ArcGIS PowerShell DSC Helper Scripts

A series of [helper scripts](https://github.com/BrendanEsri/PowerShell-DSC-Scripts/tree/main) have been developed to assist with both setting up and configuring the PowerShell DSC environment, as well as making the ArcGIS deployment more standardized and seamless. Below is a brief description of some of them:

1. dscScripts folder
   1. **clearDscConfig.ps1** – This script is good to run before and after running the Invoke-ArcGISConfiguration command, as it clears out any of the configuration settings currently in place within the PowerShell DSC module on the machine(s) specified. It’s good to run it twice, as well. The first time, it should clear out any settings. Running it again will allow you to confirm by looking at the console outputs that nothing is pending and that everything has in fact been reset/cleared.
   2. **localLcmSet.ps1** – This script sets the local DSC configuration by updating the LCM (local configuration manager) for each machine to ‘ApplyOnly’ configuration mode, and ‘StopConfiguration’ action after reboot. The ‘ApplyOnly’ configuration mode is not the default LCM setting but works best with PowerShell DSC deployments like we use for installing and upgrading Enterprise. What it does is specify that the LCM applies the configuration only, and then does not continue to monitor for changes and write logs about any drift. The ‘ApplyOnly’ setting essentially says “do what the script is telling you to do right now, and then nothing else until another script tells you to do something else, explicitly”. The ‘StopConfiguration’ setting for the action after reboot says that in the event the machine gets rebooted, the process will be stopped and upon reboot, the DSC process will not try to restart automatically.
2. transferScripts folder
   1. **transferCertificates.ps1** – this script will transfer the contents of a ‘certificates’ folder on an orchestration machine, or the local machine to remote servers to be used for the deployment. Make sure not to include the orchestrating machine in the list of $arcgisservers.
   2. **transferInstallers.ps1** – this script will transfer the contents of an ‘installers’ folder (as well as any sub-directories) from a local location to a list of machines specified for each ArcGIS component. Make sure not to include the orchestrating machine in the list of $arcgisservers.
   3. **transferLicenses.ps1** – this script will transfer the contents of a ‘licenses’ folder from a local location to a list of machines specified for each ArcGIS component. Make sure not to include the orchestrating machine in the list of $arcgisservers.
   4. **transferModule.ps1** – this script will transfer the local arcgis PowerShell module (without upgrading it) to a list of remote servers provided in the script. This is useful if utilizing an older version of the PowerShell DSC ArcGIS Module. Make sure not to include the orchestrating machine in the list of $arcgisservers.
3. **downloadUpgradeArcgisModule.ps1**
   1. This script will remove any ArcGIS Modules from the machines in the list and install the most up to date ArcGIS Module to each machine. If you do not want the most recent version, download the desired version from GitHub and utilize the script within the transferScripts folder that will transfer the module from the Orchestration machine to the list of remote servers in the $arcgisservers variable.
4. **generatePwFiles.ps1**
   1. This script allows for the generation of the password files that can be used in place of hard-coded password values in the JSON configuration file. Note: Once you specify the location of the password files in the script and run the script, the password files cannot be moved, or they will break. This is intentional. So, decide where you’d like for them to be located, make sure that path is specified in the script as well as the JSON configuration file, and then run the script. Note that the password files can only be used by the user who created them on the machine they were created on, within the folder they were created. This is for security purposed. The password files only need to be created on the machine that is being used to orchestrate the Invoke-ArcGISConfiguration command.
5. **InvokeScript.ps1**
   1. This script starts the ArcGIS deployment in PowerShell. It first changes the directory to a path where logs can be accessed easily (such as within the folder structure where the rest of the deployment files are located). Then, it runs the Invoke-ArcGISConfiguration command with the configuration file. You can change the -Mode switch to one of the other options [Install | InstallLicense | InstallLicenseConfigure | Uninstall | Upgrade | WebGISDRExport | WebGISDRImport] to adjust to what you’d like the Invoke command to do.
6. **transferPreRequisites.ps1**
   1. This script transfers the Web Adaptor prerequisites from a local machine to the web server that will be used for hosting the Web Adaptors.