



D4.3 Procedures and instructions related to common environments



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 842009

DELIVERABLE NUMBER	D4.3
DELIVERABLE TITLE	Procedures and instructions related to common environments
RESPONSIBLE AUTHOR	Ioannis Andreou, OPEKEPE Greece

GRANT AGREEMENT N.	842009
PROJECT ACRONYM	NIVA
PROJECT FULL NAME	A New IACS Vision in Action
STARTING DATE (DUR.)	1/06/2019
ENDING DATE	30/05/2022
PROJECT WEBSITE	
COORDINATOR	Sander Janssen
ADDRESS	Droevendaalsesteeg 1, Wageningen
REPLY TO	Sander.janssen@wur.nl
PHONE	+31 317 481908
EU PROJECT OFFICER	Mrs. Francisca Cuesta Sanchez
WORKPACKAGE N. TITLE	WP4 Knowledge Information System
WORKPACKAGE LEADER	8 - AGEA
DELIVERABLE N. TITLE	D4.3 Procedures and instructions related to common environments
RESPONSIBLE AUTHOR	Ioannis Andreou, OPEKEPE Greece
REPLY TO	ioannis.andreou@opekepe.gr , kostas.kountouris@opekepe.gr
DOCUMENT URL	
DATE OF DELIVERY (CONTRACTUAL)	1 December 2019 (M6)
DATE OF DELIVERY (SUBMITTED)	1 December 2019 (M6)
VERSION STATUS	V1.0 Final
NATURE	Report
DISSEMINATION LEVEL	PUBLIC
AUTHORS (PARTNER)	Ioannis Andreou, Kostas Kountouris, OPEKEPE Greece

VERSION	MODIFICATION(S)	DATE	AUTHOR(S)
0.1	Final version	21 November 2019	Ioannis Andreou, OPEKEPE Greece
0.9	Final version	28 November 2019	Ioannis Andreou, OPEKEPE Greece

Table of Contents

REPOSITORIES - VERSION CONTROL SYSTEM.....	6
Introduction	6
Organization of projects and users	6
Initializing and Managing Users.....	8
We aim to keep the project structure flat, avoiding creating sub-groups in the main group, but we will do so if requested, to help with managing access.	8
THE Git PLATFORM	9
Prerequisites	9
Git Commands.....	9
Comparison to over VCS's	9
Free tools and IDEs.....	10
Other Gitlab Tools	11
Contacts –Support Group.....	11

REPOSITORIES - VERSION CONTROL SYSTEM

Introduction

The Gitlab platform has been selected for central management and control of the source code repositories of the NIVA project. Gitlab is an online platform/locally deployable solution for the management of git repositories; it provides storage for the repositories as well as a set of other tools, accessible through a web interface, that assist with aspects of source management such as repository user management and deployment.

The Gitlab software is free; however a subscription to Gitlab has been bought, in order for the software to be used as a service on the company platform. Using the product as a service was the fastest way to have the software up and running, without having to configure network and software settings that are required by our security standards and the openness needs of the NIVA project. The user access framework used by the purchased platform conforms to the ISO standards required by our organization.

Organization of projects and users

In order to facilitate the use of the common software, as well as to be able to guide and instruct on issues that relate to software development and sharing between partners, we implement the needed infrastructure and propose general rules towards using the platform. However, feedback is welcome and we can accommodate a different approach per partner if needed, as long as the basic principles of security and openness required by the projects are adhered to.

A local installation of Gitlab probably won't require partners to define project groups; however using the service, we need to define boundaries to any other projects/organizations that are hosted on that platform. To this end, two project groups were created:

- **Test group:** <https://gitlab.com/nivagroup>

The test group may be used as a test bed for partners to try the platform. It also comes with already made dummy projects that follow the structure of the source deliverables expected, according to the NIVA agreement. You may use this as a template while creating your projects on the production group. Please note that this group is **public**, meaning that everyone in the world has access to any information uploaded.

N **NIVA Group - New IACS Vision in Action**
 - TEST
 Group ID: 6338690

Niva project source code repository

Subgroups and projects Shared projects Archived projects Search by name Last created

W	WP3 Harmonisation and Interoperability	★ 0	1 month ago
U	UC5b Scheme Eligibility and Payments Eligibility: Click-and-Pay	★ 0	1 month ago
U	UC5a Land Parcel Identification System update & change detection	★ 0	1 month ago
U	UC4b Machine data in Geo-spatial on-line aid application as added value c	★ 0	1 month ago
U	UC4a Geotagged Photos	★ 0	1 month ago
U	UC3 Farm Registry	★ 0	1 month ago
U	UC2 Prefilled application, GSAA/Land link	★ 0	1 month ago
U	UC1c Farmer Performance	★ 0	1 month ago
U	UC1b Agro-environmental monitoring	★ 0	1 month ago
U	UC1a Earth Observation Monitoring and Traffic Lights	★ 0	1 month ago

- **Production (official) group:** <https://gitlab.com/nivaeu>

This is the actual group for the NIVA project. Initially this group is private, in order to control who has access to each component individually. You will find the group to be empty. It is recommended that you create your own projects in the group, in order to gain the highest possible level of access to it, following the conventions of the project names that exist in the test group. This way each partner takes full control of their work; however it is expected from the partners to eventually provide at least guest access to the other members of NIVA. The total group will continue to be private until work is finished or the Commission instructs us to open it to the public.

If you need to create additional repositories feel free to do so, but apply at least the following constraints:

- Don't use capital letters in repository names.
- Start the name of the repository with the package that it belongs to e.g. *uc1a_docs* may be an extra documentation repository for package *uc1a*. Use underscore (_) to separate words.

- If you are not sure about the above or have questions about the use of the service please contact administration. Please use the provided e-mail(s) at the end of this document.

Other suggestions:

- Coding conventions: We don't provide coding conventions or platform suggestions as these are not requested or required by the agreement. Partners probably need this level of freedom anyway.

- **Do not rely only on the Gitlab service as a backup** for your work. Git is a powerful tool that gives more options. Please maintain your own common repository in parallel, as a local Gitlab installation or simply as git folder(s) in your file server.

- Feel free to add other resources that are needed by your project to work, such as images, libraries etc. It is assumed that the submitted components can be installed and work as they are, where applicable.

Initializing and Managing Users

To gain access to the NIVA group one must follow these steps:

a) Fill in the information on the common excel file that was created for this purpose in Sharepoint under "Workpackage folders/WP4/T4.1 Source code repository and catalogue of assets/NIVA_D4_1_GITLAB_USERS_TABLE_V1".

Complying with security constraints, we need an e-mail for each person (no group accounts!). A phone number is also needed and may be used in case of doubt. We need to assign a main contact person for any issues, to each project or working group, who is assigned with maintaining the project(s) for that group.

b) Create a gitlab.com account, using your official e-mail, which is included in the above file.

c) Apply for access to the group. It's better to contact ioannis.andreou@opekepe.gr or nikolaos.galanis@opekepe.gr too and specify if you will be maintaining projects under this account.

d) Work, after you have gained access to the project group.

e) After projects are created, each additional account has to be given access to project(s) individually.

We aim to keep the project structure flat, avoiding creating sub-groups in the main group, but we will do so if requested, to help with managing access.

THE Git PLATFORM

Prerequisites

In this section we assume that readers have (or will acquire) Version Control Systems basic knowledge i.e. they can use other simpler systems such as SVN or CVS.

Git Commands

Git provides a vast array of commands and tools, which are free and open source. Even if you never need to use another git console command, you will probably need the following two:

git init (Creates the .git "hidden" folder in your working directory in order to start using git locally.)

git init --bare (Similar but you use it to create a shared repository e.g. the one you keep on your file server.)

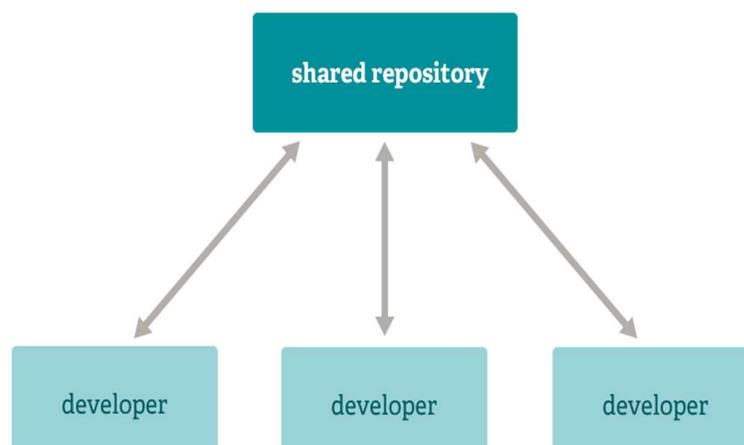
Git online resources:

<https://git-scm.com/> (main Git site)

<https://www.atlassian.com/git> (a site with Git tutorials)

Comparison to over VCS's

If you have ever use a VCS before, you know the most basic model for a shared repository for developers looks something like the following diagram:



In this diagram, every developer maintains his local working directory, which he uses to update the shared repository and vice versa.

Git allows you to have more than one shared repository though, extending this server-client architecture to p2p. Every shared repository is a "remote" and any number of "remotes" can be added to your working directory, using a command in the "git remote" group. In the context of NIVA we don't expect to have to complicate the way our partners work, however we suggest, even require, that **at least two remotes are used for each project**, the one in Gitlab and one of your own, for backup if nothing else. **Each working group is responsible for the safety of their projects, inside of the Gitlab platform or not.**

Tracking/adding/removing/changing files are completed in 2 steps:

- a) **staging** (similar to tracking in other systems) files and
- b) **committing**.

Committing does not affect the shared repository. Commit happens on the working folder, which is a repository with history information of its own.

To affect the shared repository you need to **push** to that **remote**, or **pull** and **merge** for the reverse action.

In the scenario with two shared repositories we suggest, you would

- a) **stage**, b) **commit** and then c) **push to each** remote repository separately.

Free tools and IDEs

Assuming knowledge of how git works, one does not need to use git console commands to work with it. There is a plethora of free and open source - amongst others - tools, that one can use to simplify and speed up the development process. At the end of the day, most modern IDEs/programming editors provide embedded support or plug-ins/extensions for git.

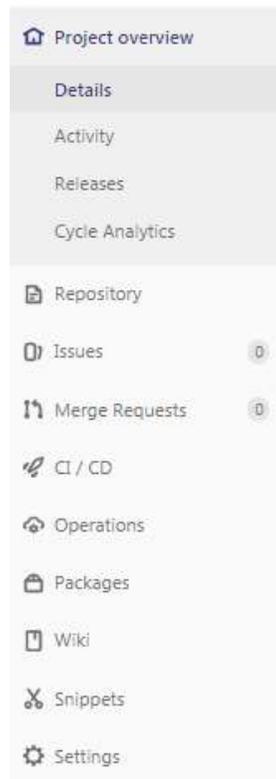
Some notable lightweight tools you could use outside of your IDE (that also run on Windows) are:

<https://git-scm.com/docs/git-gui> Git gui will provide your Windows Explorer right-click extensions for the most common commands and a GUI for management/history.

<https://gitforwindows.org/> A unix-like console to run git commands on. Given the incompatibility of the windows console (cmd) with GNU tools, you might need to have this, or similar software, installed to manually run commands.

Other Gitlab Tools

One may view and use Gitlab simply as storage space for the common git repositories and that's an acceptable approach. However Gitlab provides other tools as advertised in the platform. Accessing a project you have some of the following options:



Through this interface, you can create a wiki, track issues, merge requests, manage CI/CD and use other tools. During the course of the NIVA project it is assumed that you will have to work together with the administration in order to utilize these tools, whenever broader cooperation is needed.

Contacts –Support Group

The administrator of the Gitlab platform currently is:

Ioannis Andreou, Ioannis.andreou@opekepe.gr

Assisted in technical support by:

Nikolaos Galanis, nikolaos.galanis@opekepe.gr