# Data Storage

* Defaults (Stored in config/Defaults.json)
  + Global
    - RootPath
    - AppFolder
    - CertFolder
    - OscdimgPath
    - BaseImagePath
    - HDDPath
    - SnapShotPath
    - WorkingFolder
    - VMPath
    - VMTempFolder
    - TimeZone
  + Network
    - UseNAT
    - IPPrefix
    - SubnetMask
    - SubnetLength
    - Gateway
    - DHCPStart
    - DHCPEnd
  + BaseImage
    - LocalAdmin
    - LocalPassword
    - CustomUpdateSource
    - OSList (Array)
      * Name
      * Key
      * ImageIndex
      * AppsToInstall (Array)
  + Domains
    - Name
    - NBName
    - DomainAdmin
    - DomainPassword
  + Apps (Array)
    - Name
    - Installer
    - Transform
    - Patch
    - AdditionalFiles
    - InstallSwitches
    - PatchSwitches
    - URL
    - HardInstall
* Per VM (Stored in json under Notes)
  + Domain
  + LocalAdmin
  + LocalPassword
  + Roles
    - DC
    - CertServer
    - DNS
    - DHCP
    - Exchange
    - SQL
  + Domain (DC Only)
    - Name
    - NBName
    - DomainAdmin
    - DomainPassword

# Global Variables

* BeaterConfig – Global config data (Stored in Registry HKLM:/DaveWare/Beater)
  + LogFile – Location of log file
  + LogLevel – Detail level of log file
  + ConsoleLevel – Detail level of console output
  + RootPath – Root path for everything
  + AppFolder – Folder to store apps in
  + CertFolder – Folder with certificates to install
  + OscdimgPath – Path to oscdimg.exe
  + BaseImageFolder – Folder to store Base Images in
  + HDDFolder – Folder to store VM HDD files in
  + SnapShotFolder - Folder to store VM snapshots in
  + WorkingFolder – Temp folder
  + VMFolder – Folder to store VM config files in
  + VMTempFolder – Temp folder on the VMs
* Networks (Array) (Stored in json under Notes on vSwitch)
  + UseNAT – Y/N use NAT on switch
  + IPPrefix – first 3 octets of network
  + SubnetMask – 255.255.255.0 version of subnet mask
  + SubnetLength - /24 version of subnet mask
  + Gateway – IP of gateway on network
  + DHCPServer – Name of VM hosting DHCP
  + DHCPStart – Start of DHCP range
  + DHCPEnd – end of DHCP range
* BaseImages (Array) (Stored in JSON file in same folder as .vhdx)
  + Name – Name of Base image
  + OS – Type of
  + LocalAdmin
  + LocalPassword

# Functions

* Generic
  + Write-BTRLog – Write info to log and/or console
    - Parameters
      * **Entry** – text to log
      * Level
        + Error
        + FunctionProgress
        + Progress
        + Debug
  + Connect-BTRPSSession – Connect PS Session to VM
    - Parameters
      * **VMName**  - Name of VM to connect to
    - Returns - PSSession
  + Install-BTRSoftware – Install software, downloading if required
    - Parameters
      * **VMName** – Name of VM to install software on
      * Name – Descriptive name of product to install
      * **Installer** – Name of Installer. A \* means to find latest version that matches pattern in the folder
      * Transform – .mst file to apply. MSI installers only
      * Patch – Patch to aply after install
      * AdditionalFiles – Additional files to copy over to temp folder
      * InstallSwitches – Switches to apply to installer
      * PatchSwitches – Switches to apply to patch
      * URL – Url to download software from if it’s not in AppFolder
      * HardInstall – Switch use psexec to install remotely
    - Returns T/F
  + Wait-BTRVMOnline – Wait for VM to come online
    - Parameters
      * **VMName** – Name of VM
      * MaxWaitTime – Max time to wait in minutes. Default 5
      * WaitForLogin – Wait for logon to succeed
    - Returns T/F
  + Wait-BTRVMOffline – Wait for VM to go offline
    - Parameters
      * **VMName** – Name of VM
      * MaxWaitTime – Max time to wait in minutes. Default 10
    - Returns T/F
  + Wait-BTRVMReboot – Wait for VM to finish rebooting
    - Parameters
      * **VMName** – Name of VM
      * MaxWaitTime – Max time to wait in minutes. Default 10
      * WaitForLogin – Wait for logon to succeed
    - Returns T/F
  + Get-BtrNextIP – Get next available IP on switch
    - Parameters
      * **SwitchName** – Name of VSwitch
    - Returns – Next IP
* Base Image
  + NewBaseImage – Creates a new base image