Azure DevOps Integration with Denodo

Revision 20210913

NOTE
This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Denodo North America & APAC: 525 University Avenue, Suite 31, Palo Alto, CA 94301. USA
Denodo Iberia & Latino América: Montalbán 5, 28014 Madrid, Spain
Denodo EMEA: 21st Floor, Portland House, Bressenden Place, London SW1E 5RS. UK
Denodo DACH: Karlstraße 10, 80333 München. Germany
www.denodo.com

CONTENTS

1 CONTENT...3

2 INTRODUCTION..4

3 AZURE DEVOPS...5

3.1 CONFIGURING AZURE DEVOPS FOR VCS IN DENODO VIRTUAL DATAPORT 5

3.2 CONFIGURING AZURE DEVOPS FOR VCS IN SOLUTION MANAGER......8

4 REFERENCES..10

1 CONTENT

This document describes the detailed steps to integrate Azure DevOps for Version Control System in Denodo. Currently, the Denodo Platform supports GIT, Subversion and Microsoft TFS for the Version Control System and **GIT** can be used to configure the connection to the Azure DevOps repository.

2 INTRODUCTION

Version Control Systems helps managing workflows and changes to source code over time. In Denodo, VCS helps to manage the metadata of the Denodo elements involving developers working simultaneously on the same elements.

The VCS integration is supported in Denodo for both Virtual DataPort and the Solution Manager. In Virtual DataPort servers, it can be used for integrating the work done simultaneously by different members of the development team. In the Solution Manager, it can be used to save a backup file of the environments after deployment, both for successful and unsuccessful deployments.

3 AZURE DEVOPS

Microsoft Azure DevOps provides version control, reporting, requirements management, project management, automated builds, testing and release management capabilities.
Azure DevOps integration provides repositories for an integrated and collaborative environment supporting continuous integration for managing workflows. The Denodo Platform integration with Azure DevOps helps managing and storing metadata in Azure repositories via VCS operations.

3.1 CONFIGURING AZURE DEVOPS FOR VCS IN DENODO VIRTUAL DATAPORT

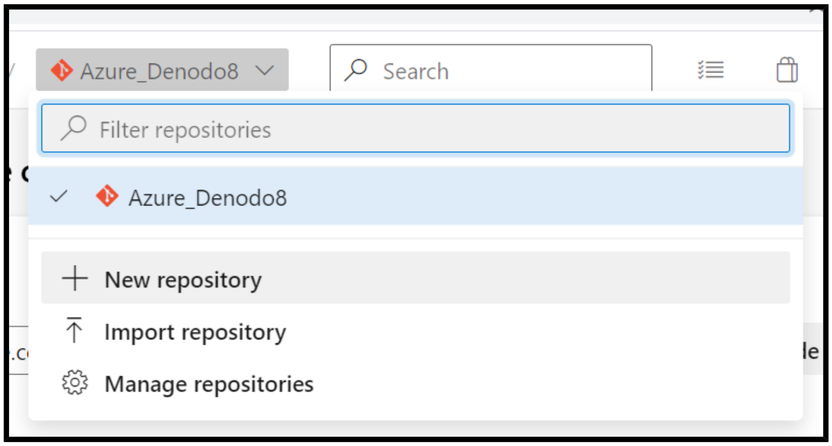
This section explains how to configure Azure DevOps for Version Control in Denodo Virtual DataPort.

Prerequisites
The user must have an organization created in Azure DevOps and a project created. If you have not created an organization adn project, follow the steps in [Create an organization or project collection] and [Create a project in Azure DevOps] sections of the Azure DevOps documentation.

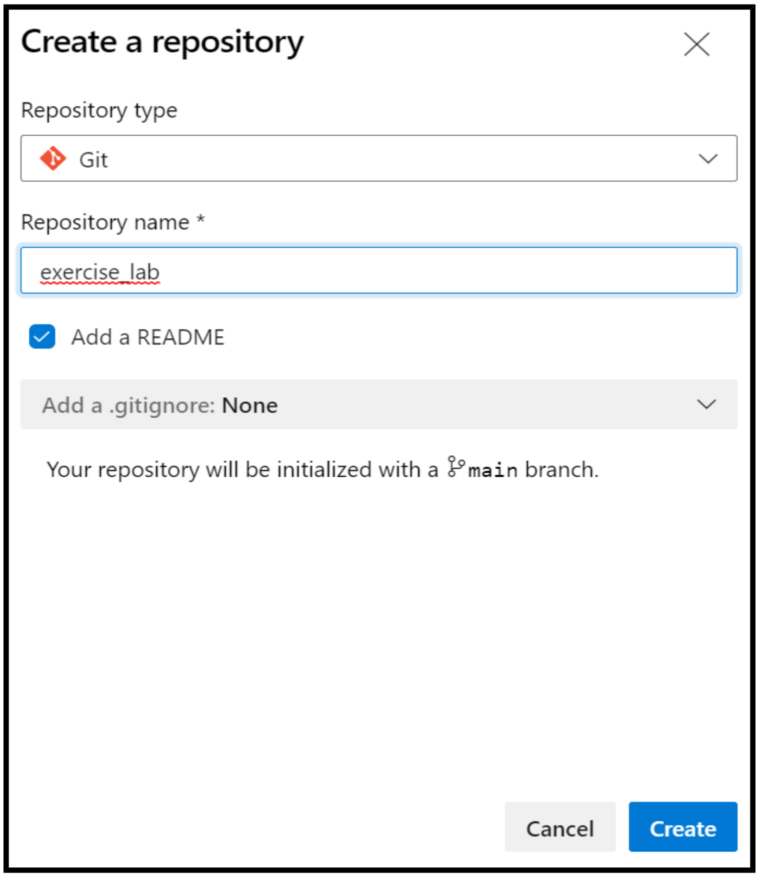
- Sign in to your organization and open the created project. The default URL to sign in to your organization is "`https://dev.azure.com/{yourorganization}`".

  For instance: `https://dev.azure.com/Denodo8`

- Navigate to the **Repos** page of the project. Create a new repository by clicking on the repository dropdown at the top and select **`New repository`**.



- In the **`Create a new repository`** window, select the **`Repository type`** as **`"Git"`** and enter the Repository name. Click on **`Create`** and the new repository will now be created.



Create a repository

Repository type: Git

Repository name *: exercise_lab

☑ Add a README

Add a .gitignore: None

Your repository will be initialized with a `main` branch.

- For configuration details to integrate the Azure DevOps repository in the Virtual DataPort Administration tool, click on the **`Clone`** option of the created repository in the upper-right corner of the `Files` window.

  The **`Clone Repository`** window contains the **URL** for VCS configuration and the user authentication credentials. The URL has the following format:

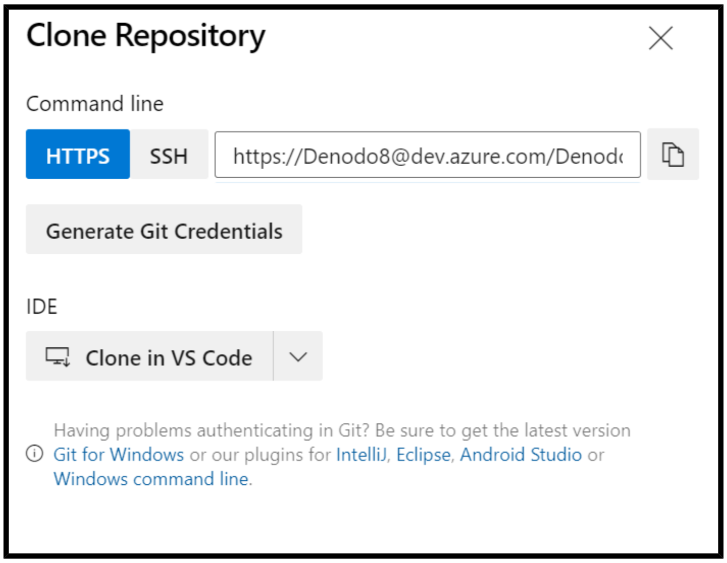
  ```
  https://<organization_name>@dev.azure.com/<organization_name>/<project_name>/_git/<repository_name>
  ```

  For instance:

  ```
  https://Denodo8@dev.azure.com/Denodo8/Azure_Denodo8/_git/exercise_lab
  ```

  The **`username`** and **`PAT token`** are necessary for configuring VCS on the Denodo side and can be obtained by clicking on the **`Generate Git Credentials`** option.

  The **`Personal Access Token`** (PAT) is used as an alternate password to authenticate into Azure DevOps and is essential to connect to Azure DevOps from Denodo. This access token value needs to be specified in the password section of the Denodo VCS configuration window. As the PAT tokens are not stored, you can copy the token for future needs or generate a new PAT token.



Clone Repository

Command line

https://Denodo8@dev.azure.com/Denod

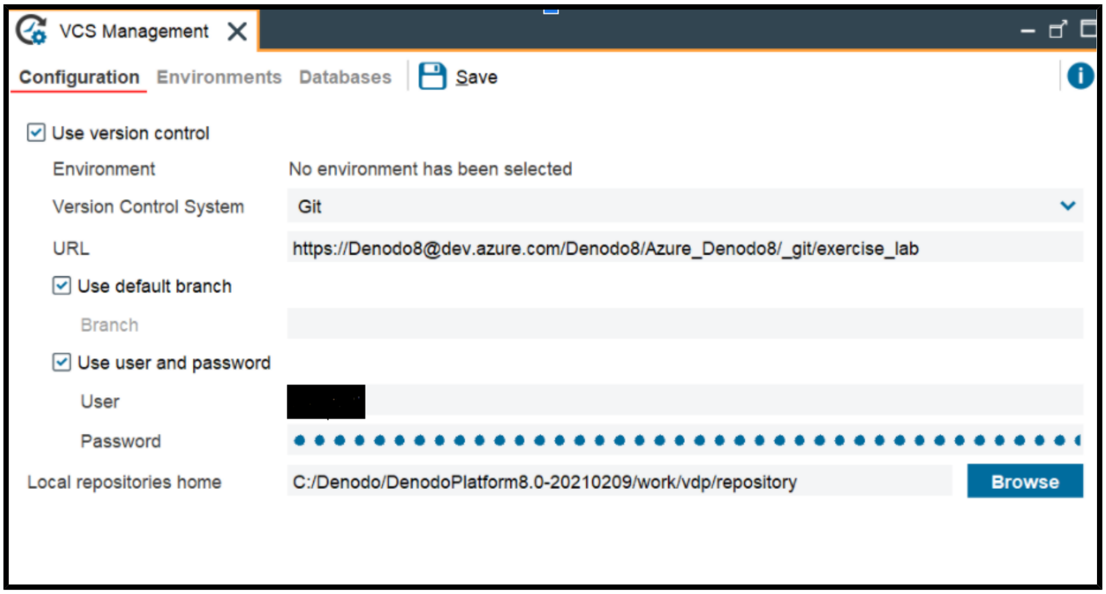
Having problems authenticating in Git? Be sure to get the latest version Git for Windows or our plugins for IntelliJ, Eclipse, Android Studio or Windows command line.

- Now, open the Virtual DataPort Administration tool and navigate to the **"Administration > VCS Management"** section.

  NOTE: The VCS configuration can be done only by the administrator. Hence, it is necessary to log in to Virtual DataPort as administrator.

- Under the **Configuration** tab, configure the following parameters:

  - **Use version control** - Enable this option.
  - **Environment - No environment has been selected** for the first time you configure the VCS integration, otherwise it shows the selected environment.
  - **Version control system** - Git.
  - **URL** - specify the URL from the Azure DevOps Clone Repository window.
  - **Use default branch** - Enable this option or enter the branch to be used by default.
  - **Use user and password** - Enable this option. Enter the credentials obtained from the Clone Repository window of Azure Devops.
    - **User** - Enter the username.
    - **Password** - Enter the PAT token (Personal Access Token).
  - **Local repositories home** - Path where the server will create the local repositories for the databases with VCS enabled. You can change the directory by clicking on the **Browse** option and specify the new path.
  - Click on **Save**.



☑ Use version control

Environment: No environment has been selected

Version Control System: Git

URL: https://Denodo8@dev.azure.com/Denodo8/Azure_Denodo8/_git/exercise_lab

☑ Use default branch

Branch:

☑ Use user and password

User: [redacted]

Password: ●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●●

Local repositories home: C:/Denodo/DenodoPlatform8.0-20210209/work/vdp/repository

The Azure DevOps repository is now integrated with the Virtual DataPort server for VCS management.
You can now create environments and configure Databases to check it in a VCS server or import a Virtual DataPort database created by another user, from the VCS server.

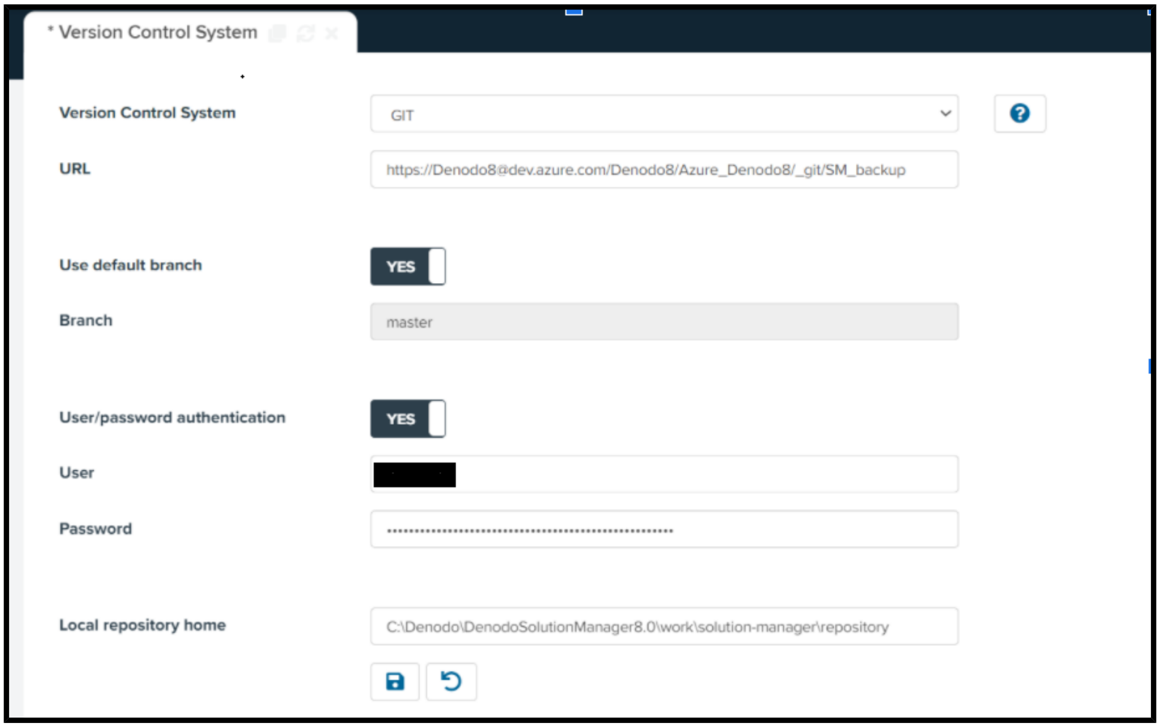
3.2 CONFIGURING AZURE DEVOPS FOR VCS IN SOLUTION MANAGER

To configure Azure DevOps for Version Control System in the Solution Manager, follow the initial steps for the Azure DevOps repository setup (create a new repo for SM) mentioned in the **Configuring Azure Devops for VCS in Virtual DataPort** section and then follow these steps:

- Log in to the Solution Manager Administration Tool and navigate to the **"Configuration > Version Control System"** section.
  To fill in configuration details under this section, open the Azure DevOps repository to be integrated for VCS setup in Solution Manager. Click on the **Clone** option which opens the **Clone Repository** window with the required details.

- Provide the values for the subsequent parameters:
  - **Version Control System** - GIT.
  - **URL** - Enter URL from the Clone Repository window.
  - **Use default branch** - Branch to be used to save backup.
    - YES - to use the default remote branch.
    - NO - to specify ta different branch name.
  - **User/password authentication** - YES.
  - **User** - username.
  - **Password** - PAT generated from the `Generate Git Credentials` option.
  - **Local repository home** - Local repository path to save the backup.

- Click on **Save**.



* Version Control System

Version Control System	GIT
URL	https://Denodo8@dev.azure.com/Denodo8/Azure_Denodo8/_git/SM_backup
Use default branch	YES
Branch	master
User/password authentication	YES
User	[redacted]
Password	••
Local repository home	C:\Denodo\DenodoSolutionManager8.0\work\solution-manager\repository

The Azure DevOps repository is now configured for the Solution Manager VCS. After this set up, change the environment configuration before a deployment so the Solution Manager stores a backup on the remote repository for every action of revision deployment to that environment.

NOTE: In these steps we have explained how to configure Azure DevOps using Personal Access Tokens (PATs). It is also possible to use SSH keys to configure the integration with Azure DevOps, see Virtual DataPort Server Configuration for more information.

4 REFERENCES

Create a new Git repo in your project
Version Control Systems Integration
VCS Configuration
VCS Operations for GIT
VCS Configuration