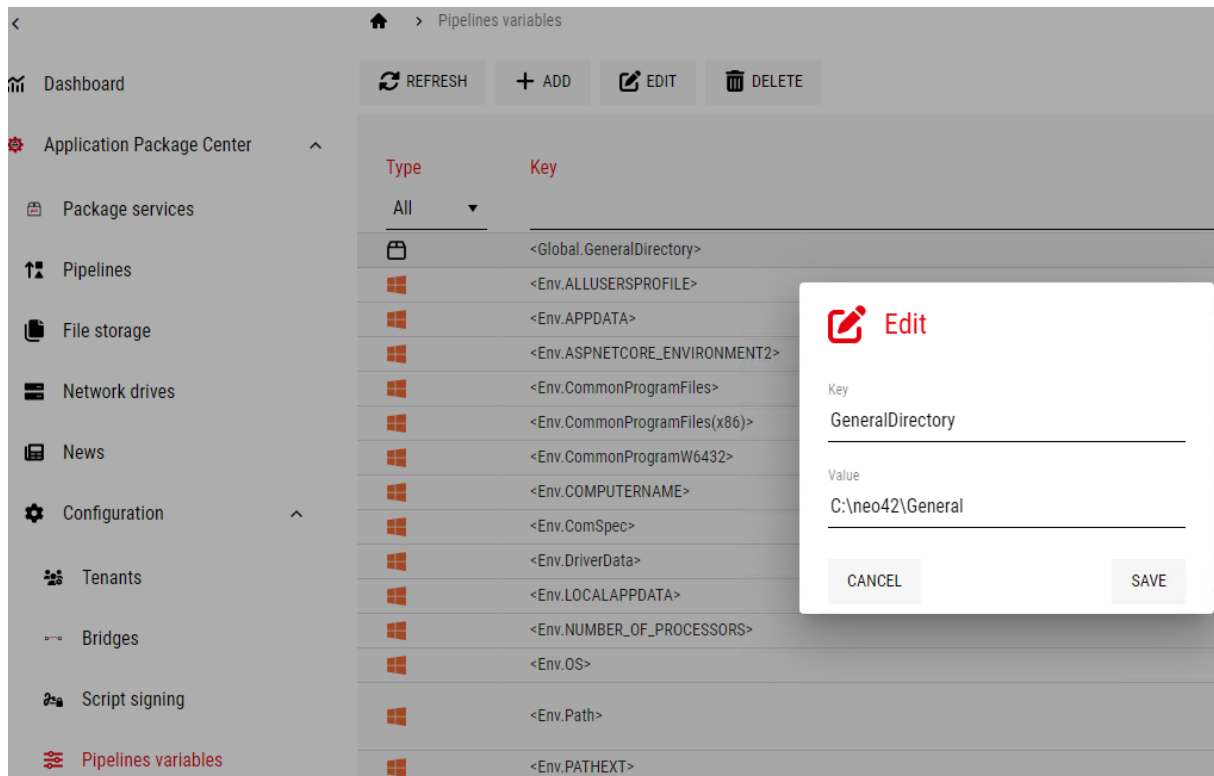


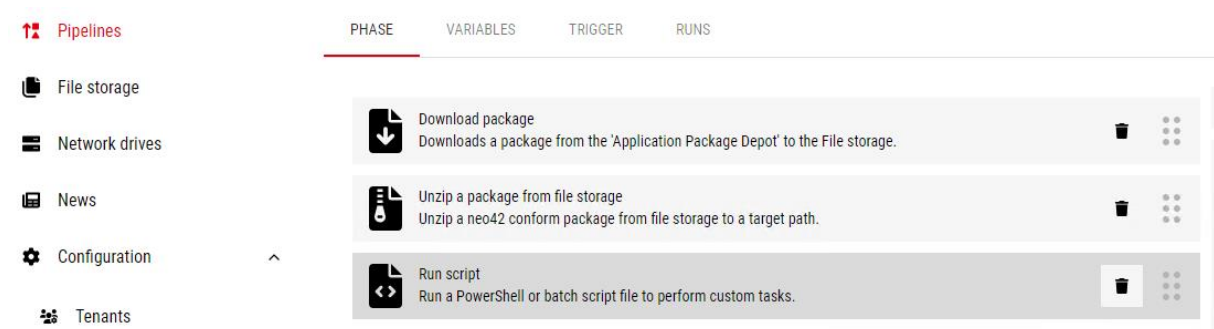
Um eine Json Datei immer wieder auf gleiche Weise bei verschiedenen Paketen ändern zu können, gibt es als Beispiel das Script Disable-BlockExecution.ps1

In unserem Beispiel setzen wir damit die Property BlockExecution in der neo42PackageConfig.json auf true.

1. Wenn noch nicht geschehen, sollte ein Verzeichnis für persistente Daten, die in Pipelines verwendet werden, angelegt werden. Beispielsweise „C:\neo42\General“ in dem das PowerShell Skript hinterlegt wird mit dem die ConfigMgr PowerShell CmdLets aufgerufen werden sollen. Diesen Pfad legen wir dann in einer globalen Variable des APC an.



2. Das Script mit dem Namen Disable-BlockExecution.ps1 in diesem Ordner „General“ hinterlegen
3. In der Pipeline den Task Skript „ausführen“ nach dem Extrahieren des Pakets einfügen













4. Unter „Skript Datei“ das Skript angeben

<Global.GeneralDirectory>\Disable-BlockExecution.ps1

5. Als Argumente den Pfad zur json Datei angeben

-Path "<Phase.PackagePath>\<Run.Version>\neo42PackageConfig.json"

6. Durch das hinzufügen weiterer Werte im Skript können weitere Properties nach dem Entpacken eines Pakets geändert werden. Bitte darauf achten falls diese für die Paketregistrierung usw. relevant sind sollten die Json Werte mit dem Task „Paket Konfiguration aktualisieren“ vom APC eingelesen werden

	Download package Downloads a package from the 'Application Package Depot' to the File storage.		
	Unzip a package from file storage Unzip a neo42 conform package from file storage to a target path.		
	Run script - Disable-BlockExecution Run a PowerShell or batch script file to perform custom tasks.		
	Package config refresh Updates the variable 'Run.PackageConfig'. The variable is required for all deploy tasks and must be updated after changes.	