

Structured and Unstructured Grid Relocatable Ocean Platform for Forecasting

SURF-NEMO User Guide

Version 1.02

Francesco Trotta, Nadia Pinardi, Lucio Quadrani Copyright © 2019 - 2023 SURF Team

Table of contents

1. Welcome to the SURF-NEMO User Guide	3
2. Introduction	4
2.1 SURF-NEMO Relocatable ocean modelling platform	4
2.2 Virtual Machine Environment	5
2.3 Source Code	5
3. Getting Started	6
3.1 Download and Install SURF Virtual Machine (VM)	6
3.2 Download and Install SURF packages	14
3.3 Running the Gulf of Taranto Case Study	16
3.4 Make a new experiments	0
3.5 Multiple downscaling experiments	0
4. SURF-NEMO Workflow Overview	0
4.1 Workflow Overview	0
4.2 Model Configuration	0
4.3 Child Meshmask Generation	0
4.4 Input data Regridding	0
4.5 Simulation	0
4.6 Post-processing	0
5. User Configuration File	0
5.1 User Configuration File - Preprocessing Sections	0
5.2 User Configuration File - Postprocessing Sections	0
6. Model Input/Output	0
6.1 Input Datasets for SURF	0
6.2 Output Model Datasets	0
7. Appendix	0
7.1 NEMO Reference Configuration	0
7.2 Disk Partitions and Directory Structure	0
7.3 Linux Root Partition and the installed packages	0
7.4 Release Notes	0
7.5 Bibliography	0

1. Welcome to the SURF-NEMO User Guide

Welcome to the user guide for the structured grid component of the **Structured and Unstructured Grid Relocatable Ocean Platform for Forecasting (SURF)**, release version 1.02. This guide provides comprehensive information and instructions to help you make the most of the SURF-NEMO platform.

For details on updates and changes in this version, please refer to the release notes.

Note	
PDF Documentation: The full user guide is also available for download in PDF format:	
surf_nemo_1.02.pdf.	
For the latest SURF-NEMO release, you can access and download it directly from the official website:	

https://www.surf-platform.org

Gcense Information

SURF is distributed as a free and open-source software package under the terms of the GNU General Public License (GPLv3).

2. Introduction

The **Structured and Unstructured grid Relocatable ocean platform for Forecasting (SURF)** is an open-source, ondemand ocean numerical modeling platform designed for setting up, running and analysing high-resolution nested ocean models in any region within a large-scale Ocean Forecasting System. Designed for flexibility and ease of use, SURF enables users, including those with minimal technical expertise, to easily set up downscaling experiments using a clear and well-documented JSON configuration file. SURF can be operated on commercially available personal computers or laptops, ensuring broad accessibility and flexibility.

SURF integrates two state-of-the-art ocean models, the structured grid model NEMO (Nucleus for European Modelling of the Ocean) and the unstructured grid model SHYFEM-MPI (System of HydrodYnamic Finite Element Modules). This User Guide covers the structured grid component of SURF, known as **SURF-NEMO**. For information on the unstructured grid component, SURF-SHYFEM, please refer to the corresponding manual, available here.

This manual provides step-by-step instructions for running the platform, along with detailed explanations of the scripts and data structures, enabling users to modify or extend SURF according to their needs.

The package also includes case studies, complete with input datasets for bathymetry, coastlines, atmospheric and tidal forcing, as well as coarser-resolution parent ocean data for initial and boundary conditions. Output datasets are also provided for validation and for checking the correct implementation.

For more information, please visit the SURF website at:

https://www.surf-platform.org

2.1 **SURF-NEMO Relocatable ocean modelling platform**

SURF-NEMO (Trotta et al. 2016, 2021) is a numerical platform designed for forecasting hydrodynamic and thermodynamic fields with high spatial and temporal resolution. It can be embedded into any region within a larger-scale ocean prediction system, which operates at a coarser resolution.

The platform utilizes a **one-way nesting** approach for downscaling, where coarser resolution parent model fields are interpolated onto the child grid, providing initial and lateral open boundary conditions for the fine-grid model. Additionally, it supports **multiple nesting capabilities**, allowing for consecutive nested models with progressively finer grid resolutions. Starting from a large-scale ocean model, it can achieve horizontal grid resolutions down to a few hundred meters. At each nesting level, the parent coarse-grid model supplies the initial and lateral boundary conditions for the nested SURF child components.

This relocatable ocean model system is intended to be a valuable tool for supporting various **Decision Support Systems (DSS)** that may require high-resolution ocean fields, such as oil spill monitoring, search and rescue operations, navigation routing, fisheries and tourism.

2.2 Virtual Machine Environment

SURF-NEMO is distributed as a **Virtual Machine (VM)** image created using **VirtualBox**. This VM contains all the necessary libraries and software required to set up, run, and analyze downscaling experiments. The source code includes the hydrodynamic **NEMO model**, along with several pre- and post-processing tools, all installed within the VM environment. Detailed instructions for downloading and installing the package are provided in Chapter 3.1 of the Installation Guide.

A virtual machine is a software-based system that emulates a computer's operating system with virtual access to hardware resources such as CPU, RAM, networking, and storage. The operating system running inside the VM is referred to as the **guest**, and it operates within a window on your physical computer's operating system, known as the **host**.

The virtualization software used is the free and open-source **Oracle VM VirtualBox**, with the **Debian Linux** operating system installed inside the VM.

Virtual machines offer several advantages. They encapsulate an entire computing environment—operating system, applications, and data—within a single file. This makes setup easier compared to installing a full suite of software that must work together. A VM can be distributed as a pre-configured, ready-to-use system, simplifying both configuration and distribution. Additionally, VMs are versatile and can run on various hardware platforms.

2.3 Source Code

The **SURF-NEMO** source code is distributed as a compressed **tar.gz** archive, which includes the NEMO model code, a suite of pre- and post-processing tools, and a template user configuration file. Detailed instructions for downloading and installing the package are provided in Chapter 3.2 of the Installation Guide.

NEMO is an open-source ocean modeling framework, written in **Fortran 90** and optimized for parallel execution using **MPI**-based domain decomposition. Simulation outputs are stored in the **NetCDF** format, a widely-adopted standard for scientific data storage and exchange.

SURF-NEMO includes a suite of **preprocessing tools** for tasks such as mesh generation, dataset downloading, and remapping of input data, including ocean data for initial and lateral boundary conditions, as well as atmospheric and tidal forcing. It also offers **post-processing tools** to visualize and analyze simulation results.

The pre- and post-processing scripts are developed in Julia, NCL, Python, and Fortran programming languages. To efficiently handle NetCDF datasets, the platform utilizes NetCDF Operators (NCO) and Climate Data Operators (CDO), both optimized for SURF to reduce computation time and ensure efficient memory usage. Currently, these pre- and post-processing tools operate in serial mode (i.e., executed on a single processor).

The structure of the SURF source code package is detailed in Appendix B.

3. Getting Started

This chapter provides a step-by-step guide to help you quickly get started with the **SURF-NEMO** platform. It includes instructions for downloading and installing the **SURF Virtual Machine**, along with all the required SURF packages, **source code** and **static datasets** necessary to perform downscaling experiments. Additionally, this guide will show you how to compile the source codes, run a case study experiment in the Gulf of Taranto, followed by instructions on how to visualize and analyze the results. Finally, you will learn how to modify the user configuration file to execute new experiments, allowing you to customize the downscaling experiment according to your specific needs.

The **template experiment** makes it easier to run simulations without requiring in-depth knowledge of the underlying scientific details. For most applications, only a limited number of default values need to be adjusted. However, for more advanced use cases, such as changing turbulence models or adjusting numerical schemes, users should consult the **NEMO** User Manual for a deeper understanding. There are also video tutorials available online here, designed for beginners to walk through the basic features of the SURF platform step by step.

3.1 Download and Install SURF Virtual Machine (VM)

The SURF-NEMO platform is distributed as a **Virtual Machine (VM) image**, allowing for easy deployment. The VM image is packaged within a ZIP archive, with the naming format of the VM image releases as follows:

surf_vm_<VERSION>.zip

where VERSION refers to the specific version number of the release (e.g., surf_vm_1.03.zip for the current version).

The instructions below provide a step-by-step guide on how to download, install, and configure the SURF-NEMO VM using Oracle VirtualBox.

3.1.1 Installing Oracle VirtualBox

Before getting started, ensure that Virtual Box is installed on your system. If Virtual Box is not already installed, follow these steps:

1. Visit the VirtualBox download page and download the VirtualBox base package (version >=6) corresponding to your operating system (Windows, Mac, or Linux).

	VirtualBox Welcome to VirtualBox.org!
About Screenshots Downloads Documentation End-user docs Technical docs Contribute Community	VirtualBox Is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GRU) version 3. See "About VirtualBox" for an introduction. Presently, VirtualBox runs on Windows, Linux, macOS, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, KP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD. VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.
	Hot picks:
	Pre-built virtual machines for developers at ⊕ Oracle Tech Network Hyperbox Open-source Virtual Infrastructure Manager ⊕ project site phpVirtualBox AJAX web interface ⊕ project site
	Contact - Privacy policy - Terms of Use

Figure 3.1. Downloads VirtualBox base package.

- 2. Follow the provided installation instructions.
- 3. In addition to the base package, download the VirtualBox Extension Pack, which adds functionality like USB device support and remote desktop support. Double-click the downloaded extension pack and follow the installation steps. Please install the same version extension pack as your installed version of the VirtualBox base package.

	VirtualBox
	Here you will find links to VirtualBox binaries and its source code.
About	VirtualBox binaries
Screenshots	By downloading, you agree to the terms and conditions of the respective license.
Downloads	If you're looking for the latest VirtualBox 6.1 packages, see VirtualBox 6.1 builds. Version 6.1 will remain supported until December 2023.
Documentation	VirtualBox 7.0.2 platform packages
Technical docs Contribute Community	OrMindows hosts OrmaCS / Intel hosts OrmaCOS / Intel hosts OrDeveloper preview for macOS / Arm64 (M1/M2) hosts Linux distributions OrSolaris hosts OrSolaris hosts
	The binaries are released under the terms of the GPL version 3.
	See the changelog for what has changed.
	You might want to compare the checksums to verify the integrity of downloaded packages. The SHA256 checksums should be favored as the MD5 algorithm must be treated as insecure!
	SHA256 checksums, MD5 checksums
	Note: After upgrading VirtualBox it is recommended to upgrade the guest additions as well.
	VirtualBox 7.0.2 Oracle VM VirtualBox Extension Pack
	● All supported platforms
	Support VirtualBox RDP, disk encryption, NVMe and PXE boot for Intel cards. See this chapter from the User Manual for an introduction to this Extension Pack. The Extension Pack binaries are released under the VirtualBox Personal Use and Evaluation License (PUEL). Please install the same version extension pack as your installed version of VirtualBox.
	VirtualBox 7.0.2 Software Developer Kit (SDK)
	• ©+All platforms
	User Manual
	The VirtualBox User Manual is included in the VirtualBox packages above. If, however, you would like to take a look at it without having to install the whole thing, you also access it here:

Figure 3.2. Downloads VirtualBox Extension Pack.

4. Once installed, verify that VirtualBox is working by running the following command in your terminal:



This should display the current version of VirtualBox installed, confirming the installation was successful.

Tip: For more information on using VirtualBox, check out the VirtualBox Documentation.

3.1.2 Downloading and Installing SURF VM

• Visit the SURF platform website and download the current version of the SURF Virtual Machine Image (e.g., surf vm 1.03.zip). To download the Image directly from the terminal, use the following commands:

```
cd /Users/USERNAME/VirtualBox\ VMs;
wget https://www.surf-platform.org/repository/surf_vm/surf_vm_1.03/surf_vm_1.03.zip
```

The default installation path for VirtualBox VMs is typically /Users/USERNAME/VirtualBox VMs/ on macOS, with similar paths for other operating systems. In the virtual machines is installed the **Debian GNU/Linux 10 (buster)** operating system and all the necessary libraries required. The Guest Additions have been also installed to optimize the guest operating system for better performance and usability. Depending on your internet speed, the download may take a few minutes.

Structured and Unstructured grid Relocatable ocean platform Home About for Forecasting	Gallery Training - Publications Down	load Documentation - Profile - Logout
DOWNLOAD Download and start using the SURF plate and case studies directly from the link pro	form now! Access the Virtual Machin vided. Follow the instructions in the do	e package, source codes, static datasets, cumentation for setup and usage.
_	SURF-NEMO SURF-SHYFEN	I
Download the SURF-NEMO Package, (SURF-VM) and sta	along with its corresponding Virtual M tic datasets (SURF-DATASETS)	achine image SURF-NEMO V1.02 V
SURF-NEMO (v1.02)	SURF-VM (v1.03)	SURF-DATASETS (v1.01)
Download 🛓	Download 🛓	Download 🛓
wget https://www.surf-platform. 📵	wget https://www.surf-platform. 📭	wget https://www.surf-platform.
File size: 100MB	File size: 8.4GB	File size: 5.2GB
Released on 2023-01-17	Released on 2023-01-17	Released on 2020-11-08
Change Log	Change Log	Change Log
Please consult the	e "Getting Started" guide for detailed inst	allation instructions

Figure 3.3. Downloads SURF Virtual Machine.

· Unzip the archive file into the VirtualBox directory:

unzip surf_vm_1.03.zip

• Open VirtualBox, navigate to **Machine > Add**, and select the surf.vbox file to add the SURF VM to the list of Virtual Machine. This file is an XML file that contains settings of the Machine.

	Oracle VM VirtualBox Gestore	
Strumenti	Nuova Impostazioni Scarta Avvia	
\vee Surf	📃 Generale	📃 Anteprima
Spenta	Nome: surf Sistema operativo: Debian (64-bit) Gruppi: Surf	
	Sistema	surf
	Memoria di base: 2048 MB Processori: 4 Ordine di avvio: Floppy, Ottico, Disco fisso Accelerazione: VT-x/AMD-V, Paginazione nidificata, Paravirtualizzazione KVM	
	Schermo	
	Memoria video: 16 MB Scheda grafica: VBoxVGA Accelerazione: 3D Server di desktop remoto: Disabilitato Registrazione: Disabilitata	
	Archiviazione	
	Controller: IDE IDE master secondario: [Lettore ottico] Vuoto Controller: SATA Porta SATA 0: surf.vdi (Normale, 15,00 GB) Porta SATA 1: surf_scratch.vdi (Normale, 40,0	0 GB)
	խ Audio	
	Driver heats Care Audia	

Figure 3.4. Add SURF-VM in VirtualBox.

• Start the VM from the VirtualBox Manager by selecting it and pressing **Start** button at the top of the window. The VM Login should look like the figure 3.5.

surf [Running]	
surf	😡 🚯 Sat, 04:03 🖒
Enter your password Cancel Log In	
(Odebian ^{1°}	

Figure 3.5. Start SURF-VM.

- · Log in using the following credentials:
 - · Username: surf
 - Password: surf2020

Note

The VM image comes pre-packaged with all the necessary libraries required to execute the SURF-NEMO code. However, to facilitate updates and keep the image size manageable, the SURF-NEMO source code and static datasets are not included and must be installed separately. Detailed instructions for this process are provided in the following section.

3.1.3 Disk Partitions mounted on the SURF Virtual Machine

The SURF VM package includes two VirtualBox Disk Image (VDI) files:

- surf.vdi : Contains the Debian GNU/Linux 10 (buster) operating system.
- surf_scratch.vdi: Contains source code files, sample datasets, and experiments.

It is divided into two main partitions:

- the disk /dev/sda "mounted" as filesystems to the root directory /
- \bullet the disk <code>/dev/sdb</code> "mounted" in the directory <code>/scratch</code> .

From the guest operating system you can see the list of partitions by typing the following command:

sudo fdisk -l

3.1.4 Shared Folders Between Host and Guest Systems

You can mount other physical hard disks with VirtualBox (see the VirtualBox Manual for details). VirtualBox has the ability to **mount a shared folder between host and guest** in order to access files of your host system from within the guest system. There are a few steps involved:

- Shut down the VM, go to Settings > Shared Folders in VirtualBox.
- Click the Plus button and select the folder you want to share. Check the Auto-mount option and click OK.
- Restart the VM, and the shared folder will appear in /media with the prefix "sf_".

000	Oracle VM VirtualBox Gestore	
Strumenti		
Surf	Generale Sistema Schermo Archiviazione Audio Rete Porte Cartelle condivise Interfaccia utente	
	Cartelle condivise Percorso della cartella: rs/franz/Desktop/share_surf	
	Nome Percor aggio automatico Su 😱 SUIT	
	Cartelle della macchi Nome della cartella: share_surf	
	Sola lettura	
	V Montaggio automatico	
	Punto di mount:	
	Annulla OK	
	Annulia OK	
	Controller: SATA Porta SATA 0: surf.vdi (Normale, 15,00 GB) Porta SATA 1: surf_scratch.vdi (Normale, 40,00 GB)	
	🕼 Audio	
	Driver host: CoreAudio Controller: ICH AC97	
	Rete	
	Scheda 1: Intel PRO/1000 MT Desktop (NAT)	
	🖉 USB	
	Controller USB: OHCI, EHCI Filtri dispositivi: 0 (0 attivo)	
	Cartelle condivise	
	Cartelle condivise: 1	

Figure 3.6. Mount shared folders.

3.1.5 Configuring the SURF Virtual Machine

By default, the VM surf is configurated as in table Table 3.1. You can keep all defaults parameters or if it is not adequate for your application you can change settings. To change the configuration you need to shut down the virtual OS before you can edit settings.

- Select the surf VM in the VirtualBox Manager, right-click it and choose Setting.
- increase/decrease the number of cores based on your performance desires.
- increase/decrease the number of GB of RAM allocated to your VM according to the size of your computational domain.
- · increase/decrease the video memory and scale factor of your screen

					surf - S	Sistema			
			\bigcirc		F			-	
Generale	Sistema	Schermo	Archiviazione	Audio	Rete	Porte	Cartelle condivise	Interfaccia utent	e
			Scheda r	nadre	Proce	ssore	Accelerazione)	
	Merr	noria di bas	:e:						4096 MB 🗘
			4 MB			T T T Y		8192 MB	
	Ord	dine di avv	io: 🔽 🗎	Floppy	+]			
				Ottico	+				
				Disco fiss	50				
				Nete					
		Chips	et: PIIX3	\$					
Disp	ositivo di	puntamen	to: Tavolett	a USB		\$			
	Funzior	nalità estes	se: 🔽 Abilita I	/O APIC					
			Abilita	EFI (solo	alcuni	sistemi d	operativi)		
			🗸 Orologi	o hardw	are in o	ra UTC			

Figure 3.7. Change VM configurations.

If you want to add more storage space to a VM you can also expande the virtual hard disk. There are a few steps involved:

- With the VM Power off, open a terminal and move to the location of the surf_scratch.vdi file that you want to resize,
- At the terminal prompt, type the command:

VBoxManage modifyhd surf_scratch.vdi --resize SIZE_MB

- Restart the SURF VM and open the GParted application from the Application Menu
- · Select the /dev/sdb partition (an unlocated drive space is now available). Resize to the unalocated area



(A) Before.

Figure 3.8. Enlarge the virtual disk.

Parameter	Description	Values
Name	Name given to the VM	surf
Guest OS	Operating system running on this VM	Debian Linux
Memory	Amount of memory available to this VM	2 GB
Cores	Number of CPU cores being used by this VM	2
Disk Capacity	Total disk capacity available to this VM	40 GB
Network Adapters	Number of network adapters available to this VM	1
IP Address	IP address assigned to the VM	x

Table 3.1: Summary of Virtual Machine fields and configurations.

3.2 Download and Install SURF packages

The installed Virtual Machine (VM) does not include the SURF packages, which consist of source codes and static datasets. You will need to download and install them in the VM's scratch directory (/scratch, VM directory structure shown in Figure B.1).

The SURF packages are distributed as a GZIP Compressed Tar Archive file (with a .tar.gz extension). The naming convention for the releases follows this format:

packageName_<VERSION>.tar.gz

where VERSION refers to the specific version number of the release (e.g., surf_nemo_1.02.tar.gz for the current version of the SURF-NEMO package version).

The instructions below explain how to install the packages in the SURF VM.

• After logging into the SURF VM, download the latest version of the SURF-NEMO (surf_nemo_1.02.tar.gz) and SURF-DATASETS (surf_datasets_1.01.tar.gz) packages directly from the SURF platform website and save it in the directory /scratch/surf/surf_install/releases/ (for simplicity, we abbreviate this location as ssurf_Releases). Use the following commands to download the packages:

```
# Download the SURF source code into the specified directory
wget -P /scratch/surf/surf_install/releases https://www.surf-platform.org/repository/surf_nemo/surf_r
```

Download the static dataset into the specified directory
wget -P /scratch/surf_surf_install/releases https://www.surf-platform.org/repository/surf_datasets/su

• Go to the directory <code>\$SURF_RELEASES</code> and run the installation bash script <code>install.sh</code> followed by the package name. Use the following commands to install both the SURF-NEMO and SURF-DATASETS packages:

cd /scratch/surf/surf_install/releases
install.sh surf_nemo_1.02.tar.gz
install.sh surf_datasets_1.01.tar.gz

The installation process will extract the archive in the directory /scratch/surf_surf_nemo/ and /scratch/surf/ surf_datasets/, respectively, and will create a symbolic link current in this directory that points to the extracted folder (for simplicity, we abbreviate this location as \$SURF NEMO, \$SURF DATASETS, respectively).

For a detailed description of the directory structure and contents of each package refer to the Appendix B.

3.2.1 Compiling the source code

Once the installation of the SURF-NEMO package is complete, you need to compile the source codes in order to create the executable files needed to perform specific tasks. The executable files should not be recreated unless you need to modify the source code.

The compilation process is handled using the Unix/Linux make utility and the following tools:

- Fortran 90 Compiler used for compiling the core numerical models.
- C Preprocessor (cpp) processes C-style macros.
- MPI Library required for running simulations in parallel mode.
- netCDF Library allows reading and writing of data in portable netCDF format.

All of these tools are already installed and configured on the SURF platform, so no additional setup is required.

To compile the source codes go to the directory /scratch/surf/surf_nemo/current/scripts/ and run the compilation bash script compile.sh followed by the package name (or by the word 'all' to compile all the packages):

```
cd /scratch/surf/surf_nemo/current/scripts;
./compile_codes.sh all
```

Compilation could take a few minutes and it will create the executable files for each program present in the SURF-NEMO package.

3.3 Running the Gulf of Taranto Case Study

With the SURF-NEMO VM-Image, source code and static datasets successfully downloaded, you are now ready to start using the SURF-NEMO platform. Follow the steps below to set up and execute the Gulf of Taranto case study.

In this case study, the SURF platform is used to implemente an high-resolution model in the Gulf of Taranto, located in the northern Ionian Sea. The simulation starts on 5 October 2014 at 00:00 and run until 7 October 2014 at 24:00. This section provides step-by-step instructions on how to set up, run, and analyze the downscaling experiment.

3.3.1 Setting Up the Experiment

1. Download and extract the user configuration file for this test case experiment (gulfTaranto_20141005_config.tar.gz) from the SURF website and saves it into the /scratch/surf/from GUI directory by running the following commands:

Download the configuration file directly into the specified directory
wget -P /scratch/surf/from_GUI https://www.surf-platform.org/repository/surf_nemo/surf_nemo_1.02/case_

Extract the downloaded file
cp /scratch/surf/from_GUI; tar -zxvf gulfTaranto_20141005_config.tar.gz

This file contains the user configuration settings specific to this case study (see chapter 5 for more details). The folder name gulfTaranto 20141005 will serve as the Experiment ID, uniquely identifying the experiment.

2. Download and extract the necessary input data required for this test case experiment (gulfTaranto_20141005_indata.tar.gz) from the SURF website and saves it into the /scratch/surf/indata_offline directory by running the following commands:

```
# Download the configuration file directly into the specified directory
wget -P /scratch/surf/indata_offline https://www.surf-platform.org/repository/surf_nemo/surf_nemo_1.02
# Extract the downloaded file
```

```
cp /scratch/surf/indata_offline; tar -zxvf gulfTaranto_20141005_indata.tar.gz
```

Note

If you want to change the local repository pat <u>indata_offline</u> to a different location, ensure that you update the path in the user configuration file accordingly.