

# Neptus:

## Command and Control Infrastructure for Networked Vehicle Systems

<neptus@googlegroups.com>



**FEUP**  
Universidade do Porto  
Faculdade de Engenharia

**U. PORTO**

AUV2012

# A little bit of history

- Development for Neptus started in 2005
  - Common software for ROV and AUV vehicles
  - Mission files stored as XML and converted to specific formats using XSLT
- LSTS acquired and / or developed several new vehicles
  - Neptus evolved for supporting UAVs, ASVs and WSNs
  - Added support for STANAG4586 protocol (for UAVs)
- LSTS created its own robotics software framework
  - Extensive support for DUNE and IMC
  - Open source as of 2012 (currently with 5 active developers)

# 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
  - A set of checklists
- The mission is usually...
  - Created prior to execution (planning)
  - Changed during execution (monitoring / revision / re-planning)



# Part 1: Using Neptus

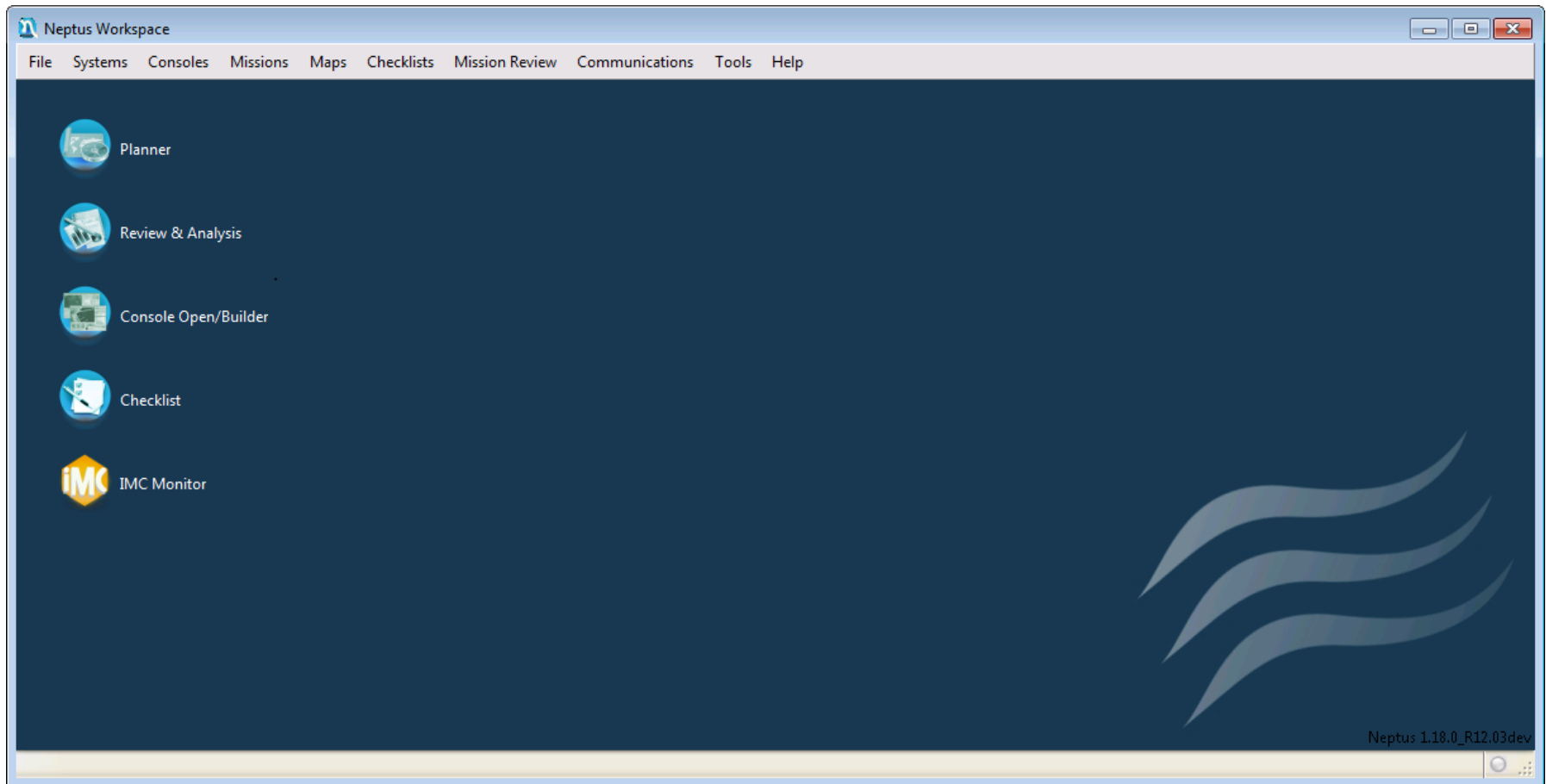
# Neptus Requirements

- 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

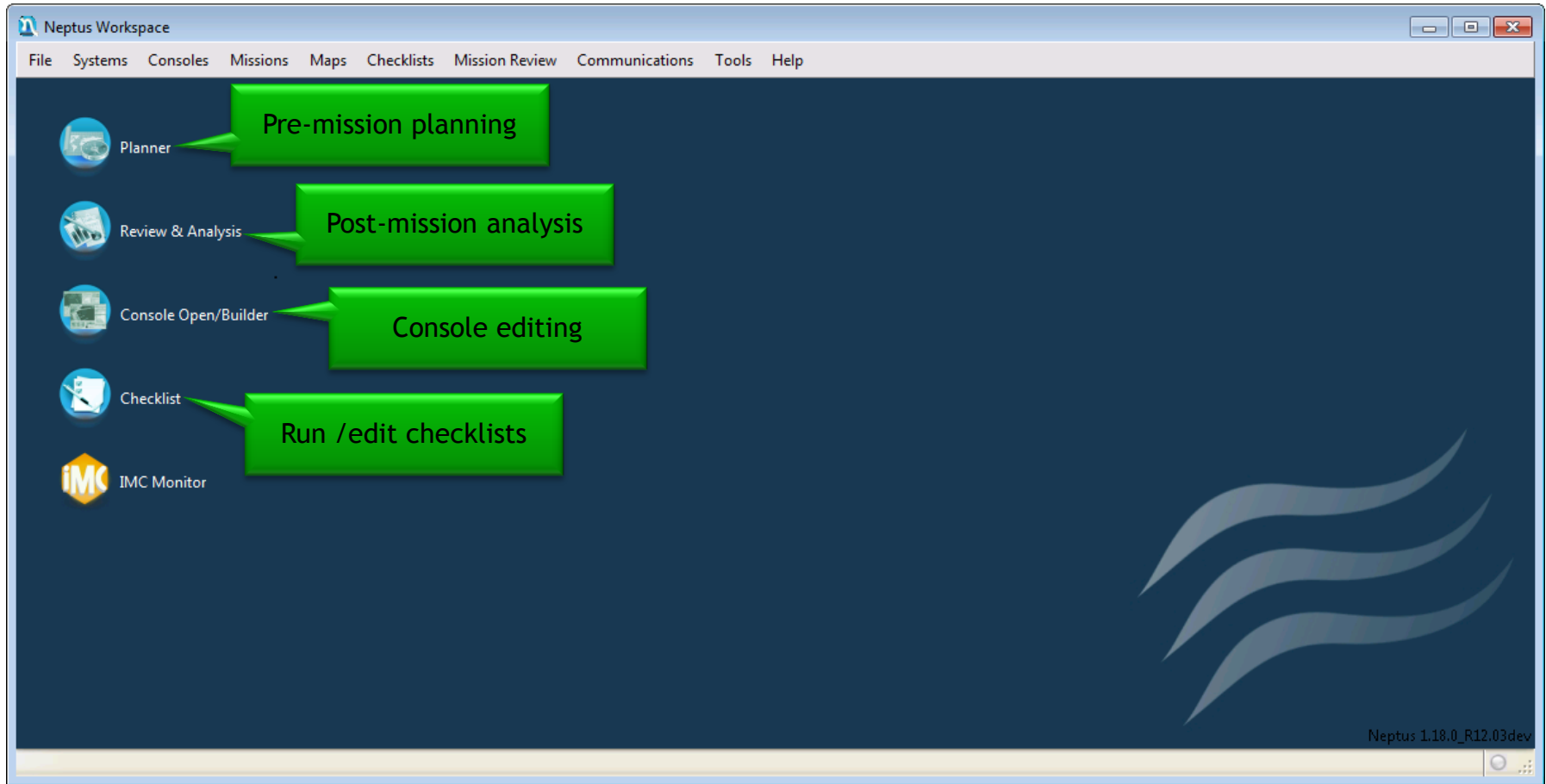
- To install Neptus, just download the latest version to a directory of choice
  - Logs will be put under this directory so make sure you leave extra room for them
- Downloading Neptus
  - Use your favorite SVN client to checkout Neptus from [https://whale.fe.up.pt/svn/neptus/public/\\*/exec](https://whale.fe.up.pt/svn/neptus/public/*/exec)
- Running Neptus
  - In Windows: run neptus.exe
  - In Linux: execute ./neptus.sh

# The Neptus Workspace





# The Neptus Workspace



# The Neptus Workspace

The screenshot displays the Neptus Workspace application window. The main menu includes File, Systems, Consoles, Missions, Maps, Checklists, Mission Review, Communications, Tools, and Help. On the left sidebar, there are icons for Planning, Review, Consoles, Checklists, and IMC Mission Manager. The central area is divided into two main panels:

- LauvAPDL - Checklist:** This panel shows a checklist titled "LauvAPDL - Checklist". It includes a "Save" button, a "Save as" button, a "To PDF" button, a "Run" button, and a "Cancel" button. Below these buttons, the "Name" is "LauvAPDL" and the "Version" is empty. The "Description" field contains:  
PT - Lista material necessário para uma missão com o Lauv na APDL  
EN - Checklist to run APDL missions with the LAUV vehicle.  
At the bottom, there is a list of items with checkboxes and dropdown arrows:
  - Material Separado (0/20)
  - Mala PC Portátil (cinza) (0/4)
  - Mala PELI PC Portátil (preta) (0/3)
- lauv-seacon-1 - Vehicle Info:** This panel displays information for a specific vehicle. It includes a 3D model of the vehicle and the following details:
  - Id: lauv-seacon-1
  - Name: LAUV-SeaCon-1
  - Model: LAUV
  - Type: AUV
  - L/W/H (m): 1.4 x 0.3 x 0.54Buttons for "3D model", "Properties", and "Consoles" are present. Below this, there is a "File info" section with fields for Type, Href, and Description. At the bottom, the "Inertial Coordinate System" section shows "Offset Angle North" with a value of "0.0 (degrees)".

Two green callout boxes highlight specific features:

- A callout box labeled "Running checklists" points to the "Run" button in the checklist panel.
- A callout box labeled "View / Edit vehicle configurations" points to the "Properties" button in the vehicle info panel.

The version number "1.18.0\_R12.03dev" is visible in the bottom right corner of the application window.

# Neptus - Hands-on exercise

- Download and install Java 7
- Download and install Neptus
- Start Neptus and...
  - Check the default IP for lauv-seacon-2
  - Load the default lauv-seacon-1 console

# 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

The screenshot displays the Neptus Console interface for a mission named 'lauv-seacon-1'. The main window shows a 3D map of a coastal area with a blue cube-shaped object and a vehicle labeled 'lauv-seacon-1' positioned near a 'Base' marked with an 'x'. The vehicle's status is shown in a bottom-right panel: 0% battery, 0.0 V, 47% fuel, and 100% health. The interface includes a menu bar (File, Mission, Settings, View, Tools, Planning, Profiles, Help), a toolbar on the left, and a right-hand panel with tabs for Systems, Operation, Info, State, and Tele-Op. The right panel displays detailed information for 'lauv-seacon-1', including its status (SIMULATION), location (Lat: 43N49.395, Lon: 15E34.290), height (-0.1 m), course (000°), heading (006°), ground speed (0.0 m/s), vertical speed (0.0 m/s), RPM (0), IMC ID (00:15), and IP (127.0.0.1@udp:6002). A large red 'ABORT' button is visible in the bottom right corner, along with 'Service Mode', 'Alarms', and 'Messages' buttons. The bottom status bar shows the time as 07h34m UTC.

# Neptus Consoles

The screenshot displays the Neptus Console interface for a mission named "lauv-seacon-1". The main window features a map with a blue highlighted area representing the selected vehicle's position. A green callout box labeled "Map interaction modes" points to the left sidebar containing various map tools. Another green callout box labeled "Selected vehicle (controlled)" points to the vehicle icon on the map. A third green callout box labeled "Systems listing and selection" points to the right-hand panel, which lists two vehicles: "lauv-seacon-1" and "lauv-seacon-2". The "lauv-seacon-1" panel shows detailed simulation data: Lat: 43N49.395, Lon: 15E34.290, Height: -0.1 m, Course: 000°, Heading: 006°, Ground Speed: 0.0 m/s, Vertical Speed: 0.0 m/s, RPM: 0, IMC ID: 00:15, and IP: 127.0.0.1@udp:6002. Below this, a red "ABORT" button is highlighted with a green callout box labeled "Send abort request (wi-fi / acoustic)". The bottom status bar shows "Main Vehicle: lauv-seacon-1" and "07h34m UTC".

# Neptus Consoles

The screenshot displays the Neptus Console interface for a vehicle named 'lauv-seacon-1'. The main window shows a 3D map with a blue and red rectangular area representing the vehicle's operational limits. A green callout box points to this area, stating 'Operational Limits map interaction mode'. Another green callout box points to a sub-system state, stating 'Sub-system states (DUNE entities)'. A third green callout box points to the error log, stating 'Latest error description'. The error log shows a 'Last error: 08h33m55s - Navigation: waiting for the solution to converge'. A yellow and black hazard sign with the word 'ABORT' is visible, along with a 'Service Mode' button. A green callout box points to the 'ABORT' sign, stating 'Green led means everything is fine'. The bottom status bar shows 'Main Vehicle: lauv-seacon-1' and '07h42m UTC'. The interface includes a menu bar (File, Mission, Settings, View, Tools, Planning, Profiles, Help) and a toolbar on the left. The right panel contains tabs for Systems, Operation, Info, State, and Tele-Op, and a table of system entities.

Entity	State	x	Description	Δt
Operational Li...	NORM...	<input type="checkbox"/>	active	0.0 s
Plan Generator	NORM...	<input type="checkbox"/>	active	0.0 s
LBL	NORM...	<input type="checkbox"/>	expecting beac...	0.0 s
Leak Sensor - N...	NORM...	<input type="checkbox"/>	active	0.0 s
Leak Sensor - Tail	NORM...	<input type="checkbox"/>	active	0.0 s
Depth Sensor	NORM...	<input type="checkbox"/>	active	0.0 s
AHRS	NORM...	<input type="checkbox"/>	active	0.0 s
Motor	NORM...	<input type="checkbox"/>	active	0.0 s
Navigation	NORM...	<input type="checkbox"/>	maximum hori...	0.0 s
Echo Sounder	NORM...	<input type="checkbox"/>	active	0.0 s
DVL	NORM...	<input type="checkbox"/>	active	0.0 s

**Error State**  
Error entities:  
  
Last error:  
08h33m55s - Navigation: waiting for the solution to converge

**Vehicle: lauv-seacon-1**  
0% 0.0 V 47% 100%

**Green led means everything is fine**

# Neptus Consoles

The screenshot displays the Neptus Console interface for a mission named "lauv-seacon-1". The interface is divided into several key sections:

- Planning map interaction mode:** A large map on the left shows a mission plan with waypoints (e.g., 1,10, 2,9, 3,10, 4,11, 5,4, 6,13, 7,14, 8,15) and a "Goto1" maneuver. A scale bar indicates 21.54 m.
- Plan control:** A panel on the right titled "Goto1" provides details for the selected maneuver:

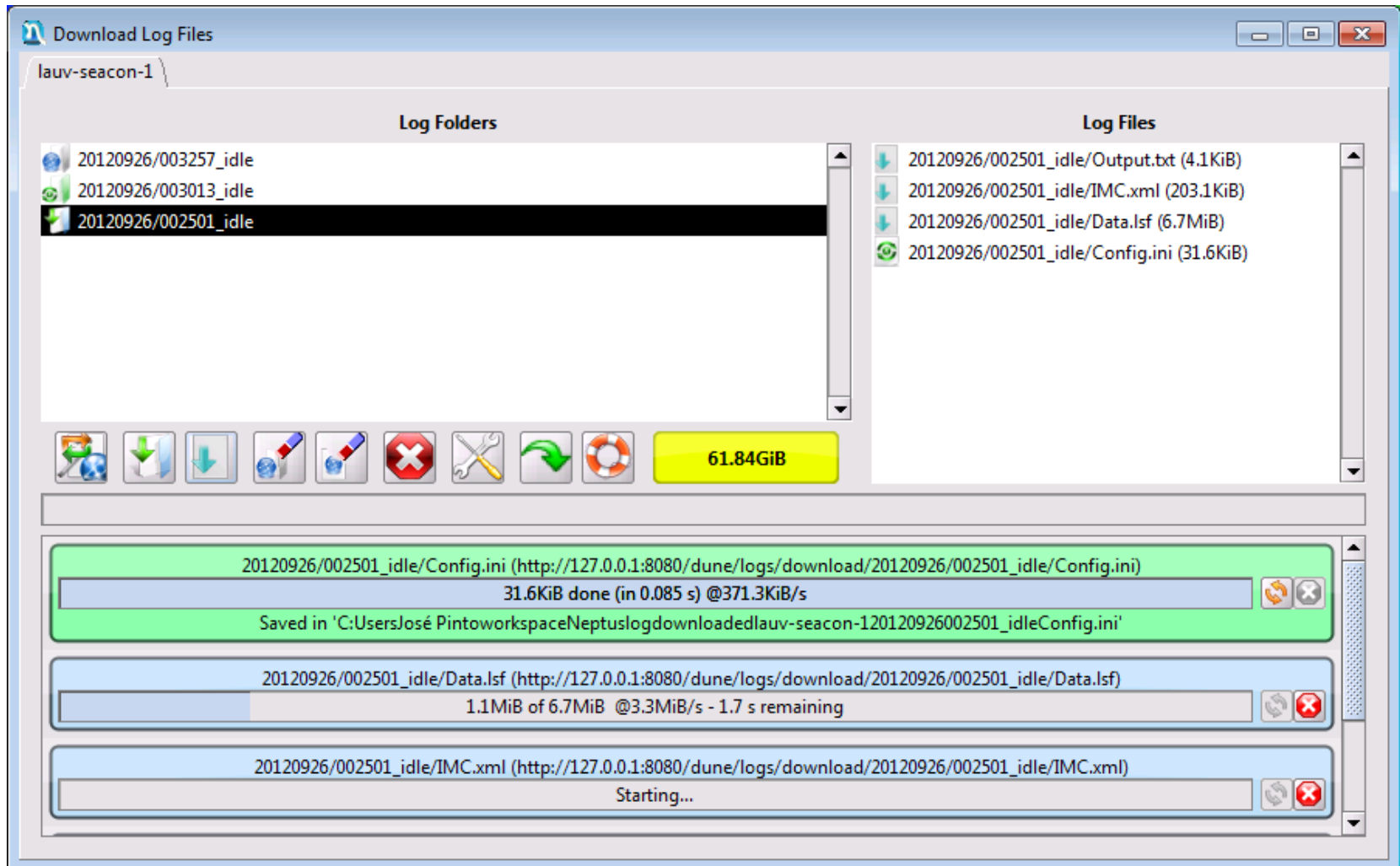
ID	Goto1
Minimum time	0
Maximum time	10.000
Initial Maneuver	<input checked="" type="checkbox"/>
Speed	900
Speed units	RPM
Location	N43.823819°, E15.5...
start-actions	Plan Actions:
end-actions	Plan Actions:
- Mission elements:** A tree view on the right lists mission components:
  - Transponders
    - b1
    - b2
  - Plans
    - regression\_test [lauv-xtreme-2]
    - rows [lauv-xtreme-2]
    - line [lauv-seacon-1]
- Plan edition controls:** Buttons for "Delete" and "Edit" are visible below the maneuver details.
- Vehicle status:** A bottom status bar shows "Vehicle: lauv-seacon-1" with battery (0%), voltage (0.0 V), and other indicators.
- Operational Limits:** A large "ABORT" button is prominently displayed.
- Service Mode:** Includes "Alarms" and "Messages" buttons.



# Neptus Consoles - hands-on exercise

1. Make sure you have Neptus correctly installed in your system
2. Open the seacon-light.ncon Neptus console
3. Set lauv-seacon-3 as the main vehicle in the console
4. Create a new plan for lauv-seacon-3
  1. Sequence of 4 Goto maneuvers, forming a square of 60x60 meters
  2. Set the plan depth to 5 meters and speed to 1200 RPM
  3. Save the plan as plan\_<yourname>
  4. Send it to the the simulated vehicle

# Neptus Consoles - log download dialog



# Neptus Consoles - log download dialog

Download Log Files

laur-seacon-1

Remote log folder

Log Folders

- 20120926/002501\_idle
- 20120926/003013\_idle
- 20120926/002501\_idle (selected)

Log folder already downloaded

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)

Log files contained in the selected log folder

Synchronize log listing with the remote vehicle

61.84GiB

Log files being downloaded are shown here

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

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...

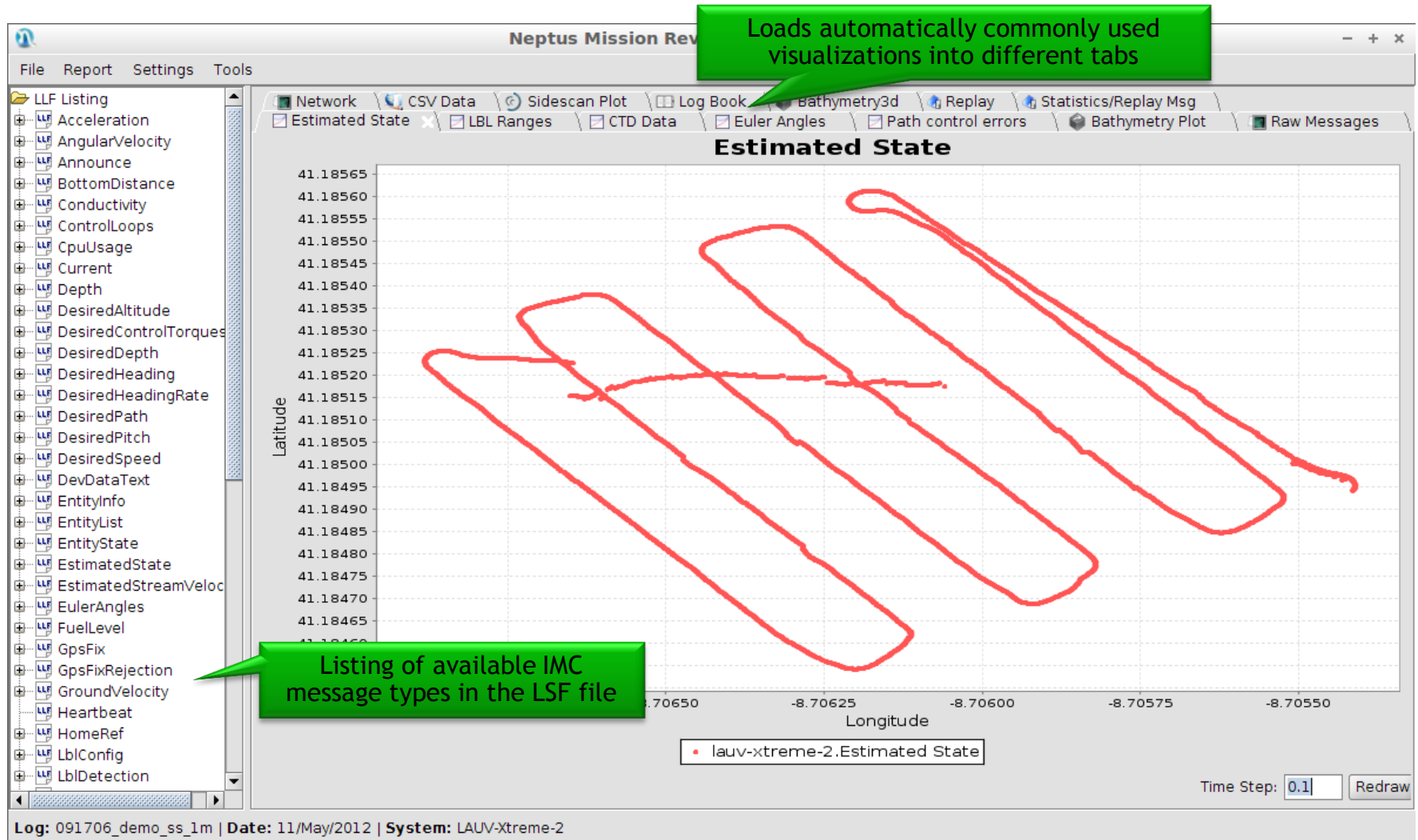
# Neptus Mission Review and Analysis

- Can be accessed
  - 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

- Can be accessed
  - 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



# Neptus Mission Review and Analysis

Neptus Mission Review And Analysis

File Report Settings Tools

Network CSV Data Sidescan Plot Log Book Bathymetry3d Replay Statistics/Replay Msg  
Estimated State LBL Ranges CTD Data Euler Angles Path control errors Bathymetry Plot Raw Messages

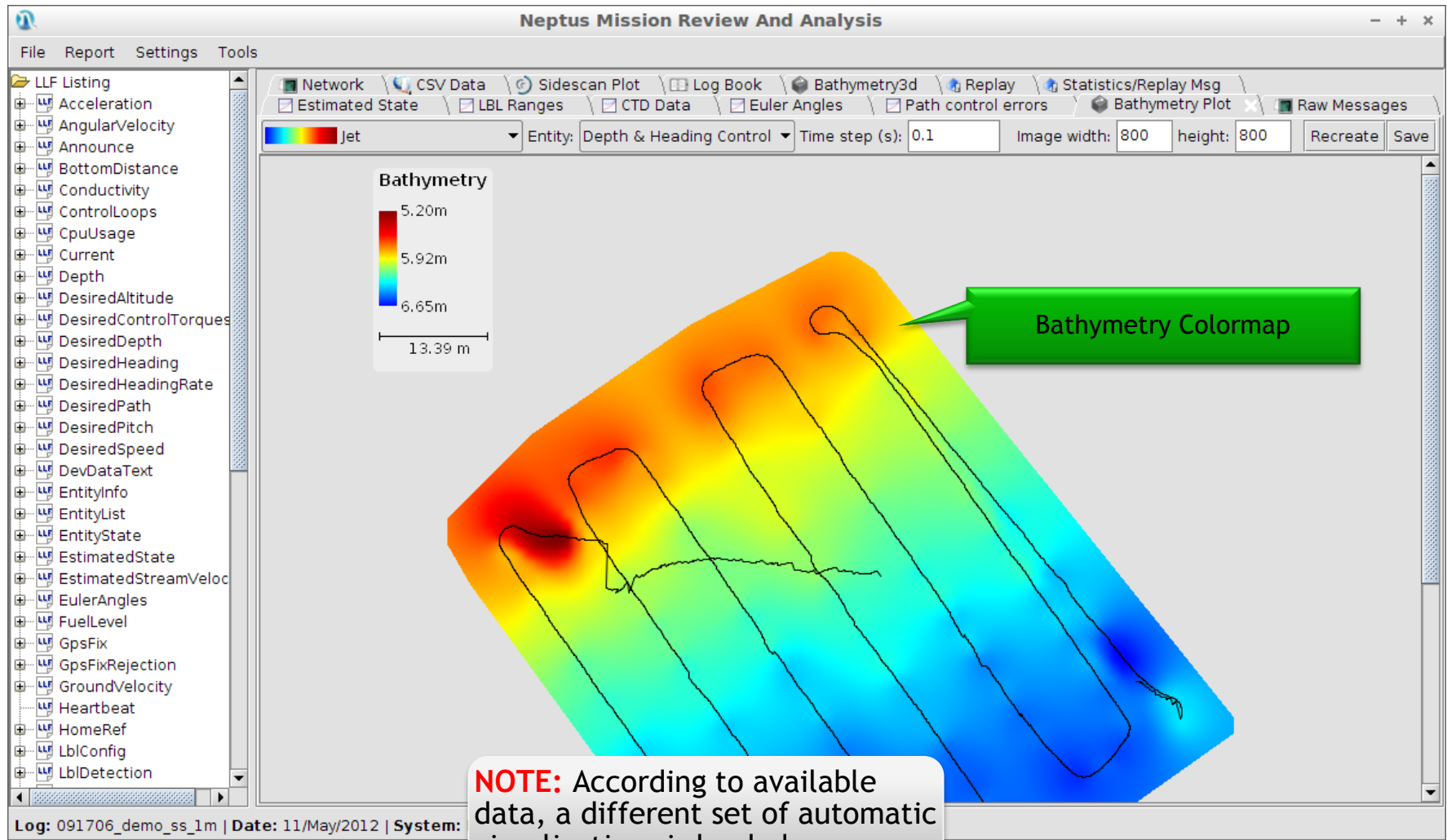
41N11.107 8W42.381  
altitude: 1.0m speed: 2.64m/s  
angle: -16.34°

Sidescan visualization

**NOTE:** According to available data, a different set of automatic visualizations is loaded

Log: 091706\_demo\_ss\_1m | Date: 11/May/2012 | System:

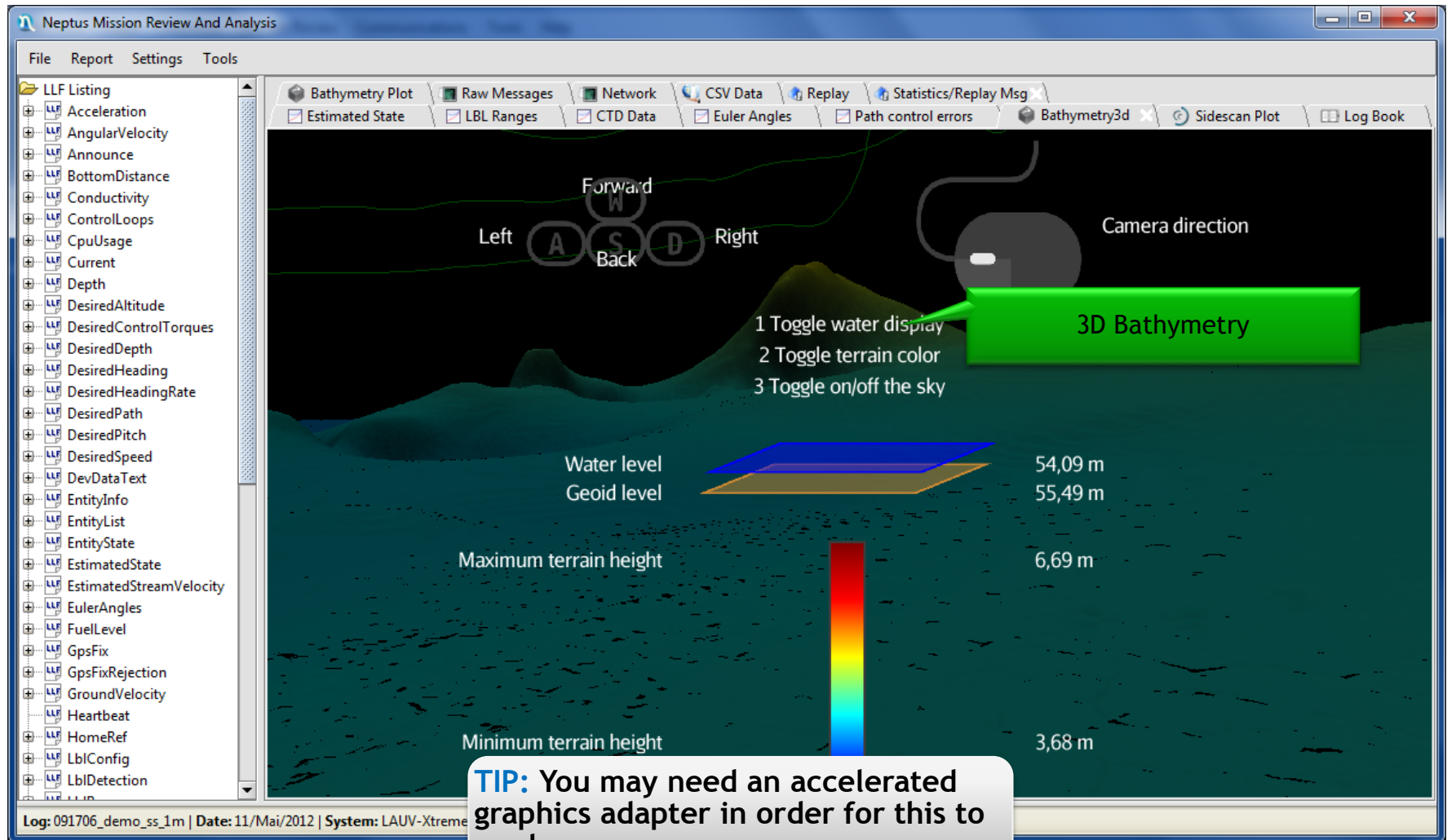
# Neptus Mission Review and Analysis



**NOTE:** According to available data, a different set of automatic visualizations is loaded



# Neptus Mission Review and Analysis



**TIP:** You may need an accelerated graphics adapter in order for this to work

# Neptus Mission Review and Analysis

Neptus Mission Review And Analysis

File Report Settings Tools

LLF Listing

- Acceleration
- AngularVelocity
- Announce
- BottomDistance
- Conductivity
- ControlLoops
- CpuUsage
- Current
- Depth
- DesiredAltitude
- DesiredControlTorques
- DesiredDepth
- DesiredHeading
- DesiredHeadingRate
- DesiredPath
- DesiredPitch
- DesiredSpeed
- DevDataText
- EntityInfo
- EntityList
- EntityState
- EstimatedState
- EstimatedStreamVeloc
- EulerAngles
- FuelLevel
- GpsFix
- GpsFixRejection
- GroundVelocity
- Heartbeat
- HomeRef
- LblConfig
- LblDetection

Network CSV Data Sidescan Plot Log Book Bathymetry3d Replay Statistics/Replay Msg

Estimated State LBL Ranges CTD Data Euler Angles Path control errors Bathymetry Plot Raw Messages

09:18:38.423 (4.0x)

11.24 m

Mission replays can be done in real-time and faster speeds

Many replay layers are available including GPS fixes, LBL fixes, Sidescan, etc

41N11.118 / 8W42.369

Log: 091706\_demo\_ss\_1m | Date: 11/May/2012 | System: LAUV-Xtreme-2

# Neptus Mission Review and Analysis

Neptus Mission Review And Analysis

File Report Settings Tools

LLF Listing

Acceleration

AngularVelocity

Announce

BottomDistance

Conductivity

ControlLoops

CpuUsage

Current

Depth

DesiredAltitude

DesiredControlTorques

DesiredDepth

DesiredHeading

DesiredHeadingRate

DesiredPath

DesiredPitch

DesiredSpeed

DevDataText

EntityInfo

EntityList

EntityState

EstimatedState

ref

x

y

z

phi

theta

Announce[10:27:10.555]

EstimatedState

EstimatedState[10:17:06.773]

Acceleration

CSV Data

Sidescan Plot

Log Book

Bathymetry3d

Replay

Statistics/Replay Msg

Announce

LBL Ranges

CTD Data

Euler Angles

Path control errors

Bathymetry Plot

Raw Messages

src	src_ent	dst	dst_ent	time	x	y	z
lauv-xtreme-2	IMU	*	*	2282.622	0.1350282645783387 m/s/s	-0.168727924280148 m/s/s	-9.614769583794475
lauv-xtreme-2	IMU	*	*	2282.702	1.0805283509762957 m/s/s	-0.17680101182693614 m/s/s	-9.654138886630534
lauv-xtreme-2	IMU	*	*	2282.722	1.0209555239472537 m/s/s	-0.17084599414635449 m/s/s	-9.654141224718094
lauv-xtreme-2	IMU	*	*	2282.742	0.09153246552995406 m/s/s	-0.1743163554043509 m/s/s	-9.668462011015414
lauv-xtreme-2	IMU	*	*	2282.762	0.0988733115457464 m/s/s	-0.17192451009806245 m/s/s	-9.696996031582355
lauv-xtreme-2	IMU	*	*	2282.772	0.07770461384872905 m/s/s	-0.13855436547840944 m/s/s	-9.737433255910872
lauv-xtreme-2	IMU	*	*	2282.782	0.08033037750874646 m/s/s	-0.12426957294307649 m/s/s	-9.775472186926006
lauv-xtreme-2	IMU	*	*	2282.782	0.06870661174305714 m/s/s	-0.1099115051089786 m/s/s	-9.81354969638586 r
lauv-xtreme-2	IMU	*	*	2282.802	0.057100148738618 m/s/s	-0.10622236826745793 m/s/s	-9.849274505239725
lauv-xtreme-2	IMU	*	*	2282.802	0.06438640132322906 m/s/s	-0.08486975623774341 m/s/s	-9.878953019666671
lauv-xtreme-2	IMU	*	*	2282.822	0.05641037180959247 m/s/s	-0.07643735212557949 m/s/s	-9.925366395795345
lauv-xtreme-2	IMU	*	*	2282.842	0.06140167272428516 m/s/s	-0.07046882650945335 m/s/s	-9.958658424544334
lauv-xtreme-2	IMU	*	*	2282.862	0.048666497950861226 m/s/s	-0.05371284438788425 m/s/s	-10.00744495954513
lauv-xtreme-2	IMU	*	*	2282.882	0.03388330003987066 m/s/s	-0.040528373210690916 m/s/s	-10.04315573987364
lauv-xtreme-2	IMU	*	*	2282.902	0.0460873773731100991 m/s/s	-0.027457771920843515 m/s/s	-10.09068905994296
lauv-xtreme-2	IMU	*	*	2282.922	0.04140026980754919 m/s/s	-0.023859094408329107 m/s/s	-10.10021325961351
lauv-xtreme-2	IMU	*	*	2282.942	0.051053301755851134 m/s/s	-0.0250537133036647 m/s/s	-10.11923126381635
lauv-xtreme-2	IMU	*	*	2282.962	0.04994726760955527 m/s/s	-0.013166864738555158 m/s/s	-10.13467433214187
lauv-xtreme-2	IMU	*	*	2282.982	0.05113740897588431 m/s/s	-0.008428501401070388 m/s/s	-10.13703930770754
lauv-xtreme-2	IMU	*	*	2283.002	0.04644901369763538 m/s/s	0.009401142400575917 m/s/s	-10.15247653081417
lauv-xtreme-2	IMU	*	*	2283.022	0.0580875112415757 m/s/s	0.021137649538205003 m/s/s	-10.12742275357842
lauv-xtreme-2	IMU	*	*	2283.042	0.04723910462940111 m/s/s	0.041311367681389675 m/s/s	-10.11550201416015
lauv-xtreme-2	IMU	*	*	2283.062	0.056630106374341994 m/s/s	0.07843370444141327 m/s/s	-10.11751393850445
lauv-xtreme-2	IMU	*	*	2283.082	0.05146824379886966 m/s/s	0.109185391924666996 m/s/s	-10.07817795341610
lauv-xtreme-2	IMU	*	*	2283.102	0.06054495543024968 m/s/s	0.15409189802119508 m/s/s	-10.04349359352588
lauv-xtreme-2	IMU	*	*	2283.122	0.05652658206778578 m/s/s	0.19787096185721456 m/s/s	-10.00412136808037
lauv-xtreme-2	IMU	*	*	2283.142	0.061211566112819124 m/s/s	0.23800323015609753 m/s/s	-9.94212113124132 r
lauv-xtreme-2	IMU	*	*	2283.162	0.06187240464172701 m/s/s	0.30545840793279927 m/s/s	-9.903864173561335
lauv-xtreme-2	IMU	*	*	2283.182	0.06187240464172701 m/s/s	0.36468032089862973 m/s/s	-9.865663329982757

Log: 091706\_demo\_ss\_1m | Date: 11/May/2012 | System:

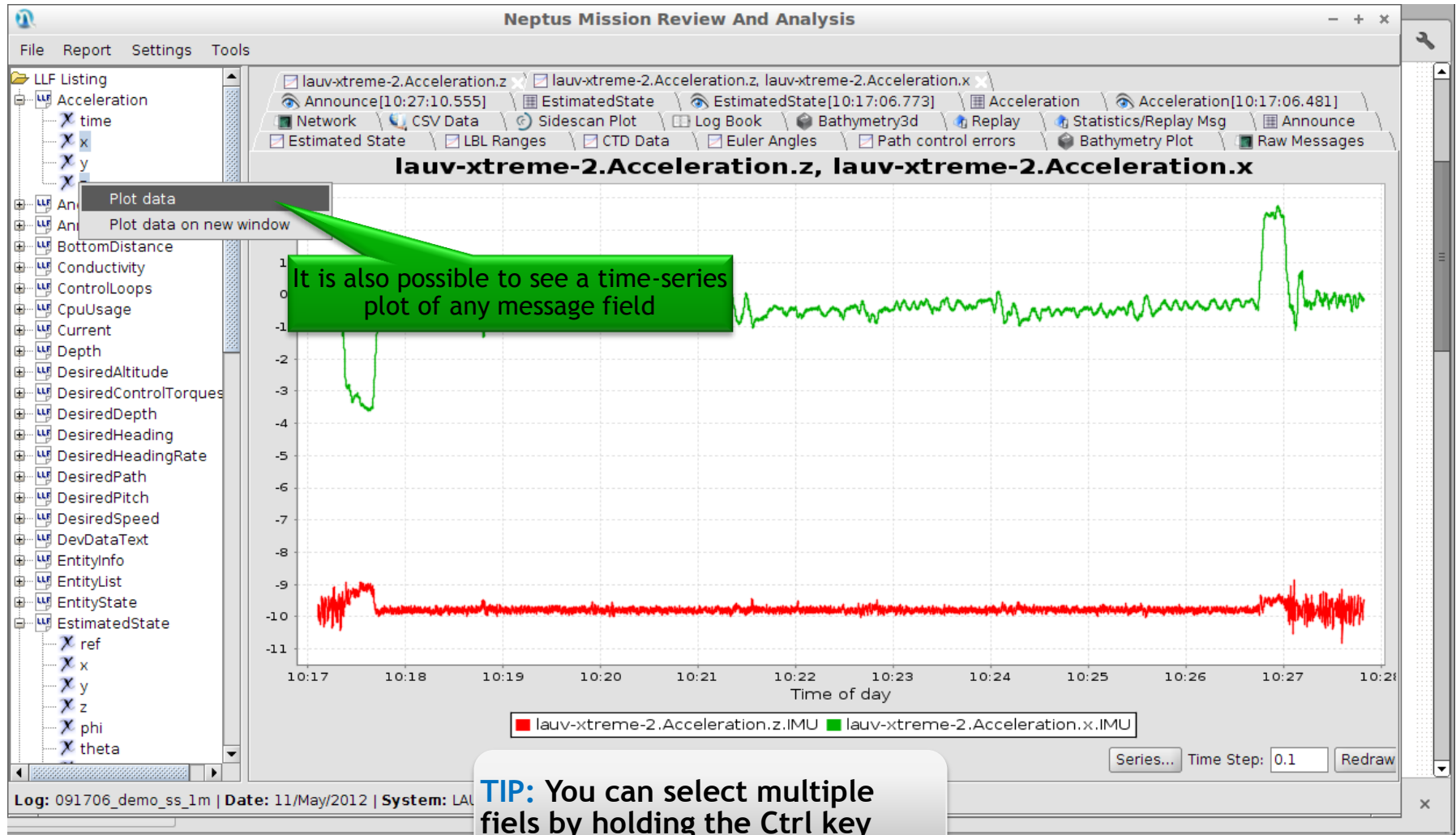
Show Acceleration data

View communications

All individual message data can be seen as a table

TIP: You can more thoroughly inspect a specific message by double clicking it in this mode

# Neptus Mission Review and Analysis



# Neptus MRA - Hands-on exercise

- Download and unpack the log archive tutorial\_log.zip
- Load the log into MRA
- Try to find:
  - Maximum depth of the vehicle
  - Maximum speed
  - Distance between start and end locations



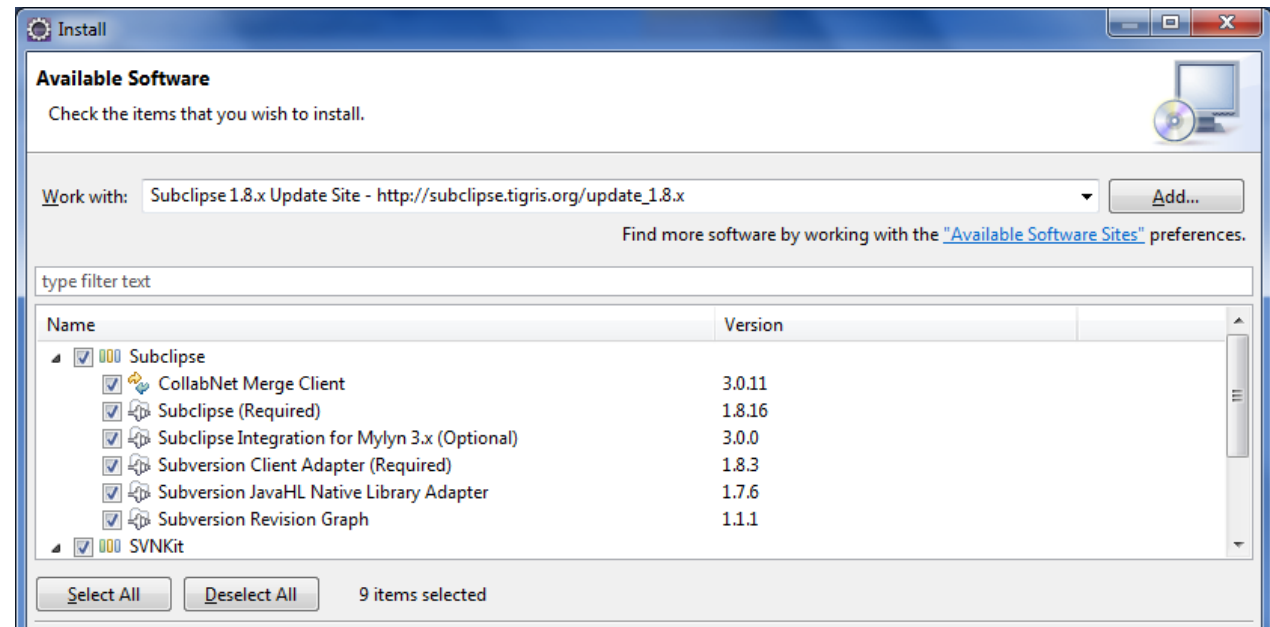
## Part 2: Extending Neptus

# Requirements for extending Neptus

- For developing Neptus extensions you need Oracle's Java 7 JDK
  - This package includes the Java virtual machine and compiler
- Neptus should be developed under Eclipse
  - Recommended version is Eclipse IDE for JEE developers
- To commit changed code you need a Subversion client
  - Subclipse Eclipse plug-in is recommended

# Configuring Eclipse - 1

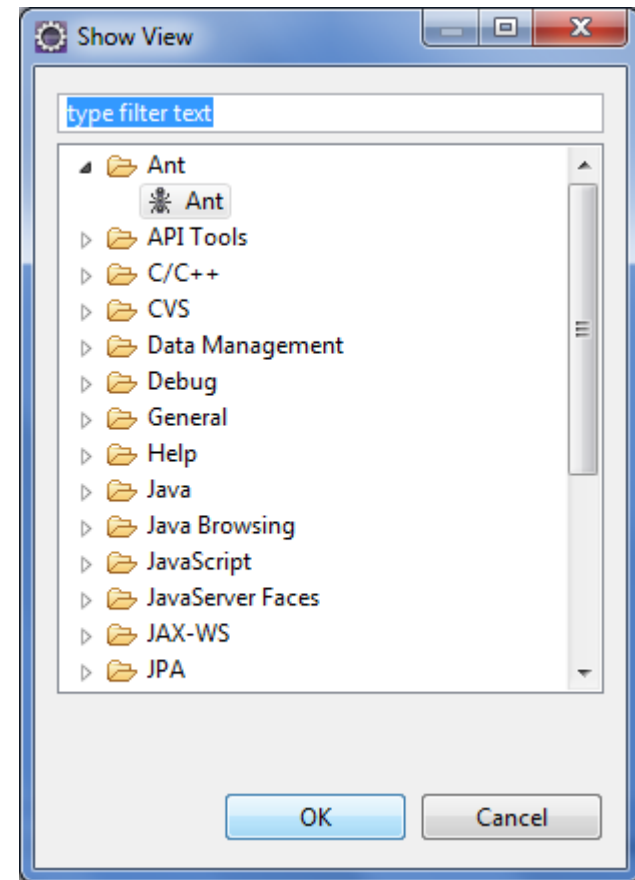
- Install Subclipse plugin (optional) in Eclipse
  - Help > Install new Software
  - Use the update site [http://subclipse.tigris.org/update\\_1.8.x](http://subclipse.tigris.org/update_1.8.x)
  - Instal Subclipse



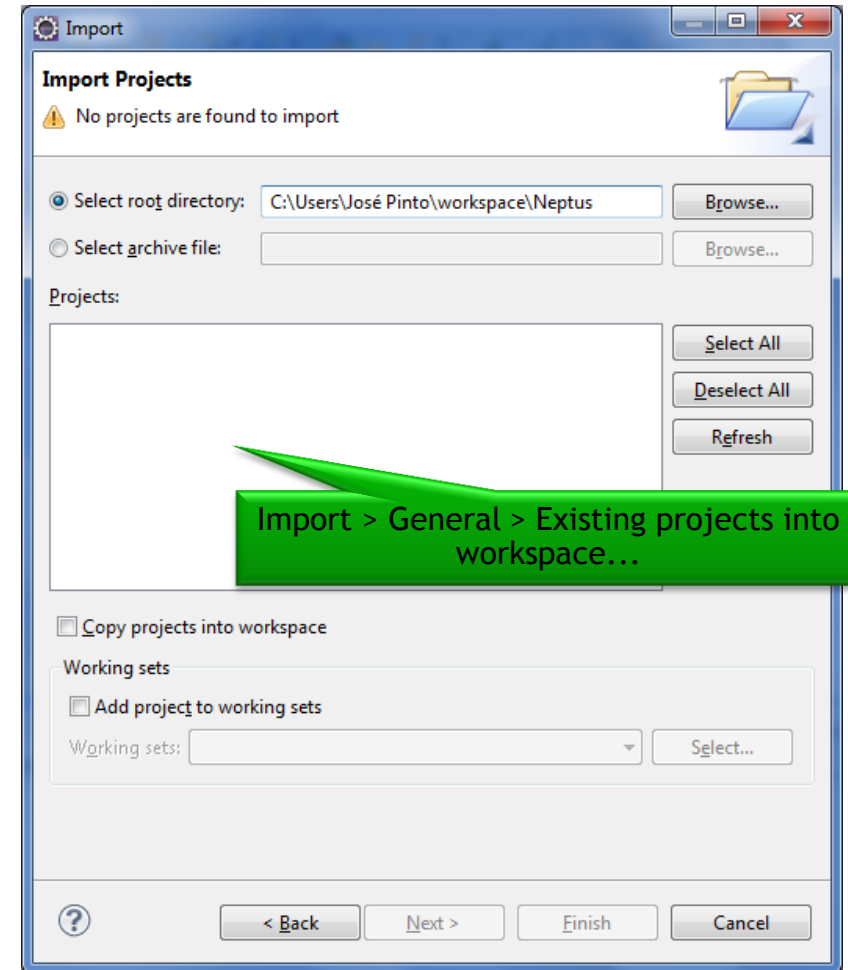
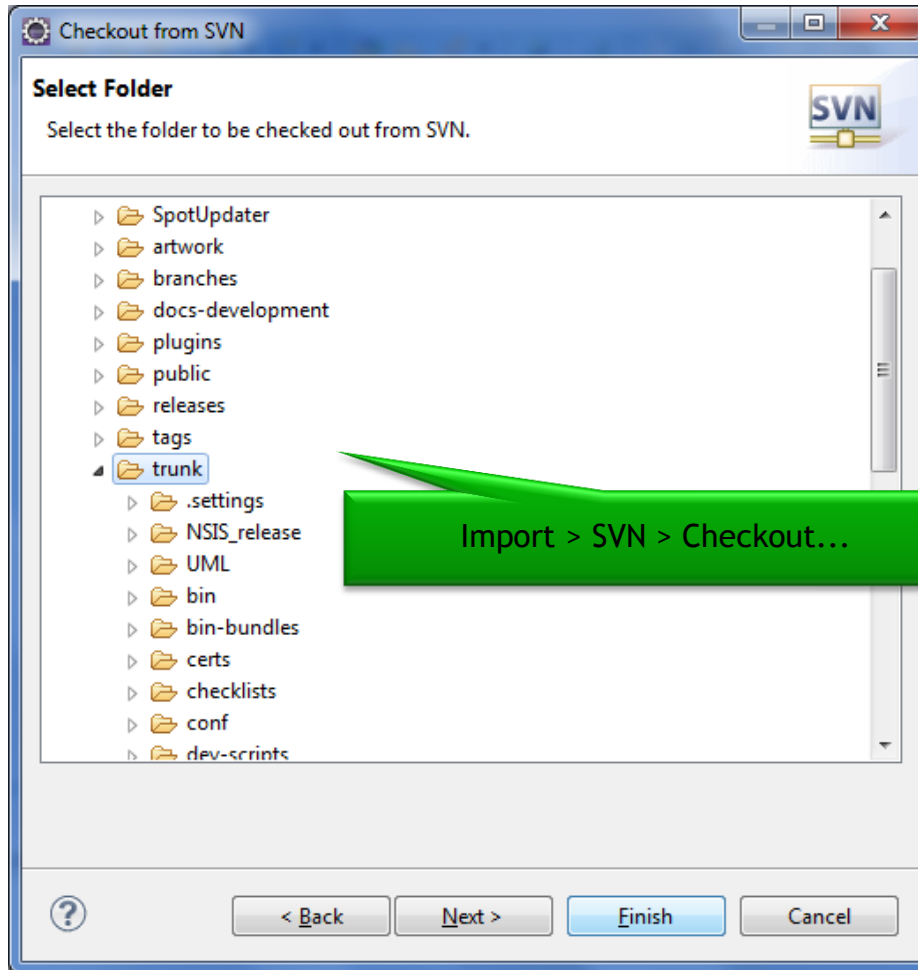


# Configuring Eclipse - 2

- Add the Ant view to the default Java perspective
  - Window > Show View > Other > Ant

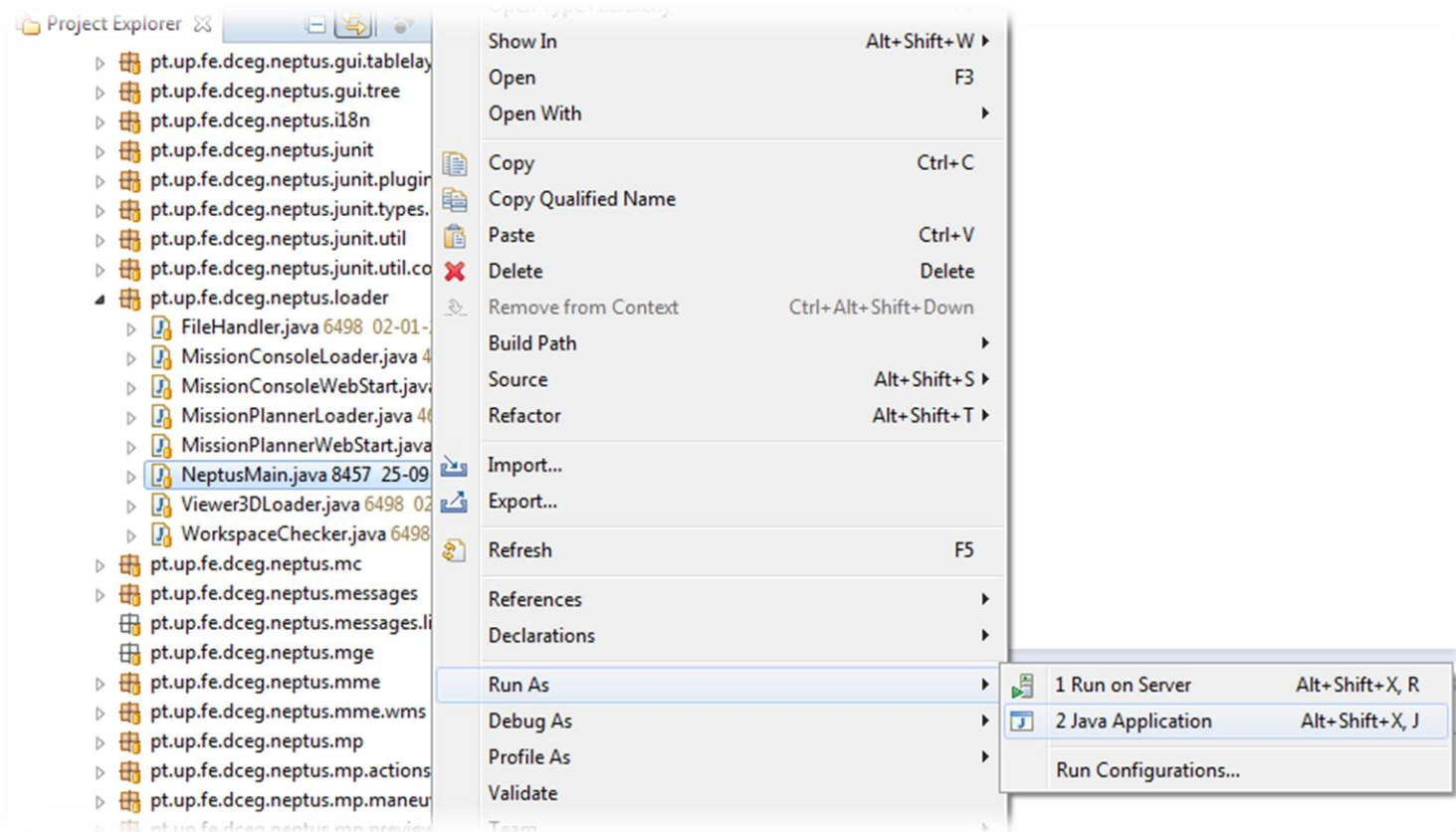


# Import or Checkout Neptus Project



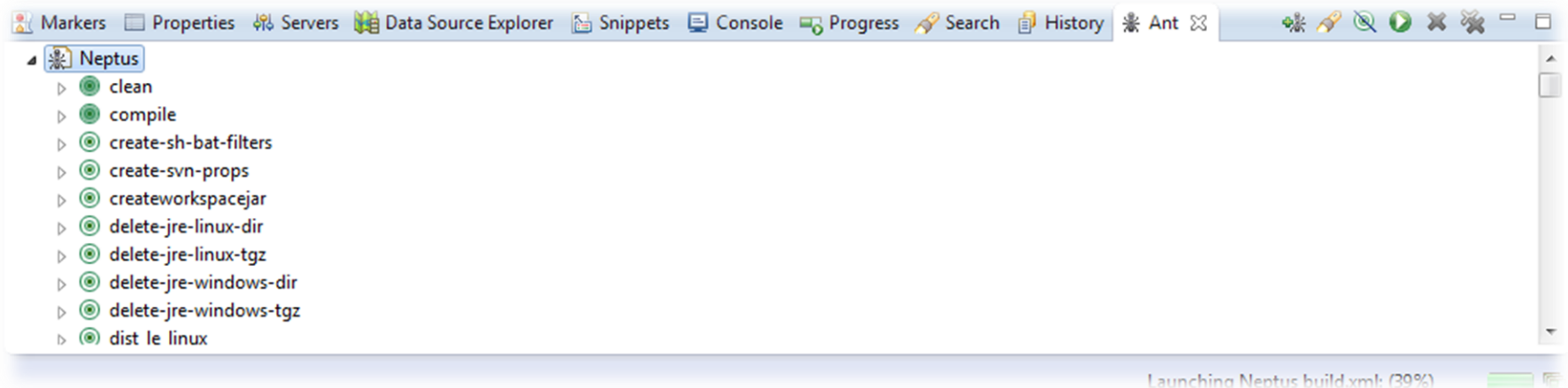
# Testing Neptus

- Main class: `pt.up.fe.dceg.neptus.loader.NeptusMain`
- Right-click it and select **Run As > Java Application**



# Compiling Neptus

- Drag the file Neptus/build.xml into the Ant view
- This will list all the available Ant targets
- The default target compiles and packages Neptus core
- To run any target double click it



# Creating a plug-in

- A Neptus plug-in is a set of Java classes that extend Neptus, packaged into a .jar file
- Plug-ins can be
  - MRA visualizations
  - Console displays
  - Map interactions / layers
  - ...
- Extensions are identified by a special Java Annotation and an entry in the plugins.lst file...
  - They are afterwards inspected for extension points

# Hello World Plug-in (Console display)

- Add a new source folder to Neptus ([plugins-dev/tutorial](#))
- Add a new package in that source folder ([pt.up.fe.dceg.neptus.plugins.tutorial](#))
- Add a new class under this package ([HelloWorld](#))
  - Make it extend the class SimpleSubPanel
  - Add the @PluginDescription annotation to this class

# Hello World plug-in

Java EE - Neptus/plugins-dev/tutorial/plugins.lst - Eclipse

File Edit Navigate Search Project Run Window Help

Quick Access Java EE Debug

Project Explorer

- plugins-dev/trex
- plugins-dev/3D
- plugins-dev/mat
- plugins-dev/ais
- srcTests
- plugins-dev/tutorial
  - pt.up.fe.dceg.neptus.plugins.tutorial
    - HelloWorld.java
    - plugins.lst
- log4j-1.2.16.jar 7583 20-06-2012 14:59 mfaria
- xml-apis.jar 2785 23-08-2008 20:12 pdias
- xercesImpl.jar 716 14-10-2005 14:46 pdias
- dom4j-1.6.1.jar 651 25-08-2005 16:23 pdias
- jaxen-1.1.1.jar 7583 20-06-2012 14:59 mfaria
- serializer.jar (from Class-Path of xalan.jar) 716 14-10-2005 14:46 pdias
- xalan.jar 716 14-10-2005 14:46 pdias
- StarfireExt.jar 161 23-03-2005 12:32 rjpp
- activation.jar 304 28-04-2005 16:06 zepinto
- mail.jar 304 28-04-2005 16:06 zepinto
- soap.jar 304 28-04-2005 16:06 zepinto
- comm.jar 319 10-05-2005 19:43 sfraga
- commons-net-1.4.0.jar 417 31-05-2005 15:10 pdias
- wms.jar 471 17-06-2005 16:20 zepinto
- l2fprod-common-all.jar 2037 28-03-2007 20:29 zepinto
- avalon-framework-4.2.0.jar 2785 23-08-2008 20:12 pdias
- jimi-1.0.jar (from Class-Path of fop.jar) 837 14-12-2005 19:12 pdias
- fop.jar 2785 23-08-2008 20:12 pdias

plugins.lst

```
# DATE=@DATE@
# SVN_PATH=@SVN_PATH@
pt.up.fe.dceg.neptus.plugins.tutorial.HelloWorld
```

Added source folder

Line in plugins.lst identifying the extensions

Writable Insert 8 : 49

<terminated> HelloWorld [Java Application] C:\Program Files\Java\jdk1.7.0\_07\bin\javaw.exe (26 de

# Hello World plug-in

The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Displays a project structure with a folder named `plugins-dev/tutorial` containing `HelloWorld.java` and `plugins.lst`. A green callout points to the `plugins-dev/tutorial` folder with the text "Added source folder".
- Code Editor:** Shows the content of `plugins.lst` with the following text:

```
# DATE=@DATE@
# SVN_PATH=@SVN_PATH@
pt.up.fe.dceg.neptus.plugins.tutorial.HelloWorld
```

A green callout points to the line `pt.up.fe.dceg.neptus.plugins.tutorial.HelloWorld` with the text "Line in plugins.lst identifying the extensions".
- Editor:** Shows the content of `HelloWorld.java` with the following text:

```
FILED BY@
_COPY@
```
- Taskbar:** Shows the application name `<terminated> HelloWorld [Java Application] C:\Program Files\Java\jdk1.7.0_07\bin\javaw.exe (26 de`.



# Hello World plug-in

```
plugins.lst HelloWorld.java
package pt.up.fe.dceg.neptus.plugins.tutorial;

// Eclipse allows automatic import organization (CTRL+SHIFT+O)
import java.awt.BorderLayout;

// This annotation is essential for the plug-in to be discovered
@PluginDescription(name="Hello World Plugin")
public class HelloWorld extends SimpleSubPanel {

    private static final long serialVersionUID = 1L;

    public HelloWorld() {
        // remove default UI
        removeAll();
        setLayout(new BorderLayout());

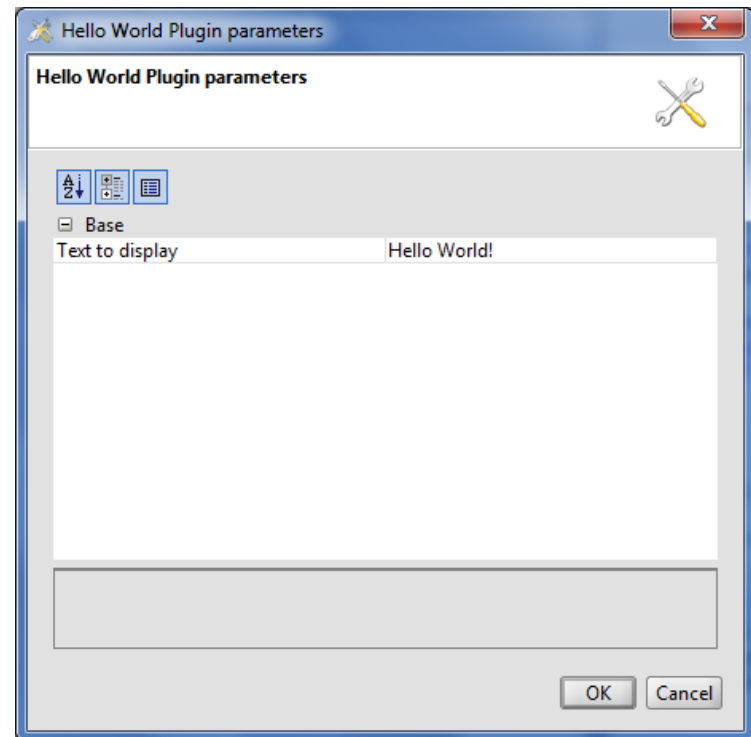
        // add a simple label that displays some text
        add(new JLabel("Hello World!"));
    }

    // Test the Console plug-in directly in an empty console
    public static void main(String[] args) {
        ConsoleParse.testSubPanel(HelloWorld.class);
    }
}
```

# Plug-in properties

- Console edition mode can be entered also by pressing F12
  - Right-click a panel and choose properties to change its parameters

```
package pt.up.fe.dceg.neptus.plugins.tutorial;  
  
// Eclipse allows automatic import organization (CTRL+SHIFT+O)  
import java.awt.BorderLayout;  
  
// This annotation is essential for the plug-in to be discovered  
@PluginDescription(name="Hello World Plugin")  
public class HelloWorld extends SimpleSubPanel {  
  
    private static final long serialVersionUID = 1L;  
  
    @NeptusProperty(name="Text to display")  
    public String text = "Hello World!";  
  
    public HelloWorld() {  
        // remove default UI  
        removeAll();  
        setLayout(new BorderLayout());  
  
        // add a simple label that displays some text  
        add(new JLabel(text));  
    }  
  
    // Test the Console plug-in directly in an empty console  
    public static void main(String[] args) {  
        ConsoleParse.testSubPanel(HelloWorld.class);  
    }  
}
```



# Plug-in properties - detecting changes

```
// This annotation is essential for the plug-in to be discovered
@PluginDescription(name="Hello World Plugin")
public class HelloWorld extends SimpleSubPanel implements ConfigurationListener {

    private static final long serialVersionUID = 1L;

    @NeptusProperty(name="Text to display")
    public String text = "Hello World!";
    protected JLabel lbl = new JLabel(text);

    public HelloWorld() {
        // remove default UI
        removeAll();
        setLayout(new BorderLayout());

        // add a simple label that displays some text
        add(lbl);
    }

    @Override
    public void propertiesChanged() {
        lbl.setText(text);
    }

    // Test the Console plug-in directly in an empty console
    public static void main(String[] args) {
        ConsoleParse.testSubPanel(HelloWorld.class);
    }
}
```

# Console Plug-ins: sending messages

```
@PluginDescription(name="Hello World Plugin")
public class HelloWorld extends SimpleSubPanel {

    private static final long serialVersionUID = 1L;

    protected void doSomething() {
        PlanControl pc = new PlanControl();
        pc.set_type(PlanControl.TYPE.REQUEST);
        pc.set_op(PlanControl.OP.START);
        pc.set_plan_id("my_plan");
        send(getConsole().getMainVehicle(), pc);
    }

    public HelloWorld() {
        // remove default UI
        removeAll();
        setLayout(new BorderLayout());
        JButton btn = new JButton("Do something");
        btn.addActionListener(new ActionListener() {

            @Override
            public void actionPerformed(ActionEvent e) {
                doSomething();
            }
        });
        add(btn);
    }
}
```

# Console Plug-ins: receiving messages

```
@PluginDescription(name="Hello World Plugin")
public class HelloWorld extends SimpleSubPanel implements MessageListener<MessageInfo, IMCMessage> {

    private static final long serialVersionUID = 1L;

    public HelloWorld() {
        Imc3MsgManager.getManager().addListener(this);
    }

    @Override
    public void onMessage(MessageInfo arg0, IMCMessage arg1) {
        if (arg1.getId() == Announce.ID_STATIC) {
            info("Received announce from "+arg1.getString("sys_name"));
        }
    }

    // Test the Console plug-in directly in an empty console
    public static void main(String[] args) {
        ConsoleParse.testSubPanel(HelloWorld.class);
    }
}
```

# Plugin example - Map interaction

```
@PluginDescription(name="Sample Map Interaction")
public class SampleInteraction extends SimpleRendererInteraction {

    private static final long serialVersionUID = 1L;
    protected String text = "inactive";

    @Override
    public boolean isExclusive() {
        return true;
    }

    @Override
    public void paint(Graphics2D g, StateRenderer2D renderer) {
        g.drawString(text, 10, 10);
    }

    @Override
    public void setActive(boolean mode, StateRenderer2D source) {
        if (mode)
            text = "active";
        else
            text = "inactive";
    }
}
```

# Plugin example - Map interaction

```
@PluginDescription(name="Sample Map Interaction")
public class SampleInteraction extends SimpleRendererInteraction {

    private static final long serialVersionUID = 1L;

    @Override
    public boolean isExclusive() {
        return true;
    }

    @Override
    public void mouseClicked(MouseEvent event, StateRenderer2D source) {
        info("mouse clicked at "+source.getRealWorldLocation(event.getPoint()));
    }

    @Override
    public void mouseMoved(MouseEvent event, StateRenderer2D source) {
        //...
    }
}
```

# Plug-in example MRA Visualization

```
@PluginDescription(name="Sample MRA Visualization")
public class SampleMraVis extends SimpleMRAVisualization {

    private static final long serialVersionUID = 1L;

    public boolean canBeApplied(IMraLogGroup source) {
        return source.getLog("EstimatedState") != null;
    };

    @Override
    public JComponent getComponent(IMraLogGroup source, double timestep) {
        IMraLog log = source.getLog("EstimatedState");

        double maxDepth = -Double.MAX_VALUE;

        for (IMCMessage m = log.firstLogEntry(); m != null; m = log.nextLogEntry()) {
            double depth = m.getDouble("z");
            if (depth > maxDepth)
                maxDepth = depth;
        }

        return new JLabel("Max depth is "+maxDepth);
    }
}
```



# Packaging the plug-in

- Run the Ant target named “**plugin-create-jar**”
- Input the name of the folder under **plugins-dev/**
- The jar is created under the **plugins/** folder

# Become a commiter

- Neptus is currently under active development and your help is most welcome
- To become a committer:
  - Send a mail to the mailing list stating why you should become a committer



**If you reached here you are  
already a Neptus developer.**

**Welcome to the team!**