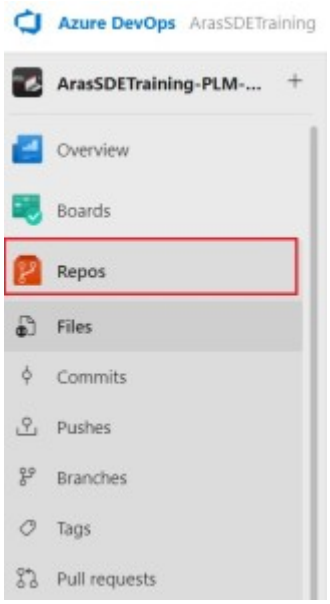
Update Repository to Use Single Package

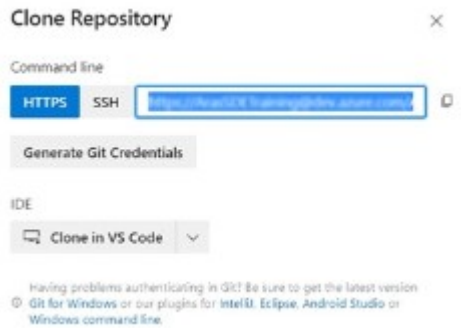
The latest release of Aras DevOps offers all essential Core **Packages** into a single NuGet **Package**. This simplifies the process of creating repositories and ensures that the Core **Packages** work seamlessly together.

The following steps outline the process of updating the Work Repository to use the Single **Packages**:

1. On **Azure DevOps**, select **Repos**.



2. Click **Clone** to clone the Repository that needs to be updated. The **Clone Repository** dialog box appears with the Repository's clone URL.

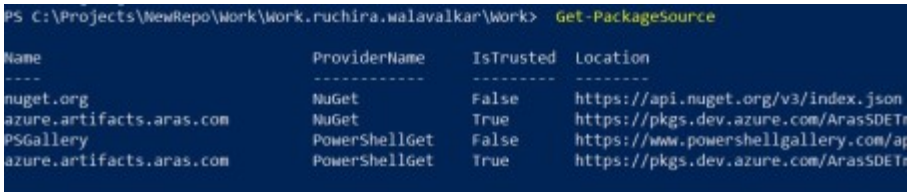


3. Copy the clone URL (HTTPS or SSH). An example URL: https://dev.azure.com/{organization}/{project}/git/{repository}. Use any Version Control Tools required to clone the Repository.
4. Find the option to clone or create a new Repository in the Version Control Tool's interface.
5. Paste the clone URL copied from the Azure DevOps project.

Browse to the destination directory to clone the Repository.

Optional: Depending on the tool used, additional configuration options are available during the cloning process. This could include selecting **Branches**, specifying authentication credentials, or choosing the desired clone depth.

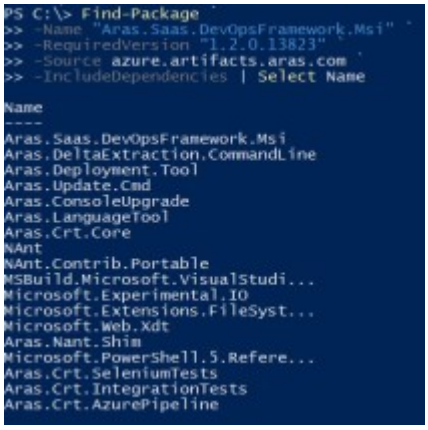
6. Click **Clone** within the Version Control Tool.
7. Open **Windows PowerShell** as Administrator.
8. Run **./repo-init.ps1** command for initializing the Repository with Initial **Packages**.
9. To identify the source name of the **Package** that points to the NuGet feed containing the Single **Package**, run the **Get-PackageSource** command.



For the correct update, a list of Single **Package Dependencies** is required. The list will be compared to the individual Packages specified in the AutomatedProcedures/tools/packages config file.

10. Remove-Module -Name “Aras.Devops” -Force
11. Install-Module -Name “Aras.Devops” -RequiredVersion <new version>
12. Import-Module -Name “Aras.Devops” -RequiredVersion <new version>

13. To get dependencies of the NuGet Package, run Find-Package -Name "Aras.Saas.DevOpsFramework.Msi" -RequiredVersion "<single_package_version>" -Source <package_source_name> -IncludeDependencies | Select Name



14. Navigate to the Local Repository and select the AutomatedProcedures file.
15. Click **Tools** and open packages.config file.

16. Locate the dependencies identified in step 10 and remove them from the packages.config file.

The following screenshot presents an example of the packages.config file with all Single **Package Dependencies**:


```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Aras.IOM" version="14.0.15.38102" />
  <package id="Aras.Crt.InnovatorConfigs" version="14.0.15.38102" />
  <package id="Aras.DeltaExtraction.CommandLine" version="1.0.0.20" />
  <package id="Aras.Deployment.Tool" version="1.2.0.221" />
  <package id="Aras.Update.Cmd" version="1.22.1328" />
  <package id="Aras.ConsoleUpgrade" version="14.0.17.38577" />
  <package id="Aras.LanguageTool" version="14.0.17.38577" />
  <package id="Aras.Crt.Core" version="1.2.0.13732" />
  <package id="Aras.Nant.Shim" version="1.2.0.13732" />
</packages>
```

The following screenshot demonstrates the example of the packages.config file after the dependencies are removed:

```
<?xml version="1.0" encoding="utf-8"?>
<packages>
  <package id="Aras.IOM" version="14.0.15.38102" />
  <package id="Aras.Crt.InnovatorConfigs" version="14.0.15.38102" />
  <package id="Aras.Saas.DevOpsFramework.Msi" version="1.2.0.13823" />
</packages>
```

17. To restore the new **Packages** list with dependencies from NuGet, open Windows PowerShell as Administration and run the following command: Restore-ArasDevopsPackages -WorkRepository "<path_to_work_repository>"

Add the Work Repository path as a parameter. Ensure that the command is executed successfully and no errors appear in the console.

1. **Important**

   The **init.ps1**, **update.ps1** and **cleanup.ps1** scripts are executed (if present) only for **Packages** with names starting with Aras*.

18. Verify any changes to NuGet **Packages** by checking the output of **Restore-ArasDevopsPackages -WorkRepository** " <path_to_work_repository> command on Window PowerShell console.

This display indicates which **Packages** have been added, updated, or deleted in the Work Repository:

```
NuGet packages added to the work repository as a result of 'Restore-ArasDevopsPackages' call:

Name                            Version
----                            -------
Aras.Crt.AzurePipeline          1.2.0.13823
Aras.Crt.IntegrationTests       1.37.3838
Aras.Crt.SeleniumTests          1.37.3838
Aras.Saas.DevOpsFramework.Msi 1.2.0.13823

NuGet packages updated in the work repository as a result of 'Restore-ArasDevopsPackages' call:

Name           NewVersion  OldVersion
----           ----------  ----------
Aras.Crt.Core  1.2.0.13823 1.2.0.13732
Aras.Nant.Shim 1.2.0.13823 1.2.0.13732
```

19. Commit the changes using the required Version Control System.