Argument list too long

Last edited by **VREDDY10** 2 years ago

Issue Description:

`ninja: fatal: posix_spawn: Argument list too long` indicates when a user feeds too many arguments into a single command which hits the **ARG_MAX** limit... An argument, also called a command-line argument, can be defined as the input given to a command, to help control that command line process.

Solution 1:

On Linux, the maximum amount of space for command arguments is 1/4th of the amount of available stack space. So, a solution is to increase the amount of space available for the stack.

1. The default amount of space available for the stack is something like 8192 KB. You can see the amount of space available, as follows:

```
vvadlamu@IND6R0JVP2:~/15Nov$ ulimit -s
8192
```

2. Choose a larger number, and set the amount of space available for the stack. For instance, if you want to try allowing up to 65536 KB for the stack, run this:

```
vvadlamu@IND6R0JVP2:~/15Nov$ ulimit -s 78457
78457
```

❗ **Note:** You may need to play around with how large this needs to be, using trial-and-error.

This solution is Linux-specific. We suspect it probably won't help on any other Unix operating system (not tested).

References:

1. https://unix.stackexchange.com/a/45584
2. https://codeforces.com/blog/entry/80688

Solution 2:

Adding the following Entries to the cmake files

```
SET(CMAKE_C_USE_RESPONSE_FILE_FOR_OBJECTS 1)
SET(CMAKE_CXX_USE_RESPONSE_FILE_FOR_OBJECTS 1)

SET(CMAKE_C_RESPONSE_FILE_LINK_FLAG "@")
SET(CMAKE_CXX_RESPONSE_FILE_LINK_FLAG "@")

SET(CMAKE_NINJA_FORCE_RESPONSE_FILE 1 CACHE INTERNAL "")
```

References:

1. https://stackoverflow.com/questions/43184251/cmake-command-line-too-long-windows

- You can use `devNext --help` or `devNext <command> --help` for more options/help.
- ❓ **Support:** If you've any Queries w.r.t above mentioned steps, please reach out to devNext.support@visteon.com

Artifact upload and ensuring the retention policy using devNext

Last edited by **VREDDY10** 7 months ago

❗ **Note:**

- This feature is available with devNext V4.0.0.
- This devNext upload supports currently Jenkins CI.

This feature is designed to simplify the process of uploading build artifacts to Artifactory. It introduces a devNext command that allows Jenkins CI jobs to upload build artifacts along with Build Info details to Artifactory. Additionally, it ensures compliance with the repository's configured retention policy.

1.1 Information Required in the Profile File (.ini File)

```
[artifact]
ARTIFACTORY_SERVER= <SERVER_NAME_OF_THE_ARTIFACTORY> ;# Mandatory
ARTIFACT_REPOS= <LIST_OF_ARTFATORY_RESPOSITORIES_SEPARATED_BY_COMMA> ; # Mandatory
ARTIFACT_OUT= <RELATIVE_PATH_TO_THE_SOURCE_DIR/FILE_TO_UPLOAD> ; # Mandatory
ARTIFACT_EXCLUDES= <RELATIVE_PATH_TO_THE_SOURCE_DIR/FILE_TO_BE_EXCLUDED_SEPARATED_BY_COMMA>
ARTIFACT_NAME= <NAME_FOR_THE_ARCHIVE_WHICH_WILL_BE_UPLOADED_TO_ARTIFACTORY>
```

Example:

```
[artifact]
ARTIFACTORY_SERVER = jfrog.chennai.visteon.com
ARTIFACT_REPOS = NISSAN_P13A_EL2_CRE_MY2024_EP30319_NightlyBuilds,NISSAN_P13A_EL2_CRE_MY2024_EP30319_Release
ARTIFACT_OUT = programs/nissan/my2024/p13a-el2/out/IMG/P13A_RUN1/release/IMG_BUILD/images
ARTIFACT_EXCLUDES = programs/nissan/my2024/p13a-el2/out/IMG/P13A_RUN1/release/CMakeFiles
ARTIFACT_NAME = ${projectname:OEM}_${projectname:VARIANTNAME}
```

a. ARTIFACTORY_SERVER (Mandatory):
This key holds the value of the domain name of the Artifactory server where your repositories are available to store the build artifacts.

Example:

> `ARTIFACTORY_SERVER=jfrog.chennai.visteon.com`

b. ARTIFACT_REPOS (Mandatory):
This key must be assigned with a list of the names of the project's Artifactory repositories separated by a comma as mentioned in the example.

Example:

> `ARTIFACT_REPOS=NISSAN_P13A_EL2_CRE_MY2024_EP30319_NightlyBuilds,NISSAN_P13A_EL2_CRE_MY2024_EP30319_Release`

❗ **Note:** The Command line input '--repo-name' will be validated against repo list provided in this variable.

c. ARTIFACT_OUT (Mandatory):

- Path of the directory/file, relative to the workspace. If the path is a folder, the folder will be archived in ".tar.gz" format, and the name for the file will be taken, if specified, from the following key **ARTIFACT_NAME**
- If it is a file, it will be directly uploaded with the same name of the file.

Example:

i) If it is a directory path:

> `ARTIFACT_OUT= programs/nissan/my2024/p13a-el2/out`

ii) If it is a File Path

> `ARTIFACT_OUT= programs/nissan/my2024/p13a-el2/out/archive.tar.gz`

d. ARTIFACT_EXCLUDES (Optional):
Path of the directory(s)/file(s), relative to the workspace, separated by a comma which are to be excluded from the archive.

Example:

> `ARTIFACT_EXCLUDES= programs/nissan/my2024/p13a-el2/out/IMG/P13A_RUN1/release/CMakeFiles`

❗ **Note:** This variable is applicable only in case of the ARTIFACT_OUT is a directory.

e. ARTIFACT_NAME (Optional):
Name for the archive file that will be created for the ARTIFACT_OUT directory.

- ❗ **Note:** If ARTIFACT_NAME is not defined or --artifact-name argument is not passed, "JOB_BASE_NAME" value will be used as default artifact name.
- This variable is applicable only in case of the ARTIFACT_OUT is a directory.

Example:

> `ARTIFACT_NAME= P13A_EL2_OUT`

1.2. Command for Artifact upload

devNext command to upload build artifacts with retention policy being upheld.

```
dn ci upload --profile <profile-url> --dir <directory-path> --repo-name <repo-name> [--artifactory-server <server_Name>
```

❗ **Note:**

- In the above command, it is mandatory to provide the "repo-name" details and it must be of one of the repositories mentioned in the "ARTIFACT_REPOS" variable in the profile.
- "**--dst-dir**" is the Path of Artifact file within the Artifactory Repository

Examples:

1. General Example:

> a. **With --dst-dir**
> `dn ci upload --profile profile.ini --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_Release --dst-dir REL_20241210_V3 --dir .`

> b. **Without --dst-dir**
> `dn ci upload --profile profile.ini --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_Release --dir .`

```
Usage: dn ci upload [OPTIONS]

  Artifact upload

Options:
  --profile TEXT                  Profile to use for upload  [required]
  --d, --dir PATH                 Path of the workspace
  --repo-name TEXT                Artifactory repository Name  [required]
  --artifactory-server TEXT       Artifactory server URL  Example:
                                  jfrog.sofia.visteon.com
  --src-dir TEXT                  Path of folder/file to be archived and
                                  uploaded, Relative to workspace folder
  --artifact-name TEXT            Artifact name
  --excludes TEXT                 Path of files/folders to be excluded from
                                  the artifact archive, Relative to workspace
                                  folder
  --dst-dir TEXT                  Path of Artifact files within the
                                  Artifactory Repository
  --env TEXT                      <ENVNAME>=<ENVVALUE> can be used to set an
                                  environment variable. If you have more than
                                  one env variable then need to repeat '--env'
                                  for each variable.Example: --env
                                  MANIFEST=/home/users/manifest.xml --env
                                  BRANCH=develop
  -l, --location, --loc [sofia|bangalore|chennai|pune|mexico]
                                  Nearest Location for Packages and
                                  Sources(Mirrors) Download
  --help                          Show this message and exit.
```

2. Overrides the Artifactory Server Name:

> `dn ci upload --profile profile.ini --dir . --artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS`

3. Overrides the Artifactory Server Name, Artifact out

> `dn ci upload --profile profile.ini --dir .--artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS --src-dir program/my2025/out`

4. Overrides the Artifactory Server Name, Artifact out and excludes:

```
dn ci upload --profile profile.ini --dir .--artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS --src-dir program/my2025/out --excludes program/my2025/out/lib,program/my2025/out/doc
```

5. Overrides the Artifactory Server Name, Artifact out, excludes and Artifact Name:

```
dn ci upload --profile profile.ini --dir .--artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS --src-dir programs/nissan/my2024/p13a-el2/out/archive.tar.gz --artifact-name b10_integration_out
```

6. Other Examples

```
dn ci upload --profile profile.ini --dir .--artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS --src-dir programs/nissan/my2024/p13a-el2/out/archive.zip
```

```
dn ci upload --profile profile.ini --dir .--artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS --src-dir programs/nissan/my2024/p13a-el2/out/archive.tar.gz --artifact-name b10_integration_out
```

```
dn ci upload --profile profile.ini --dir .--artifactory-server jfrog.chennai.visteon.com --repo-name NISSAN_P13A_EL2_CRE_MY2024_EP30319_NIGHTLYBUILDS --artifact-name b10_integration_out
```

1.3. What must be configured?

To maintain the retention policy, the following must be configured for the smooth flow complete process.

1.3.1 Artifactory Retention Policy

The following property "**repository.retention.builds**" must be configured to the Artifactory Repository(s) to be considered for the Retention Policy and to delete the builds and respective artifacts automatically. Note that the value for this field is a number and should be greater than **1**.

Raise a ServiceNow Request for the same: https://visteon.service-now.com/vss?id=sc_category&sys_id=40e25548dbb6fb40370e94d6db9619a0&catalog_id=af482c51db847300656e9674db961901

For further support reach out to Artifactory Team: artifactory@visteon.com

Example:

```
repository.retention.builds = 30
```

Where 30 is the number of Build Artifacts allowed for the repository. If it exceeds, the older Builds and their related artifacts will be deleted prior to uploading the next Build artifact. This is done to ensure that the retention policy and thus the repo.quota are satisfied.

1.3.2 Configuration of SSH key

You need to configure the **SSH Key** in the default location of **$HOME/.ssh**. This key should have permission to upload and delete the builds and artifacts from the project-specific repositories.

Note: Once the ssh keys got generated, the public key must be configured in the Artifactory. To configuring the same please follow the instructions /steps mentioned in the below PPT. For further support reach out to Artifactory Team: artifactory@visteon.com

Link to PPT: SSH Configuration - Artifactory

The Default .ssh key(Private key) path is consider as follows:

```
${HOME}/.ssh/id_rsa
```

Example:
/home/jenkins/.ssh/id_rsa

In case you want to keep the ssh key in a customized file/directory, define JFROG environment variable as described below:

```
(for Linux) export JFROG_CLI_SSH_KEY_PATH=<path_to_the_SSH_KEY>
Or
(for Windows) set JFROG_CLI_SSH_KEY_PATH=<path_to_the_SSH_KEY>
```

export **JFROG_CLI_SSH_KEY_PATH**=/home/jenkins/.ssh/id_rsa

Example (Linux) 2:

export **JFROG_CLI_SSH_KEY_PATH**=/home/jenkins/mykeys/id_rsa

1.3.3 Configure the .netrc file with Artifactory Access Credentials

To ensure retention policy, a user id having permission to **read** the retention property of the Artifactory repository must be configured in the “.netrc” file as follows:

```
machine <ARTIFACTORY_SERVER_URL> login <USERNAME(CDSID)> password <ACCESSTOKEN>
```

Example:

> *machine jfrog.bangalore.visteon.com login CDSID password ACCESSTOKEN*

If the “.netrc” file was not configured with the credential of Artifactory

1. If PAT is enabled in gitconfig (devnext.auth)
    1. devNext upload command and the Job will be terminated.
2. If PAT is not enabled in gitconfig (devnext.auth)
    1. Credentials provided for Visteon git servers will be considered.
    2. If no Visteon git server entries are found in “.netrc”, default credentials provided in “.netrc” are taken.
    3. If no credentials are defined in “.netrc”, devNext upload command and the Job will be terminated.

Q2A Forum: Q&A platform helps find answers, ask your questions, learn and share knowledge.

Support: If you've got any questions about above-mentioned steps, please reach out to devNext.support@visteon.com

Automation Script for Pre Requisites Setup

Last edited by **SMOHANA1** 5 months ago

Executing devNextv3-install.sh

This script is to ease the devNext installation and prerequisite Management.

Run the below command to install devNext:

```
source <(curl -ks https://jfrog.sofia.visteon.com/artifactory/Devops-Application-Engineering/devNext/devNextv3-install.s
```

Supported Functionalities of devNextv3-install.sh script.

- Python 3.8.10+ installation
- PIP 23.3.2 Installation
- Latest devNext version Installation
- QNX License file Management
- Git Config Management
- Shared Conan User Home Config
- Support for sudo and dzdo
- Disk Space available (free disk space) is shown to the User
- The devNextv3-install.sh script will always install the latest version of devNext

(or)

User can still follow the devNext Wiki to do the setup Manually.

(or)

- Download the devNextv3-install.sh

Once you download the script run the below command:

- `chmod +x devNextv3-install.sh`
- `source devNextv3-install.sh`

Q2A Forum: Q&A platform helps find answers and ask your questions, learn and share knowledge https://q2a.visteon.com

Support: If you've any Queries w.r.t above mentioned steps, please reach out to devNext.support@visteon.com

Accessing workspace with Eclipse

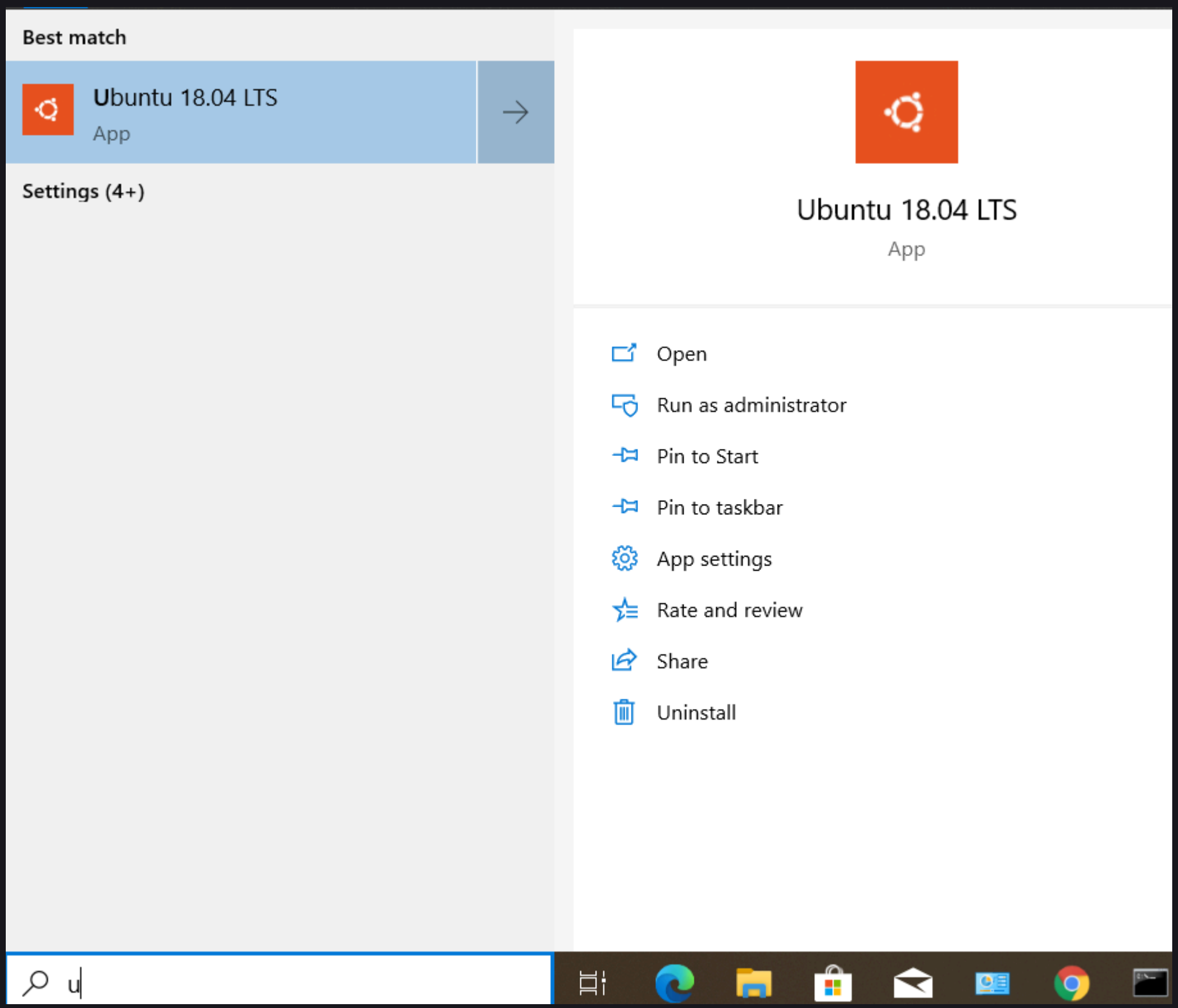
Last edited by **VVADLAMU** 3 years ago

Pre-Requisites

- Please ensure you've made followed the GUI with WSL

Steps

1. Open Ubuntu distro from start Menu.



1. Go to desired location and Execute the following command to download eclipse inside WSL

```
wget https://jfrog.glcc.visteon.com/artifactory/Devops-Application-Engineering/RTC/702/Linux/Eclipse411_EWM702_Linux64.
```

For different packages, Please refer to the : https://jfrog.sofia.visteon.com/ui/repos/tree/General/Devops-Application-Engineering/RTC

1. Execute the following command to extract Eclipse

```
unzip -q Eclipse411_EWM702_Linux64.zip
```

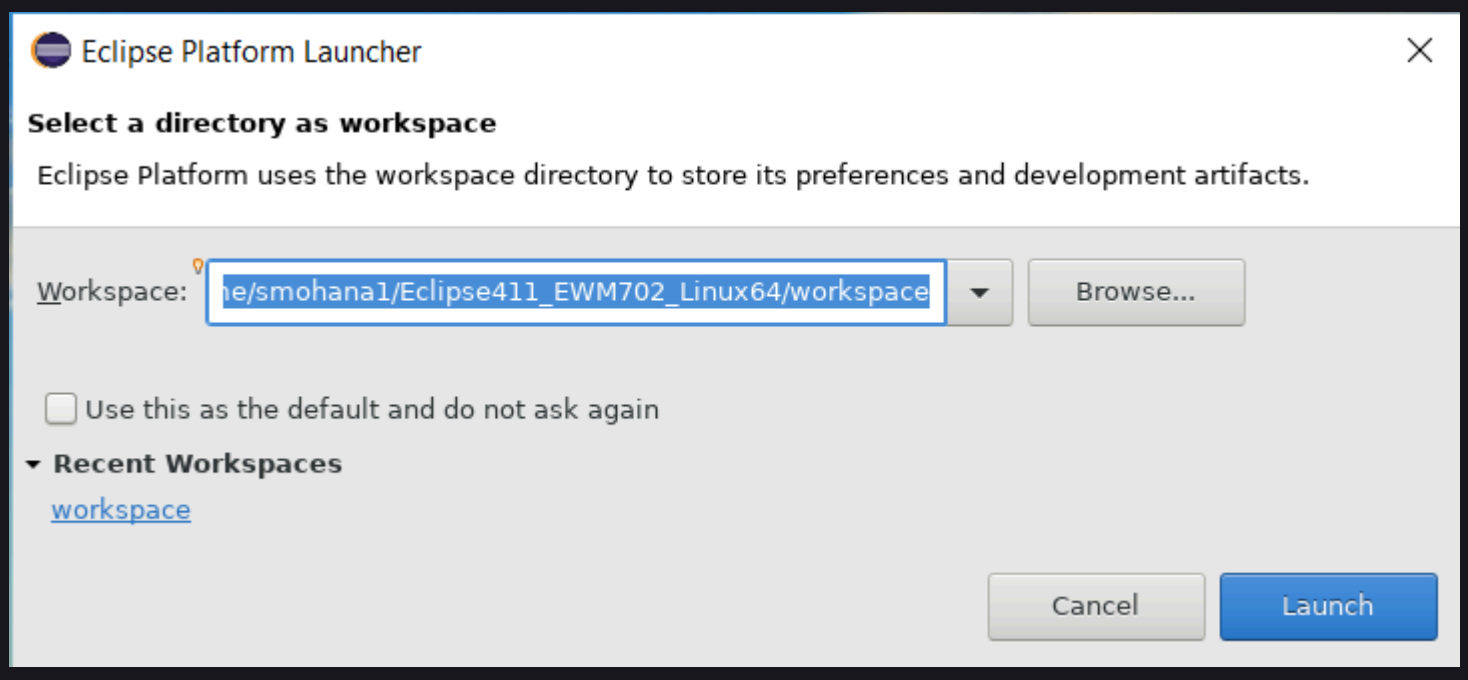
1. Go to the path and execute eclipse binary

```
cd Eclipse411_EWM702_Linux64/client/eclipse && ./eclipse
```

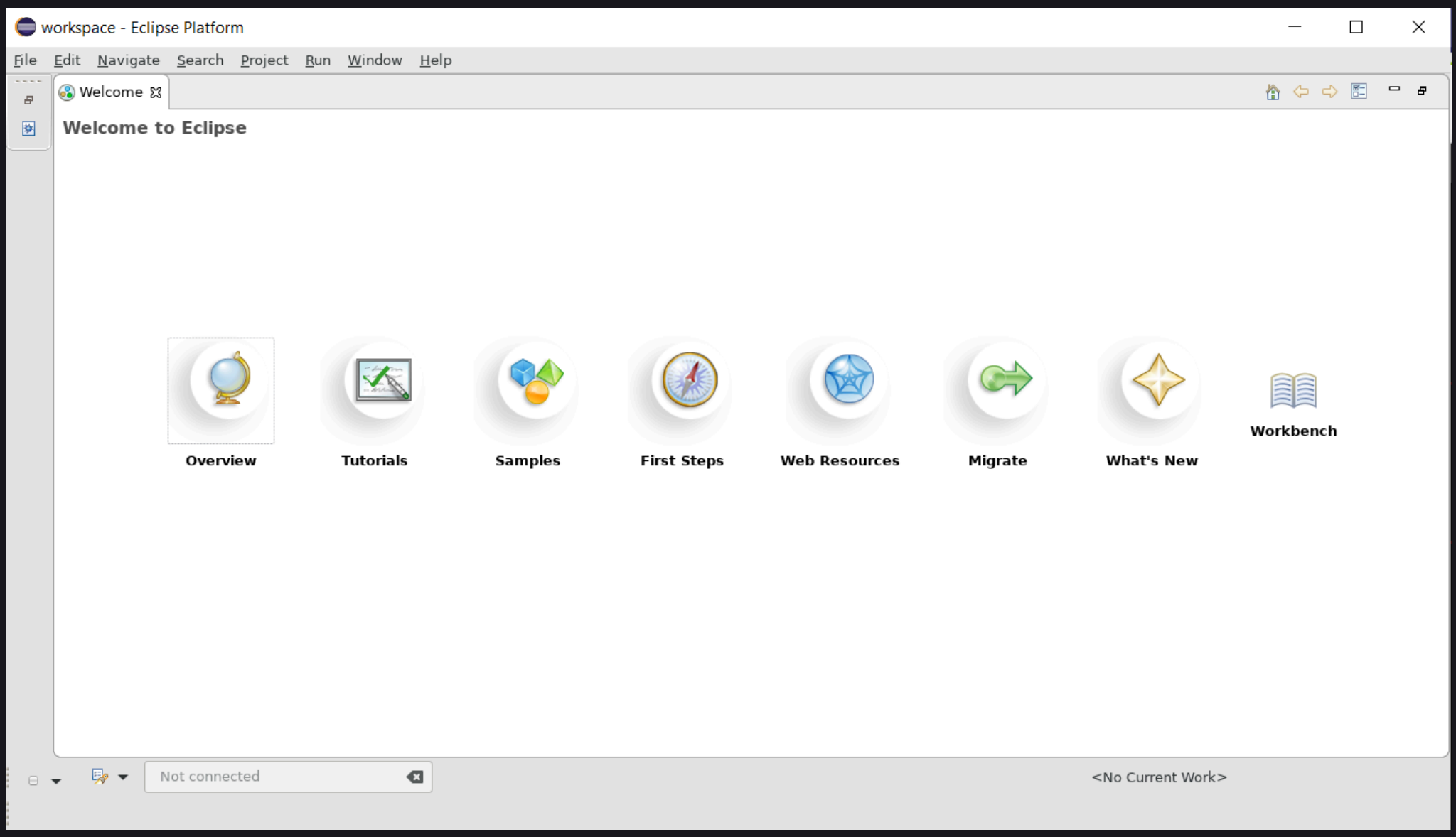
Eclipse Platform Launcher will pop-up

- Eclipse platform takes the default workspace directory to store its preferences and development artifacts

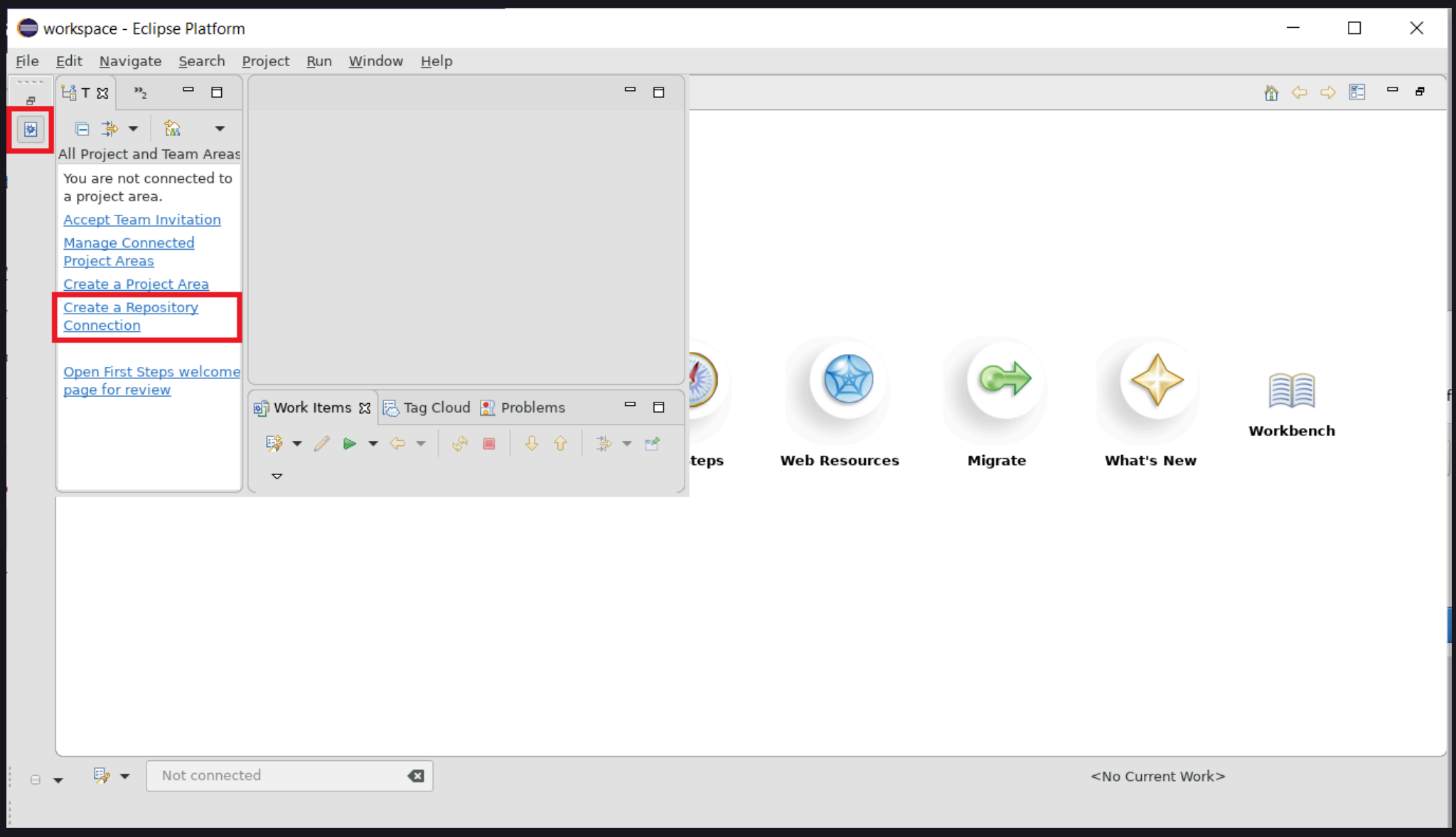
Example-



Eclipse Dashboard



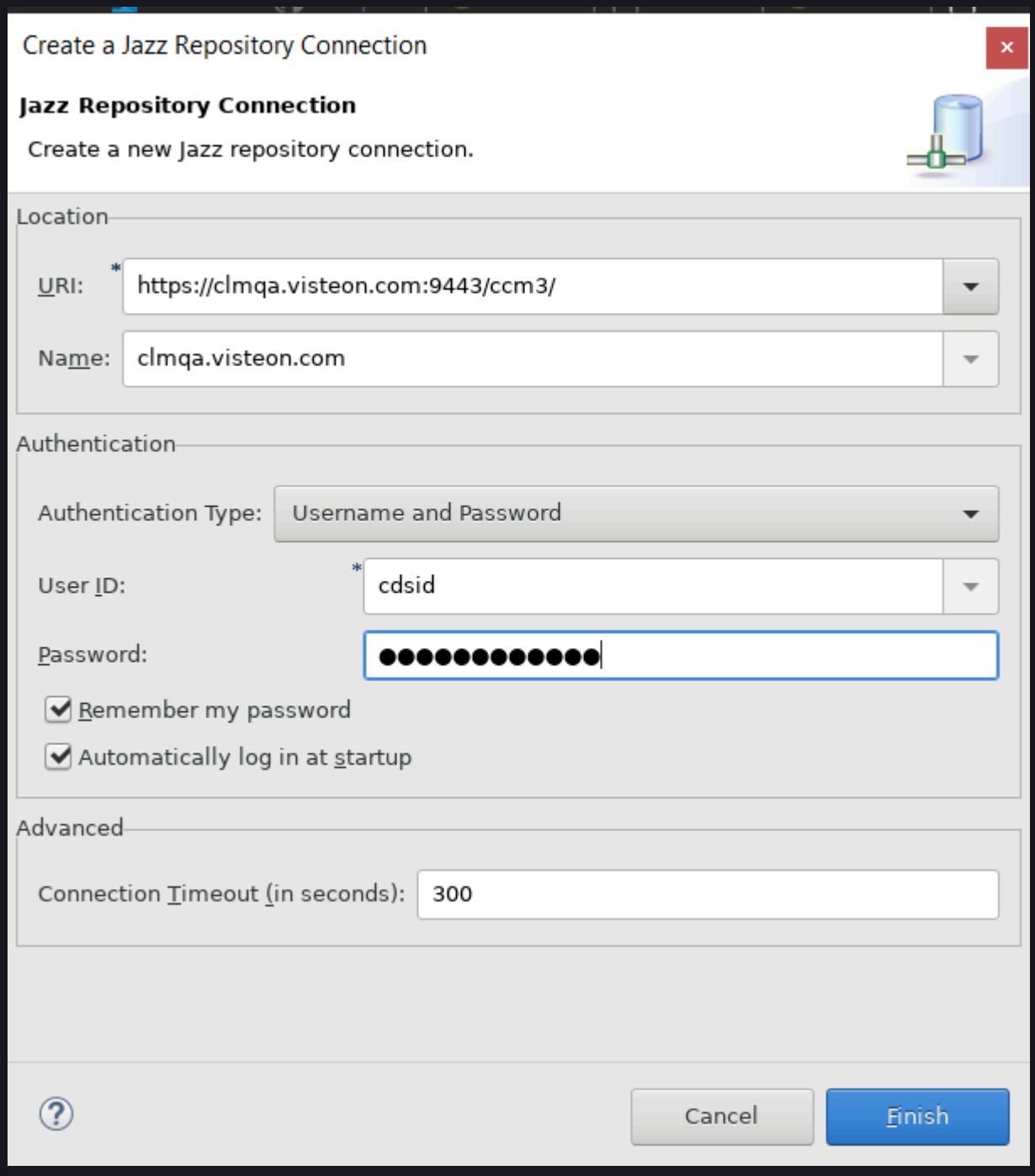
Create a Repository connection



Repository connection creation

- Make sure you've changed below Entries as per your project Need

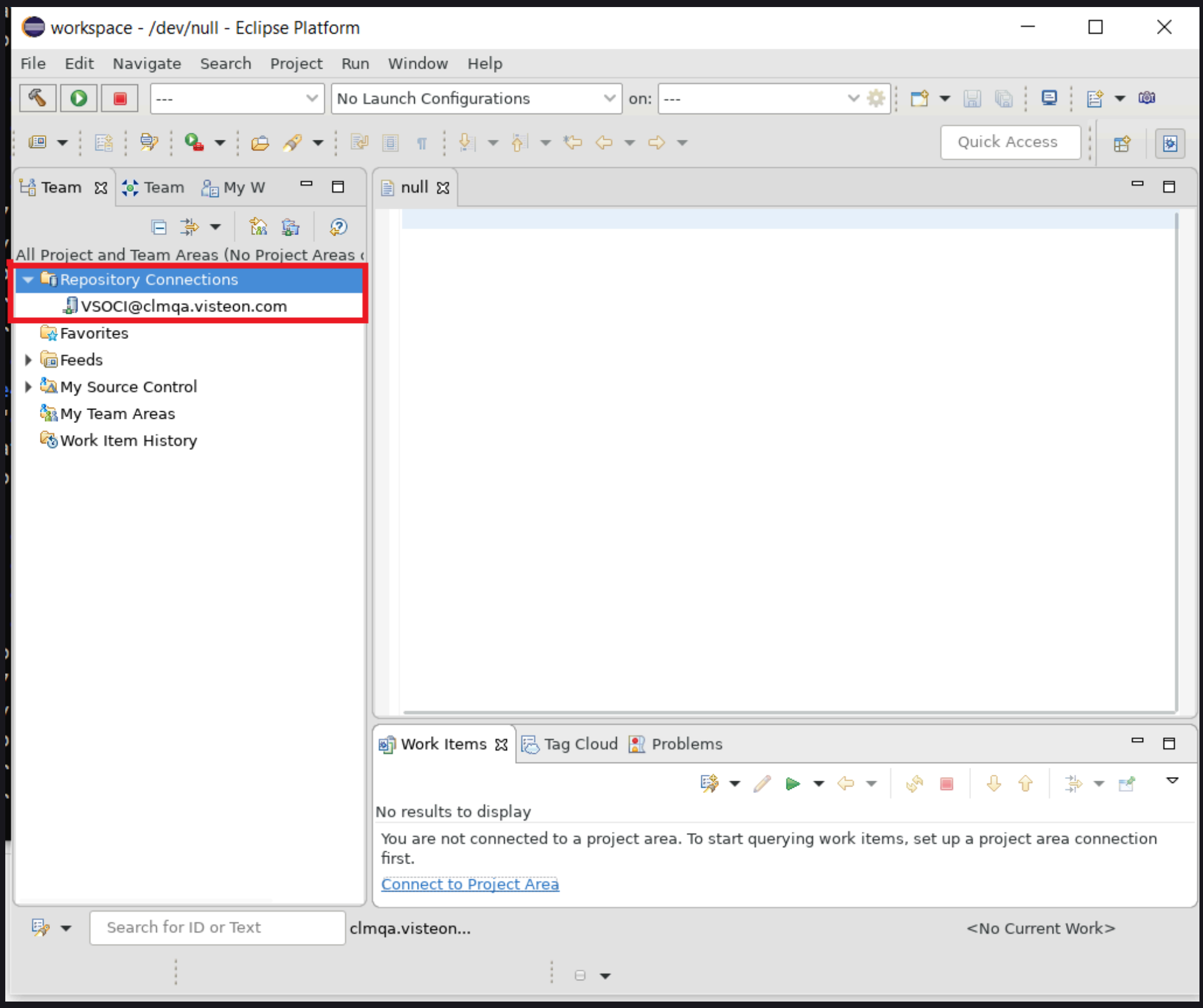
Example:-



Repository connection Established

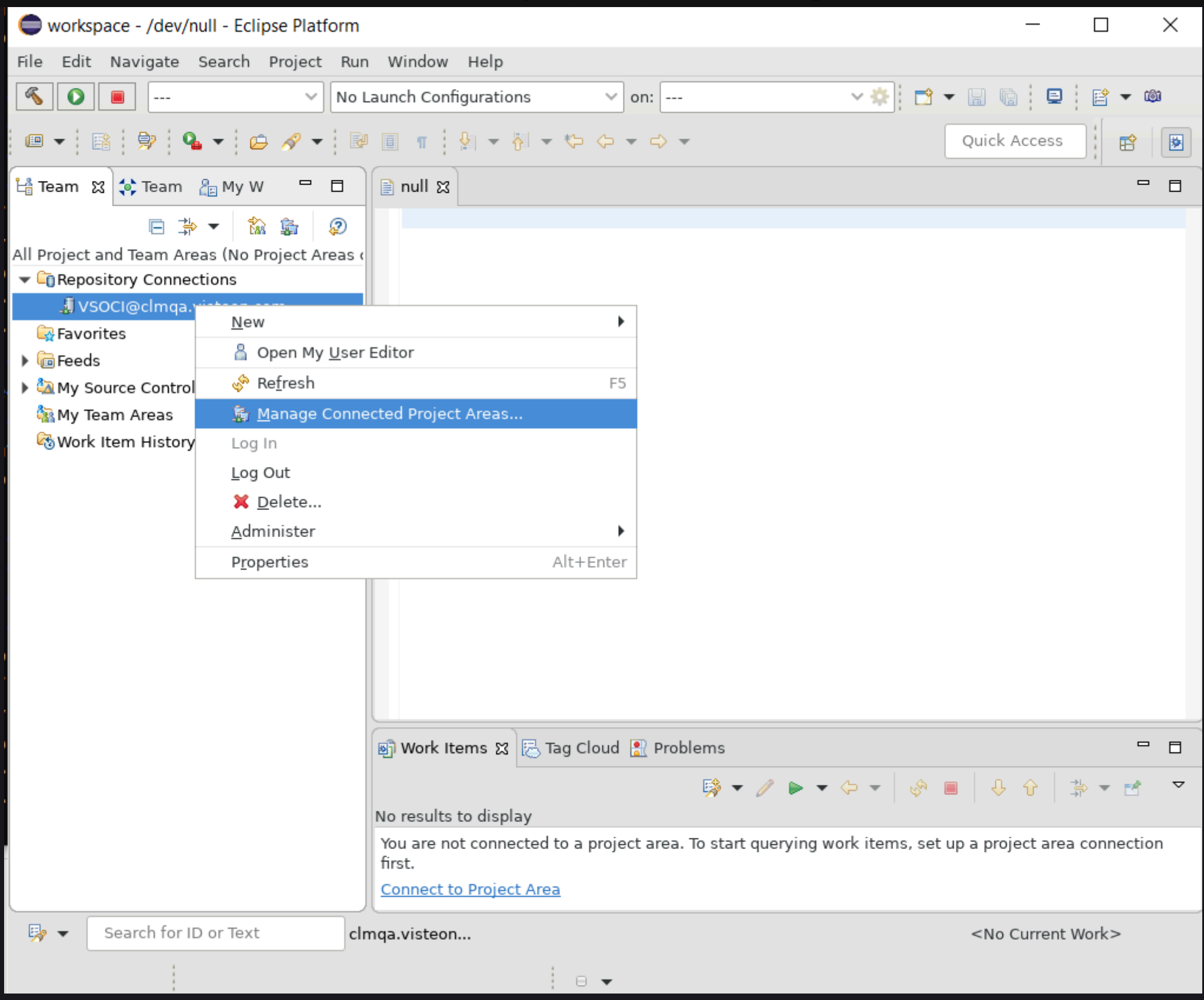
- Repository will be available under Repository connection once the repository connection is established.

Example:-



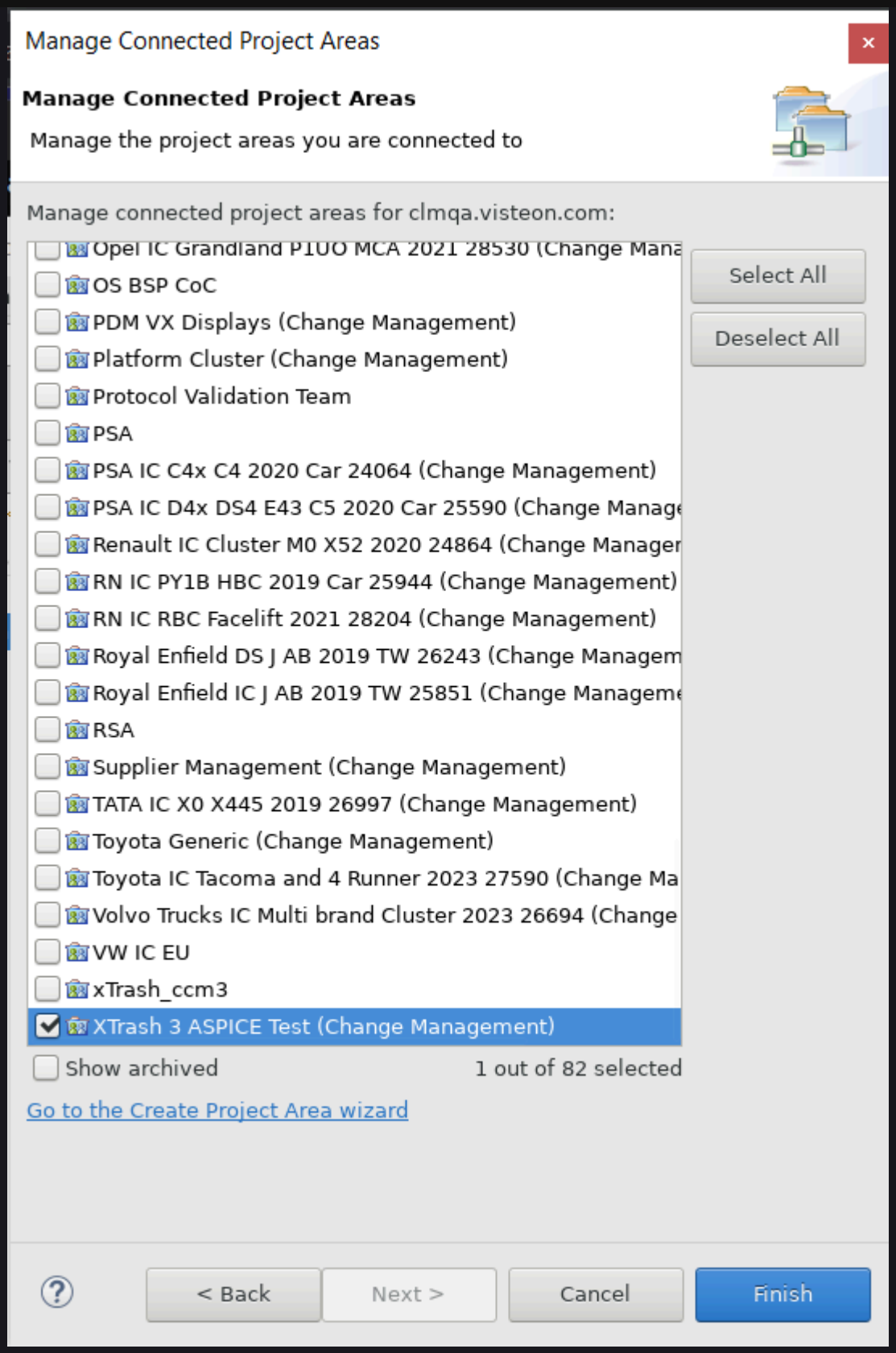
connect to project Area

- `click particular Repository connection` >> `Manage connected Project Areas`

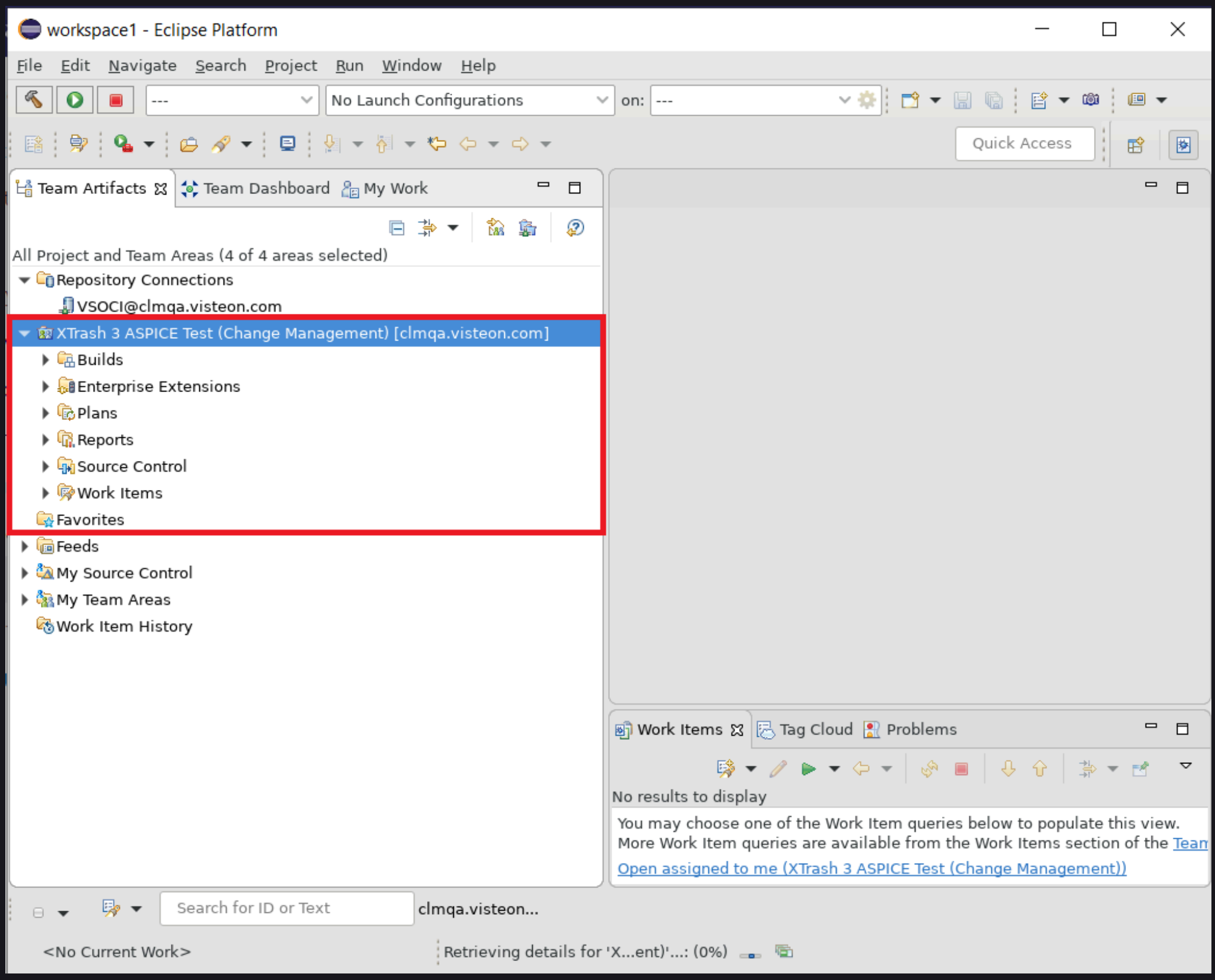


Select Project

- Select particular project >> Hit `Finish`



Connected Project



For More details Refer below link

- http://insight.visteon.com/wiki/Jazz_Source_Control(SCM)

Adding Remote Index URL Permanently(PIP Version >= 18)

Last edited by **VVADLAMU** 4 years ago

- If you plan to permanently add pip remote URL's

```
export location="glcc"     # You can Choose any one of the Locations from the below Mentioned available locations
python3 -m pip config set global.extra-index-url "https://jfrog.${location}.visteon.com/artifactory/api/pypi/pypi-virtua
python3 -m pip config set global.trusted-host "jfrog.${location}.visteon.com"

python3 -m pip install devNext --user
```

- Replace the {location} with the Locations mentioned in the Note section.
- Currently PIP packages are available in Locations:
  - glcc
  - bangalore
  - sofia

Support: If you've any Queries/Issues/Concerns, Please reach out to devNext.support@visteon.com

Access Control for Git

Last edited by **SMOHANA1** 1 year ago

This document provides an overview of Access Request feature in devNext to raise VESS requests for Git access. This is an optional feature in devNext for selective users. If you wish to leverage it, please contact devNext.support@visteon.com.

Supported OS: `Ubuntu 20.04`

Benefits of Access Control:

- **For the Team**:
  - Streamlines the process of granting repository access, ensuring that everyone has the necessary permissions.
  - Reduces overhead on verifying and validating the right repository.
  - Maintains all required repository details in one place.
  - Provides visibility for the whole team, showing which repositories they need access based on the information from the profile file.
- **For an Engineer**:
  - Simplifies the onboarding process by providing immediate access to the necessary repositories.
  - Eliminates the need for users to manually submit several access requests.
  - Saves time by eliminating the need for manual access requests.
  - Provides a seamless experience, allowing new users to focus on their tasks.

Enable Access Control for your Program

To enable Access Control, add the following section to your profile file:

```
[AccessControl]
GIT_SERVER = <Server>
GIT_GROUPS = <Groups>
GIT_ROLE = <Role>
```

- **Example**:

```
[AccessControl]
GIT_SERVER = bsp-os.git.visteon.com,rtc-proj.git.visteon.com
GIT_GROUPS = platform/bsp-os,platform/bsp-os/dijkstra,platform/bsp-os/programs,platform/bsp-os/turing,platform/integration,programs/ford
GIT_ROLE = Developer
```

Note 📢 : These values are case-sensitive, so ensure that values entered in the command line or profile match with those in the VESS portal.

For more information about key-value pair and its syntax, kindly check the sample profile.

Command Usage

Once you have configured workspace with `[AccessControl]` section, one can run the below command to raise git access to their project.

```
dn accessreq git
```

Options

```
Usage: dn accessreq git [OPTIONS]

  Helps to raise VESS request for Git Access.

Options:
  -t, --type [ADD|DELETE]         Specifies the request type, default [ADD].
  -c, --cdsids TEXT               Comma-separated CDSIDs. Ex: -c cdsid1,cdsid2
  -s, --server TEXT               Specify the Git server which you need
                                  access.
  -g, --groups TEXT               Comma-separated git groups. EX: -g
                                  platform/bsp-os,platform/bsp-os/dijkstra,..
  -r, --role [Developer|Integrator|Guest|Reporter]
                                  Specify one role at a time
  --name TEXT                     Name of the Workspace
  --help                          Show this message and exit.
```

Overriding Profile Values with Command Line Arguments

In certain situations, you might need to override the values specified in the profile. To accomplish this, use command line arguments that supersede the profile values.

Note 💡 : These values are case-sensitive, so ensure that the values entered in the command line or profile match those in the VESS portal.

Example:

```
dn accessreq git --type ADD --cdsids <cdsid1,cdsid2,cdsid3,..> --server <serve1,server2,..> --groups <group1,group2,..>
```

- **Example**:

```
smohana1@IND3Y87TG3:~/workspace$ dn accessreq git --type ADD --c cdsid1,cdsid2 -s *.git.visteon.com,*.git.visteon.com -g group1,group2 -r role
```

Q2A Forum: Q&A platform helps find answers and ask your questions, learn and share knowledge https://q2a.visteon.com.

Support: If you've any Queries w.r.t above mentioned steps, please reach out to devNext.support@visteon.com.

Add executable permissions for Hooks using Git

Last edited by **RKATTIMA** 2 years ago

It may be necessary to add execute permissions to custom build and hook scripts included in a project repo. This most frequently presents with the error “Permission denied” during Deploy or Package builds To add execute permissions to a specific file and commit the changes in Git, you can use the following command:

```
git add --chmod=+x -- <path/to/file>
git commit -m 'adding execute permissions'
```

Replace “<path/to/file>” with the actual path to the file you want to make executable. For example, if the file is located in the scripts directory, and its name is “scripts.sh” `git add --chmod=+x -- scripts/scripts.sh`

[Review] Working with GitLab Personal Access Tokens (PAT) in devNext

Last edited by **SMOHANA1** 1 year ago

Overview

This document explains what a GitLab Personal Access Token (PAT) is and provides guidance on how to configure and use PATs with devNext. **It is applicable for both Continuous Integration (CI) workflows using jenkins and general developer workflows**. The document also includes the necessary commands to enable and disable the PAT configuration.

Why GitLab Personal Access Token (PAT)?

A GitLab Personal Access Token (PAT) is a secure way to authenticate with GitLab and access API. **It acts as a replacement for username-passwords and OAuth tokens**, offering a higher level of security and control. PATs are used for authenticating API requests and accessing GitLab repositories.

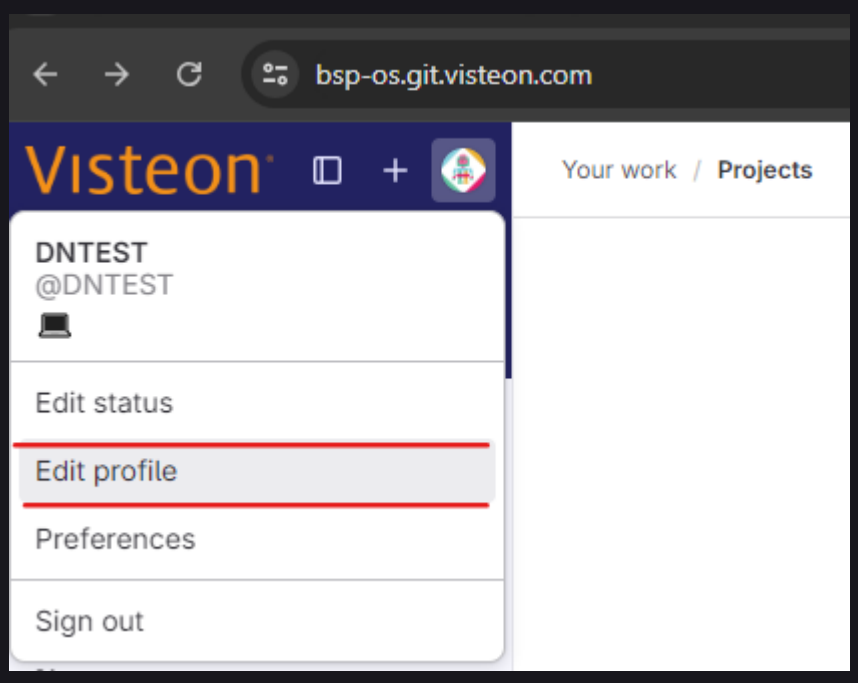
Advantages of Using PATs

- **Enhanced Security**: PATs can be scoped to limit access to specific resources.
- **Revocability**: PATs can be easily revoked if compromised.
- **Ease of Use**: They simplify the processes of automation and integration by providing an easy way to authenticate without the need for username and password.

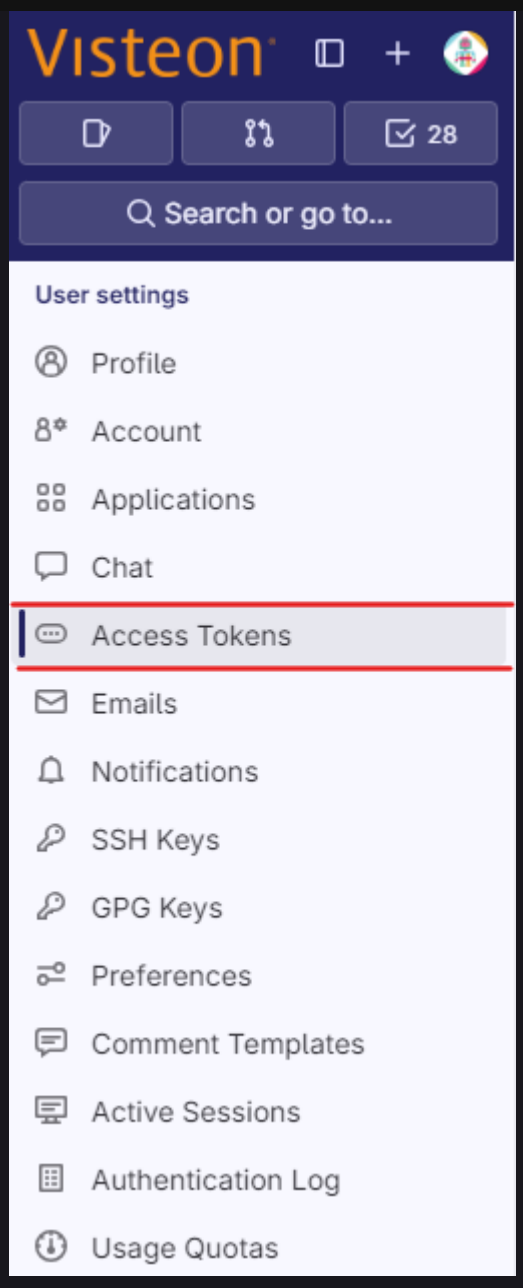
Using PAT with devNext

When working with devNext, having a PAT allows you to authenticate and interact with GitLab repositories securely. Follow the steps below to enable and configure PAT for devNext.

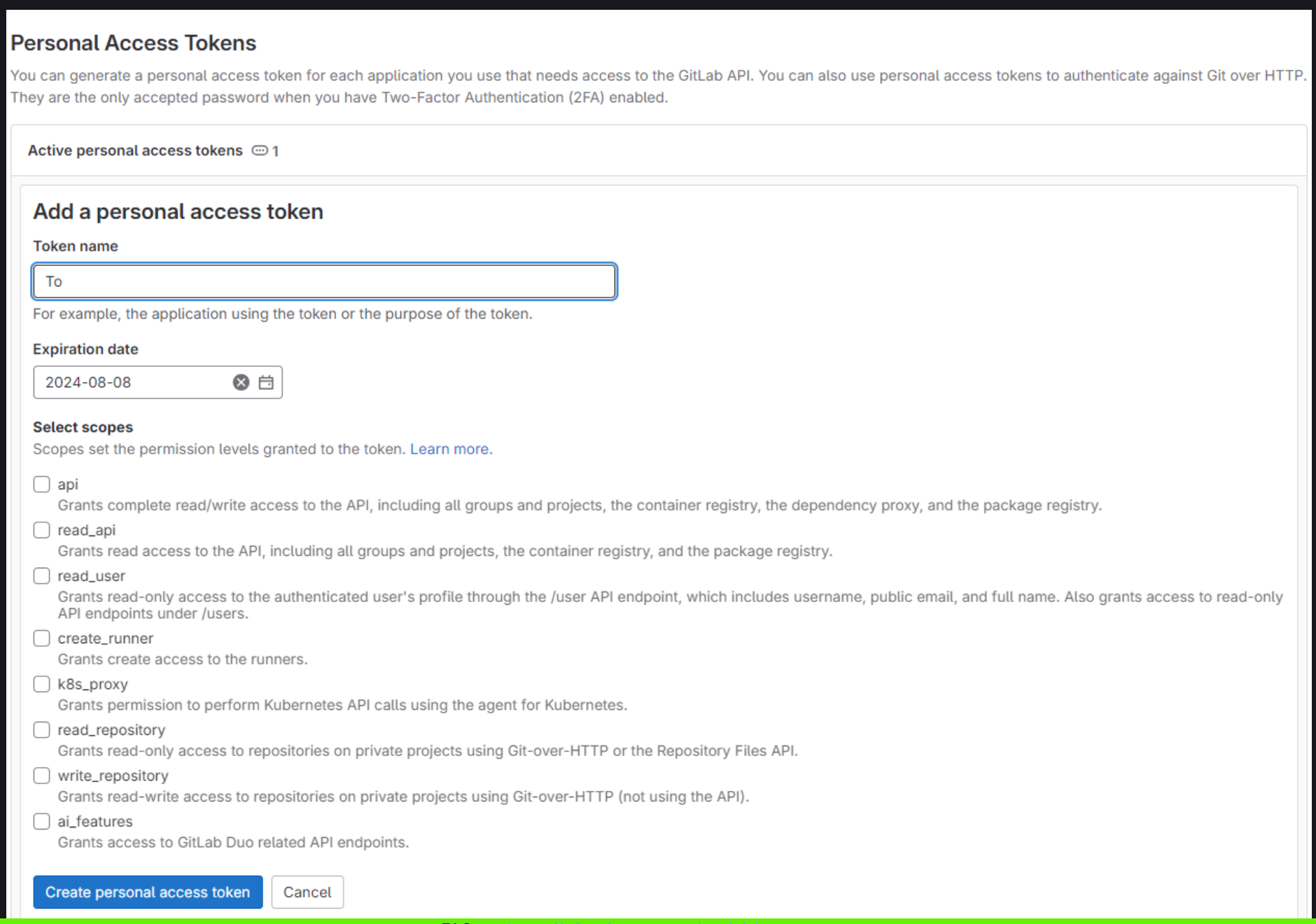
Enabling PAT for devNext

1. **Generate a PAT**: First, generate a PAT in GitLab with the necessary scopes. Follow the GitLab documentation to create a PAT: Creating a personal access token.
    - Login to the server
    - on the left sidebar, select your avatar
    - select `Edit Profile`
    - 

    - On the left sidebar, select Access Tokens.



- Select Add new token.
- Enter a name and expiry date for the token.
  - The token expires on that date at midnight UTC.
  - If you do not enter an expiry date, the expiry date is automatically set to 365 days later than the current date.
  - By default, this date can be a maximum of 365 days later than the current date.



- Select the desired scopes.
- Select Create personal access token.
- Save the personal access token safe. After you leave the page, you no longer have access to the token.

2. **Configure netrc to Use PAT**: Once you have your PAT (Personal Access Token), You need to write it as your machine credential in the `netrc` file. Refer to the document for detailed instructions on how to write machine credentials in `netrc` and this will be used by devNext to access the GitLab resources.

   Example Here, the user `devNext` and their PAT configuration for the servers are shown below:

   - Syntax : `machine {hostname} login {username} password {your-password}`
   - Example my server url **https://bsp-os.git.visteon.com** and user cdsid is **devNext**
   - `machine bsp-os.git.visteon.com login devNext password <PAT_HERE>`

```
machine git.visteon.com login devNext password git-vist-1862e6sfdvs2vesdty
machine eu.git.visteon.com login devNext password eu-simxiwozarxknair
machine blr.git.visteon.com login devNext password blr-git-suenzsin23sinsixnw9
machine jlr.git.visteon.com login devNext password glpat-82kwcseni9sne9wn9
machine bsp-os.git.visteon.com login devNext password bsp-os-1dAESYMGo6XdsZshJzzv
machine rtc-proj.git.visteon.com login devNext password rtc-proj-zrunixmwuzp1894ssd
```

Command to Enable PAT for devNext

```
git config --global devnext.auth PAT
```

This command sets the `devNext.auth` configuration to use your PAT, allowing devNext to access the GitLab resources using the PAT for operations like configure workspace, gitfetch and gitupdate.

Command to Unset PAT Configuration for devNext

```
git config --global --unset devnext.auth
```

This command removes the `devnext.auth` configuration, stopping devNext from using the PAT for authentication.

Q2A Forum: Q&A platform helps find answers and ask your questions, learn and share knowledge https://q2a.visteon.com

Support: If you've any Queries w.r.t above mentioned steps, please reach out to devNext.support@visteon.com

Home

Last edited by **Reddy, Viswanatha (V.P.)** 1 month ago

What is devNext?

> devNext v4.2.0 is a Workspace Management Tool which helps developers to quickly setup and manage different projects in a single machine. All the configuration happens through an INI file and access to all the workspaces is centralized.

Here are the salient features of devNext:

▼ Workspace Isolation

All the required toolchains are managed through configuration file. Your system can have different versions of same tool, devNext makes sure that required version of the tool is configured for the workspace.

▼ Centralized Workspace Management

Why to remember all the workspaces available in your system, when devNext can do that for you. You can have multiple workspaces in your system belonging to different projects. All these workspaces can be managed through devNext. Developer can list, add, delete, modify and activate any workspace from anywhere in the system.

▼ Build Aliases

Forget about the long build commands that a developer needs to remember. Developer can now create aliases for the build commands. Aliases will be part of the configuration file and can be shared across Team. Any number of aliases can be created provided all the aliases are unique.

▼ Faster builds

Gone are the days when user had to wait for the builds to get completed. With devNext you get minimal Virtualization which helps to get the best performance out of the system. Now your system will be utilized completely for the tasks you need.

▼ Tested and Supported Platforms and software for devNext

OS with version	Python Version	PIP Version	Repo Tool
Ubuntu 20.04	v3.8.10	v23.3.2	Repo_Launcher 2.29.9
Ubuntu 22.04	v3.10.6	v23.3.2	Repo_Launcher 2.29.9
Ubuntu 24.04	v3.12.3	v23.3.2	Repo_Launcher 2.29.9
Windows 10	v3.12.8	v23.3.2	Repo_Launcher 2.29.9
Windows 11	v3.12.8	v23.3.2	Repo_Launcher 2.29.9

For older program's backward compatibility, support is provided for previous versions of devNext on:

OS with version	Python Version	PIP Version	Repo Tool
Ubuntu 18.04	v3.6.9	v21.3.x	Repo 2.8

Getting Started:

1. Setting Up WSL2
2. Automation Script for Pre-Requisites Setup

   (or)
   devNext Prerequisites (Manual)

   Single CONAN_USER_HOME for Build Servers
3. devNext Installation and Configuration of Workspace
4. devNext Upgradation
5. Instructions for New Release Candidate install for Conan 2.0 on Ubuntu 24.04
6. devNext Support Requests

FAQs at https://g2a.visteon.com/tag/git/

❗ DEPRECATION WARNING:

- Conan 1.x is deprecated and is no longer recommended. The upcoming devNext release will come with default Conan v2.

Training and Video

- 📎 Gitlab Presentation Deck
- devNext Training Video
- Code Quality with devNext using Klocworks - Training
- Tasking Compiler Training

Q2A Forum: Q&A platform helps find answers and ask your questions, learn and share knowledge https://q2a.visteon.com
Support: If you've any Queries w.r.t above mentioned steps, please reach out to 📧 devNext.support@visteon.com