How to Use AutoStashDep - Automation of repository commits and file transfers to jump server

Created by Dante Sebastian, last modified on Jul 12, 2021

Section 1: Prerequisites

Prerequisite 1: Powershell 6 or above should be installed in the user's personal machine

Rationale: Some of the commands inside the predeploy.ps1 file are only executable only by Powershell 6 or above.

- Press Windows key then type "Powershell". Run the application as Administrator. Enter the following commands
  - iex "& { $(irm https://aka.ms/install-powershell.ps1) } -UseMSI"
  - Get-Module -ListAvailable PackageManagement, PowerShellGet
  - Install-PackageProvider Nuget –Force
  - Install-Module –Name PowerShellGet –Force
  - Set-ExecutionPolicy RemoteSigned
  - Install-Module –Name PowerShellGet –Force -AllowClobber
  - Update-Module -Name PowerShellGet
- Once installed, open cmd.exe and enter "pwsh". The following should be displayed to indicate that Powershell 7 has been successfully installed.

```
C:\Windows\system32>pwsh
PowerShell 7.1.3
Copyright (c) Microsoft Corporation.

https://aka.ms/powershell
Type 'help' to get help.
```

Prerequisite 2: Downloading the AutoStashDep

- Download the AutoStashDep.zip in the attachments tab of this wiki page.

Prerequisite 3: Setting of permanent variables in your Windows OS.

Rationale: In this way, the user won't need to use the 'cd' command because the directories of each path will be pre-configured after accomplishing this prerequisite.

- Open the AutoStashDep folder, then open the permanentVariableSetup.bat
- The first input prompt would be for the MFS_DEPLOYMENT_PATH. Input the Windows path where you prefer to store your FTS-XXX folders. This would also be the path for your AutoStashDep folder, which contains the files that needed to make the automation work.
- The second input prompt would be for the MFS_STASH_PATH. Input the Windows path where the Stash folder located/to be stored.
- The third input prompt would be for the MFS_MFC_PATH. Input the Windows path where the managed-fix-configs is located/to be stored.
- The fourth input prompt would be for the MFS_STASH_URL. Input the URL of the managed-fix-configs (https://stash.orcsoftware.com/scm/imf/managed-fix-configs.git)
- The fifth input prompt would be for the WINSCP_PATH. Input the Windows path where your WinSCP is installed.
- The sixth input prompt would be for the AUTOSTASHDEP_PATH. Input the Windows path of your AutoStashDep

Your variable configurations can be viewed by running the permanentVariableView.bat file. The variable configuration should resemble to something like the following.

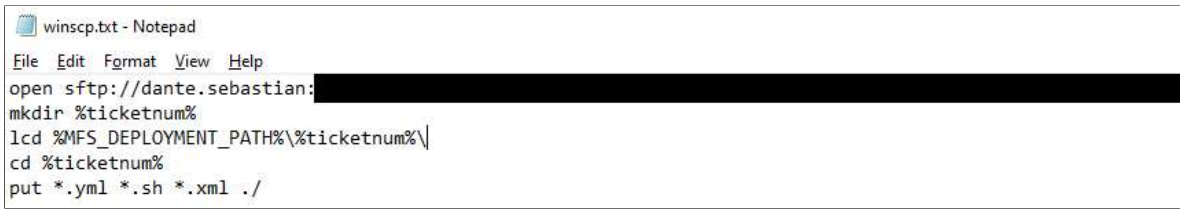
```
C:\Windows\system32\cmd.exe
ECHO is off.

The following variables have been set.

MFS_DEPLOYMENT_PATH     C:\Users\dante.sebastian\Desktop\deployments
MFS_STASH_PATH          C:\Users\dante.sebastian\Desktop\Stash
MFS_MFC_PATH            C:\Users\dante.sebastian\Desktop\Stash\managed-fix-configs
MFS_STASH_URL           https://stash.orcsoftware.com/scm/imf/managed-fix-configs.git
WINSCP_PATH             C:\Program Files (x86)\WinSCP
AUTOSTASHDEP_PATH       C:\Users\dante.sebastian\Desktop\deployments\AutoStashDep

Press any key to continue . . .
```

Prerequisite 4: Setting of the host key to be used by the WinSCP

Rationale: This would eliminate the repetitive login to the jump server. Think of it as something like 'keep me signed in' feature, but only for the purpose of transferring files.

- Login to the jump server via WinSCP.
- In the Menu tab, navigate to Session > Generate Session URL/Code...
- Select the Script tab and set the Format dropdown option to 'Script file'. The user would should see something like the following (some of the info has been censored due to privacy)

- Copy the whole information right after 'open'
- Navigate to the AutoStashDep folder. Open the winscp.txt file
- In the first line, right after 'open' paste the copied text. It should be something like the following.



- Save the file.

Section 2: Phases

Phase 1: Cloning the latest managed-fix-configs repository/Deleting the previous local clone.

- Just run the [1] delete-cloneRepo.bat file

Phase 2: Checking, staging, committing, and pushing your local changes to the remote master repository

- At this point, it is assumed that the user has done the changes to the local clone. Once done, run the [2] autoCommit.bat file.
- The first input prompt would ask the user of the ticket number associated with the commit/deployment. Once inputted, a folder inside MFS_DEPLOYMENT_PATH with the inputted ticket number would be created.
- 'git pull', 'git diff', 'git status' commands would then be auto-executed at this point. The user must review the outputs on the batch program before proceeding the staging phase.
  - Note: If there are a lot of changes, the git status would display a long text output. To proceed, just type :q! or :q on the batch program.
- The second input prompt would ask the user if the staging phase shall be proceeded on. (1 as proceed, 0 as cancel)
- The third input prompt would ask the user if the committing phase shall be proceeded on. (1 as proceed, 0 as cancel)
- The fourth input prompt would ask the user for the commit message.
- The fifth input prompt would ask the user if the adding phase shall be proceeded on. (1 as proceed, 0 as cancel)

At this point, git_status_output.txt would be created inside the ticket folder which would list all the changes made to your local repository. This information would be used in the next phase.

Phase 3: Copying the changes to the ticket directory inside the MFS_DEPLOYMENT_PATH

- Just run the [3] preDeploy.bat file.

the user's directory in the jump server.

- Run the [4] deploy.bat file and input the ticket number. Changes can be validated in WinSCP or PuTTY, or for a very quick check, just run viewDeployedFiles.bat

(wiki is in work in progress state)

No labels