**VSS to GIT – STEPS AND CHANGES DURING MIGRATION**

**Software Used:** Castellum by Abstrakti (to transfer files from VSS to GIT), Visual Source Safe (VSS), Visual Studio (IDE for GIT)

**Location:** Software to be used on Ausdbarch03 server

**Functionality:** It transfers specifically selected repositories, covering all the content (paid version)

**Source:** We have the software downloaded on remote desktop (aus-dbarch3).

Important Notes for tool:

* Remember to select Simple Migration when migrating files using the tool

**User Manual for Software:** <https://www.abstrakti.com/Products/Castellum/UserManual>

**Important Settings:**

* Remember to log in under your username (not admin)
* Leave everything on the software at its default setting apart from the merge settings
* Set merge settings to never under both columns.

**Push and Pull:** We pull repositories from Azure DevOps using visual studio, import files into it using castellum, and then push the repositories back onto Azure DevOps. To pull and push the repositories, we use the simple pull and push features under the git tab in VS (Figure 1).

A screenshot of a computer

Description automatically generated with medium confidence

Figure 1: Push and Pull Options on Visual Studio.

**Changes/Migration So Far:**

1. 06-22-2022: **DBA Central** transferred from VSS to GIT using Software.
2. Transferred using trial version
3. Destroyed files present
4. Watermark left on commits section of Azure DevOps.
5. Pushed onto Azure DevOps once transferred.
6. 06-27-2022: **New DBA Central Code** Pushed to Azure DevOps
7. Transferred by emptying previous DBA Central file in VS and copying new contents to it
8. Above led to changes, which were then committed
9. Commit was pushed to online git data base
10. Certain files were restricted from being transferred as they were unnecessary. Using gitignore the following type of files were ignored: .svn/, vssver.scc
11. 06-29-2022: **DBA Central Transferred** from VSS to GIT on VS. (**Transfer without Destroyed Files**)
12. Data was committed and then pushed onto Azure DevOps.
13. Updated code was also pushed onto the repository.
14. 07/ 14/2022**: B2B, DBA Projx, dbarchive, eda, custom objects, and translations** were transferred from VSS to GIT using the tool
15. Simple Migration was used for all
16. Files were then pushed onto their Azure Repositories using Visual Studio.
17. **DBA Central has 2 branches on Azure** – Dev and Prod. DataDictionary, dbarchive, and DBRequest were then transferred to become individual repositories: dbacentral.datadictionary, dbacentral.dbarchive, and dbacentral.dbrequest.
18. **Pipelines were created for the dev branches of dbacentral.datadictionary, dbacentral.dbarchive, and dbacentral.dbrequest**. The pipeline’s function was to transfer the builds to a file on dbarch03.

**Software Step by Step Guide**

1. When you first open Castellum, you need to create a folder to keep all of your metadata. To do this create and select the folder in a location of your preference.

Graphical user interface, text, application, email

Description automatically generated

1. Always delete the packages, as you may end up with duplicate meta data.

Text

Description automatically generated

1. Find and select the vss file on your computer and enter your username. The password is not required.

Graphical user interface, text, application

Description automatically generated

1. Select the repository you want to transfer. You can also select specific files within the repository. In this example, DBA Projects App has been selected

Graphical user interface, text

Description automatically generated

1. Keep the settings as shown in the screenshot below.

Graphical user interface, text, application

Description automatically generated

1. Select the git repository on your local that you aim to move the vss repo to. In this example, the git repository is called “Castellum-Demo-Suvir.Wadhwa. Keep all the other settings the same as the screenshot below.

Graphical user interface, text, application

Description automatically generated

1. Select target GIT path in the folder you selected earlier

Graphical user interface, text

Description automatically generated

1. Select option 2 to preserve the same path structure

Graphical user interface, text, application, email

Description automatically generated

1. The files will start to transfer, and you may be prompted with some messages depending on the repository you are moving. Once these messages are managed, your repository will transfer, and you’ll find it in the GIT folder you selected earlier.

**Working with online Repo on Local**

1. To clone the repo, go to the git menu on visual studio and select clone. If you already have the repo cloned, then just select pull in in git menu to get the updated repo version.

Graphical user interface, application, Word

Description automatically generated

1. When you get to the below window, enter the clone link and the local path you want to store the repo at. You can also select the Azure DevOps option below and clone it from there.

Graphical user interface, text, application, email

Description automatically generated

1. Now that you have the repo, you can make changes to it. Once those changes are made, you need to commit them using the box as shown in the image below.

Graphical user interface, application, Word

Description automatically generated

1. Now you need to push the repository back to Azure using the push option shown in the git menu.

Note: You may be working on your on your own branch before pushing, in this case remember to commit your changes to that branch and then once on Azure, pull your temporary branch into the main branch using a pull request.

Commit Page on Azure DevOps

A screenshot of a computer

Description automatically generated

Pipeline Section on Azure DevOps

