

    /**
     * @param console
     */
    public MyConsoleViz(ConsoleLayout console) {
        super(console);
    }

    @Override
    public void initSubPanel() {
    }

    @Override
    public void cleanSubPanel() {
    }
}
```

Every plugin is anotated with PluginDescription

Optionally the panel may be a popup dialog instead of living in the main window

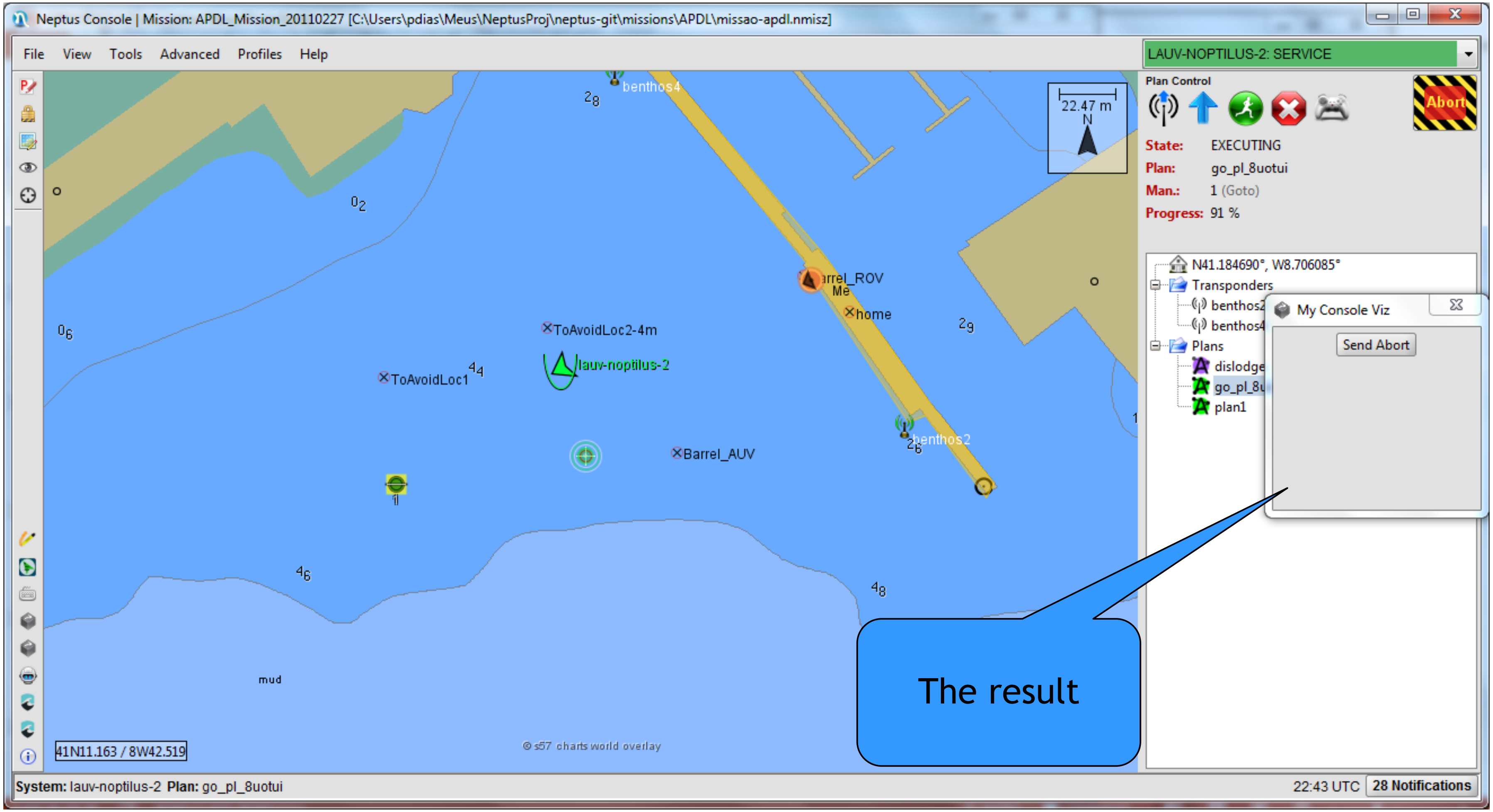
Base console widget extends ConsolePanel

Plug-in example - Console Widget

```
@Override
public void initSubPanel() {
    removeAll();

    Action sendAbortAction = new AbstractAction(I18n.text("Send Abort")) {
        @Override
        public void actionPerformed(ActionEvent e) {
            Abort abortMsg = new Abort();
            send(abortMsg);
        }
    };
    sendAbort = new JButton(sendAbortAction);

    add(sendAbort);
}
```

Lets make a panel to send na abort command

Plug-in example - Console Widget



Neptus Console | Mission: APDL_Mission_20110227 [C:\Users\pdias\Meus\NeptusProj\neptus-git\missions\APDL\missao-apdl.nmisz]

LAUV-NOPTILUS-2: SERVICE

Plan Control

State: EXECUTING
Plan: go_pl_8uotui
Man.: 1 (Goto)
Progress: 91 %

N41.184690°, W8.706085°
Transponders
benthos2
benthos4
Plans
dislodge
go_pl_8u
plan1

My Console Viz

Send Abort

41N11.163 / 8W42.519

System: lauv-noptilus-2 Plan: go_pl_8uotui

22:43 UTC 28 Notifications

The result

Plugin example - Map Layer

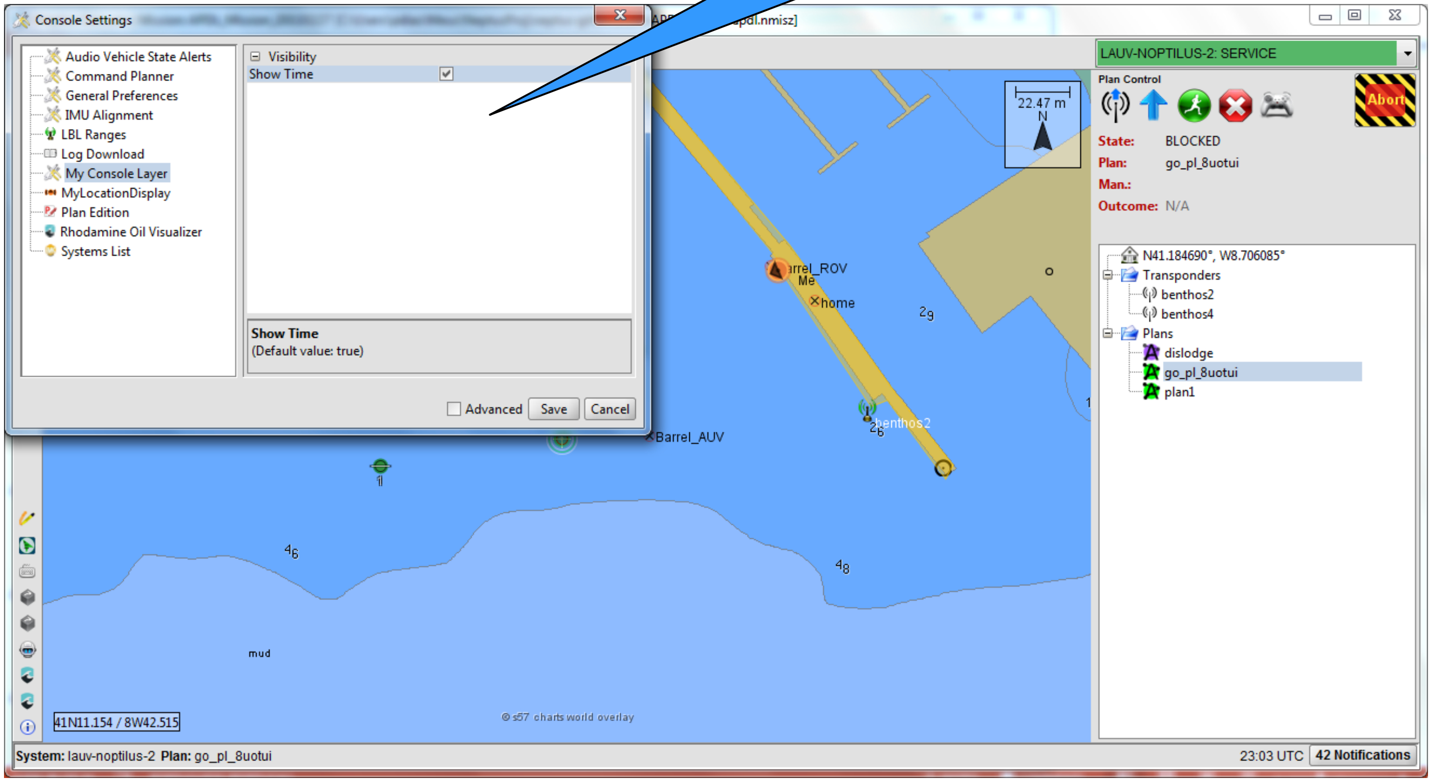
```java
package org.acme.myplugin;
import pt.lsts.neptus.console.ConsoleLayer;
…
/**
 * @author You
 *
 */
@PluginDescription(name = "My Console Layer")
@LayerPriority(priority = 66)
public class MyConsoleLayer extends ConsoleLayer {
    public MyConsoleLayer() {
    }

    @Override
    public void initLayer() {
    }

    @Override
    public void cleanLayer() {
    }

    @Override
    public boolean userControlsOpacity() {
        return false;
    }
}
```

Console layer widget extends ConsoleLayer

Plug-in example - Map Layer

```
…
public class MyConsoleLayer extends ConsoleLayer {

    @NeptusProperty(name = "Show Time", userLevel = LEVEL.REGULAR,
            category="Visibility", editable = true)
    public boolean showTime = true;

    public MyConsoleLayer() {
    }
…
```

Adding properties for the operator to change



Console Settings

- Audio Vehicle State Alerts
- Command Planner
- General Preferences
- IMU Alignment
- LBL Ranges
- Log Download
- My Console Layer
- MyLocationDisplay
- Plan Edition
- Rhodamine Oil Visualizer
- Systems List

⊟ Visibility

Show Time ☑

Show Time
(Default value: true)

□ Advanced Save Cancel

LAUV-NOPTILUS-2: SERVICE

Plan Control

State: BLOCKED
Plan: go_pl_8uotui
Man.:
Outcome: N/A

N41.184690°, W8.706085°
⊟ Transponders
- benthos2
- benthos4

⊟ Plans
- dislodge
- go_pl_8uotui
- plan1

41N11.154 / 8W42.515

System: lauv-noptilus-2 Plan: go_pl_8uotui

23:03 UTC 42 Notifications

Plug-in example - Map Layer

```
…
public class MyConsoleLayer extends ConsoleLayer implements MainVehicleChangeListener {

    @NeptusProperty(name = "Show Time", userLevel = LEVEL.REGULAR,
            category="Visibility", editable = true)
    public boolean showTime = true;

    private LocationType location = null;
    private String positionStr = null;
    private String dateTimeStr = null;

    public MyConsoleLayer() {
    }
…
    @Override
    public void mainVehicleChange(String id) {
        ImcSystem sys = ImcSystemsHolder.getSystemWithName(getConsole().getMainSystem());
        if (sys != null && sys.getLocation() != null) {
            LocationType loc = new LocationType(sys.getLocation());
            loc.convertToAbsoluteLatLonDepth();
            positionStr = I18n.text("Position:") + " " + loc.getLatitudeAsPrettyString() +
                    " " + loc.getLongitudeAsPrettyString();
            dateTimeStr = I18n.text("Age:") + " " +
                    DateTimeUtil.dateFormaterXMLNoMillisUTC.format(new Date(sys.getLocationTimeMillis()));

            location = loc;
        }
        else {
            positionStr = I18n.text("Position:") + " ?";
            dateTimeStr = I18n.text("Age:") + " ?";
            location = null;
        }
    }
```

Adding main vehicle change listener

Plug-in example - Map Layer

```
…

    @Subscribe
    public void on(EstimatedState msg) {
        if (!msg.getSourceName().equals(getConsole().getMainSystem()))
            return;

        LocationType loc = new LocationType();
        loc.setLatitudeRads(msg.getLat());
        loc.setLongitudeRads(msg.getLon());
        loc.setOffsetNorth(msg.getX());
        loc.setOffsetEast(msg.getY());
        loc.convertToAbsoluteLatLonDepth();
        positionStr = I18n.text("Position:") + " " + loc.getLatitudeAsPrettyString() +
                " " + loc.getLongitudeAsPrettyString();
        dateTimeStr = I18n.text("Age:") + " " +
                DateTimeUtil.dateFormaterXMLNoMillisUTC.format(new Date(msg.getTimestampMillis()));

        location = loc;
    }

 …
```

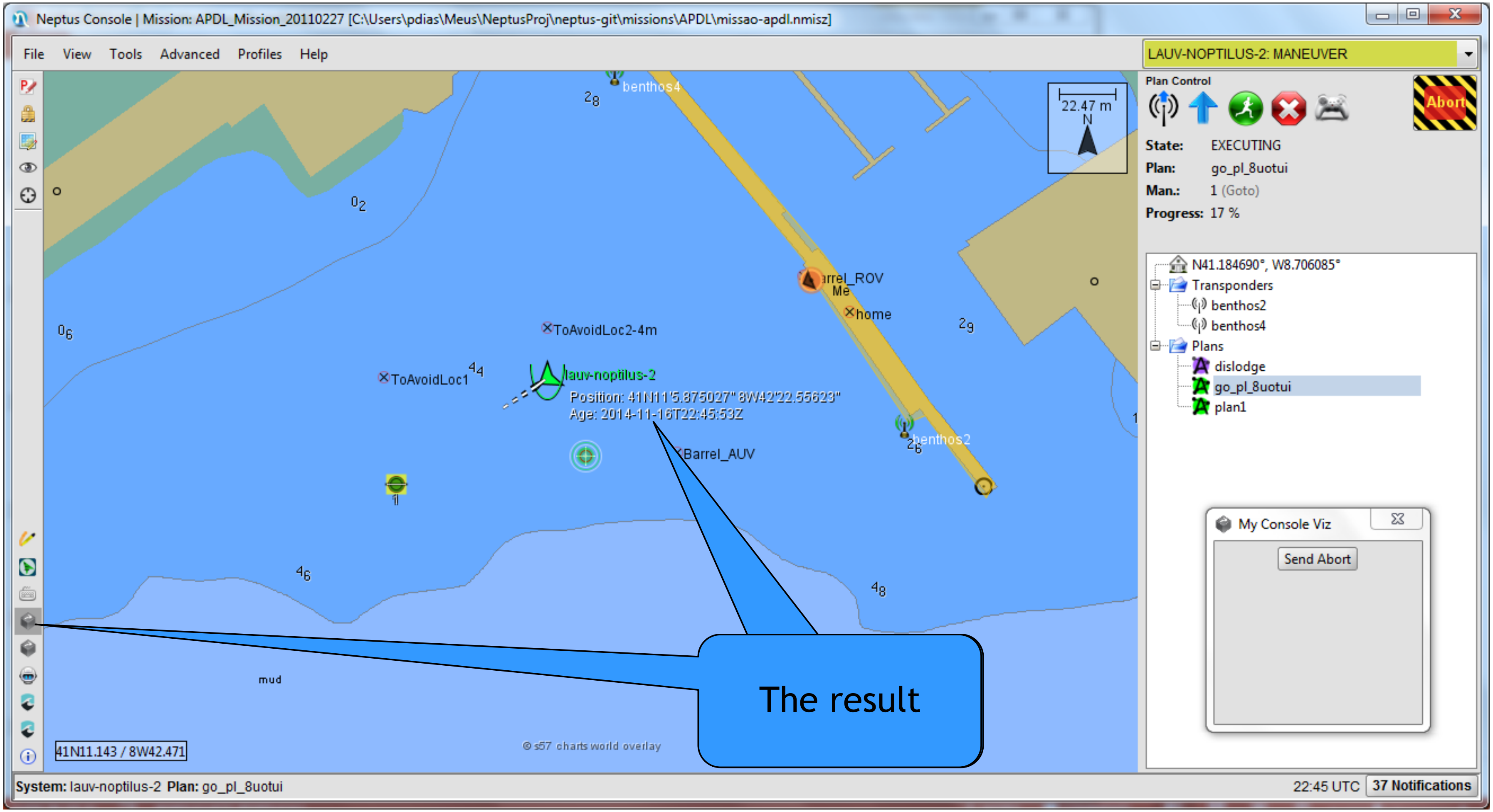
Subscribing to messages

Plug-in example - Map Layer

```
…

    @Override
    public void paint(Graphics2D g, StateRenderer2D renderer) {
        super.paint(g, renderer);

        if (location == null)
            return;

        Graphics2D g2 = (Graphics2D) g.create();

        Point2D pt = renderer.getScreenPosition(location);
        g2.translate(pt.getX(), pt.getY());
        g2.translate(20, 20);
        g2.setColor(Color.BLACK);
        g2.drawString(positionStr, 1, 1);
        g2.setColor(Color.WHITE);
        g2.drawString(positionStr, 0, 0);

        if (showTime) {
            g2.setColor(Color.BLACK);
            g2.drawString(dateTimeStr, 1, 16);
            g2.setColor(Color.WHITE);
            g2.drawString(dateTimeStr, 0, 15);
        }
        g2.dispose();
    }
…
```

Finaly painting the data

Plugin example - Map Layer



Neptus Console | Mission: APDL_Mission_20110227 [C:\Users\pdias\Meus\NeptusProj\neptus-git\missions\APDL\missao-apdl.nmisz]

LAUV-NOPTILUS-2: MANEUVER

State: EXECUTING
Plan: go_pl_8uotui
Man.: 1 (Goto)
Progress: 17 %

lauv-noptilus-2
Position: 41N11'5.875027" 8W42'22.55623"
Age: 2014-11-16T22:45:53Z

System: lauv-noptilus-2 Plan: go_pl_8uotui

22:45 UTC

37 Notifications

The result

Packaging the plug-in

- By using Ant you can compile all
  - ant
- It will create a jar in plugins folder name my-plugin.jar
- To add to console to test
  - Run pt.lsts.neptus.loader.NeptusMain
  - Open lauv.ncon console
  - Click menu View>Plugin Manager add your plugins elements and save console

Become a commiter

Try it out