Configuring the Admin Script Repository

Created: **08 Jan 2007** / Last Modified: **02 Jun 2024**

This document explains how to setup a repository of Windows automation scripts for Windows Server (versions 2000-2022+) to manage a Windows-based network of any size, across one or more locations. The various components of the archive are presented here, along with a brief description of what they accomplish and how to customize them for your particular needs.

For more information on the scripts, see the following website:
https://www.BrainWaveCC.com/brainwave-utilities

Overview

Details

The Purpose Behind the Admin Script Repository

The BrainWave Admin Script Repository is a collection of Windows shell scripts, other scripts (KiXtart, VBscript, Powershell, etc.), input and configuration text files, and executable files that simplify the configuration and monitoring of Windows-based network environments of all sizes. This collection of scripts has been assembled over many years, and refined for portability and utility. The scripts you choose for the repository will almost always be a subset of the scripts that you, as a systems administrator, maintain for your personal use.

Some of the scripts exist to manage the repository itself (and its distribution), enabling you to publish to remote systems only those scripts which you want to have running there, while keeping all other scripts confined to your primary workstation.

The distributed nature of these scripts, and the abstraction of the configuration files from the script body, make it very easy to use a single collection of scripts across different divisions, departments or even organizations (in the case of consultants or MSPs).

Key Components of the Admin Script Repository

There are two key components to the Admin Script Repository:

1. The script runtime location
2. The script repository staging location

Runtime Location for Scripts

C:\Scripts is the default runtime location for the scripts and utilities that are available on any particular system. The scripts will install to the **%SystemDrive%** location, which is almost always **C:** Almost all default settings can be changed by simply altering INPUT files.

Repository Staging Location for Scripts

On at least one system in your environment, probably on a key workstation or server, there needs to be a second set of folders that contain all the scripts and utilities that will be replicated to every other system. And this tree needs to be shared for remote access on the network. The folder location is up to you, as is the share name, but this information is needed by several scripts and should be stored in the **ScriptSource.TXT** file of the **\INPUT** folder (more on that later). I recommend the following naming convention for the script repository on your primary workstation: "**D:\Storage\Source\***YourCompany*". To get started, you can use the **MakeRepository.BAT** script to create a folder and share for you.

NOTE: The repository will almost always contain a subset of the scripts available on your local system, as there will be plenty of scripts and utilities that you are not going to want placed on every server – whether for licensing or other reasons. The primary purpose of the staging area is to keep track of what scripts are replicated, and keep them in a pristine state while you test or make changes to your local copy. The replication scripts will then update your staging area(s) with the latest versions of all input files, scripts and executable files that reside in your runtime location before pushing them out to remote systems.

Some Notes about Scheduled Jobs

Starting with Windows Vista and Windows Server 2008, scheduled jobs can be placed into a custom task scheduler folder, rather than the root folder which list many default jobs.

As of Dec 2012, all of the scheduled jobs created by this scripts repository are stored in the **\BrainWave** folder of the Task Manager to make them easier to identify and manage. This folder name can be changed, if desired, by altering the value for **@TaskFolder** found in the **SetDrive.BAT** script, or by adding the desired value to the **CustomVariables.TXT** file (recommended).

NOTE: Starting with Windows Vista (Windows 2008+), it is necessary to run the initial configuration scripts in an elevated Command Prompt in order to successfully make all the changes to the machine that the script repository requires. There is no problem with running the script more than once, so if you forget to elevate the prompt, it is easy enough to correct.

Description of the Script Runtime Folders

Below is a brief description of the folder structure for the Script Runtime area (**C:\Scripts**). The four key folders that are part of this tree are as follows:

C:\Scripts\BAT	Windows Batch/Shell Scripts for Administration
C:\Scripts\BAT	KiXtart Scripts are also stored here
C:\Scripts\BAT\Input	Input/Source Files for the Scripts
C:\Scripts\Perl	Perl-based Scripts for Administration
C:\Scripts\PowerShell	Windows PowerShell Scripts for Administration
C:\Scripts\VBS	Visual-Basic for Applications Scripts for Administration
C:\Scripts\Utils	Administrative Executable Files (3rd party utilities)
C:\Scripts\Utils\Misc	Optional or infrequently used Executable Files
C:\Scripts\Utils\x64	64-bit Executable Files (3rd party utilities)
C:\Scripts\Zips	Monthly zip archives of scripts and utilities

Description of the Script Staging Folders

Below is a brief outline of the folder structure for the Script Staging area. This is the source tree for replication of scripts and utilities to remote systems.

```
Directory of D:\Storage\Source\MyCompany

05/18/2024  10:06 PM    <DIR>          .
12/22/2023  03:35 PM    <DIR>          ..
05/05/2016  05:23 PM    <DIR>          Scripts
08/27/2023  06:00 PM           157,216 Now.exe
03/25/2024  10:00 PM           624,672 DateInfo.exe
05/09/2024  11:00 PM           195,616 FindFiles.EXE
05/20/2024  05:00 PM             3,200 Install.BAT

D:\STORAGE\SOURCE\MyCompany
├───Scripts
│   ├───BAT
│   │   ├───Input
│   │   ├───LogonScripts
│   │   │   ├───Scripts
│   │   │   │   ├───Bat
│   │   │   │   │   └───Input
│   │   │   │   └───Utils
│   │   │   └───System32
│   │   │       └───x64
│   │   └───Services
│   │       └───EvtSys
│   │           ├───x86
│   │           └───x64
│   └───Utils
│       ├───Misc
│       └───x64
└───Zips
    └───Archives
```

Staging and Replicating the Scripts

Before you begin with the staging area, copy all the scripts into the **C:\Scripts** tree as indicated in the previous section, and establish your local runtime area. Now, take some subset of those scripts and utilities, and copy them to the location you have designated for the staging area. For the purposes of this documentation, we will use the following staging location: **D:\Storage\Source\***CompanyName*

You will need to share this folder, and ensure that all other systems can access this share by setting both file and share permissions to **READ** access for the account(s) you will use when replicating the scripts. And you need to have **R/W** access to all the remote repositories from the same account(s). The default share name is **Scripts_Source$**, but this can be customized and stored in the **INPUT\ScriptSource.TXT** file for access by the replication scripts:

▪ **ReSyncScripts.BAT**	Replicate scripts to all systems found in **FolderDefaults.TXT**
▪ **SyncScripts.BAT**	Refresh scripts in local archive from central repository on same box
▪ **UpdateScripts.BAT**	Refresh scripts on an individual server from the source repository

Edit the **ScriptSource.TXT** file and enter the UNC path of your Repository Staging area(s) and their corresponding local paths. The first two scripts rely on the local path, while the latter uses the UNC path. For convenience, both entries are stored in the same file. The **ReSyncScripts.BAT** script also relies on the **INPUT\FolderDefaults.TXT** file to determine which drive to use on the destination systems during the replication.

NOTE: Be sure to maintain the existing format of the input files, as this will allow the data locations to be properly parsed. For instance, the **ScriptSource.TXT** file requires that the share path used be the parent folder of the **BAT** and **UTIL** directories.

A basic **ScriptSource.TXT** file, based on our example above would be:

```
UNC PATH of Valid Script Repository     Physical Drive Path of Repository
-------------------------------------------------------------------------
"\\DESKTOP1\Script_Source$\Scripts",    "C:\Storage\Source\MyOrg\Scripts"
"\\SERVER03\Script_Source$\Scripts",    "D:\Storage\Source\MyOrg\Scripts"
```

Don't remove the first two lines, because the scripts all skip the first two lines before parsing this INPUT file. Once this is done, you can replicate the scripts by typing the full path of the script plus "." as a parameter: **C:\Scripts\Bat\ReSyncScripts.BAT .**

NOTE: This is the only script in the repository which requires the full path, case sensitivity in the filename, **and** mandatory parameter in order to successfully execute it. This is done merely as a safeguard. Accidental automation is dangerous!

After the repository has been created, you can use it to push or pull the scripts. To pull the scripts to a machine, whether or not it listed in the **FolderDefaults.TXT** file, you can run the following command from that machine: **\\SourceSystem\Scripts_Source$\Install.BAT**, where **SourceSystem** is a valid script repository source that you created earlier, assuming you used the default share name. This will download a copy of the scripts to the current system and then set the configurations for folder, shares, scheduled jobs and system path based on the contents of the **FolderDefaults.TXT** file. If the current machine has no entry in this input file, then all folders and shares are created on **%SystemDrive%**.

This process is explained in more detail in the next section.

Pushing/Pulling Updated Scripts

Once you have deployed the scripts to all the target desktops and servers, you will want to keep them up to date. There are two ways to refresh them – centrally or individually. The central updates are pushed to the relevant systems from an authorized repository machine, while the individual update is pulled down from the repository to the local machine.

PUSH: The scripts for all systems listed in the **FolderDefaults.TXT** file are updated centrally using the **ReSyncScripts.BAT** command. This script must be run from one of the machines listed in the **ScriptSource.TXT** file – the authorized script repository list.

PULL: The other option is to go to the system that you want to update, and update the scripts there using the **GetScripts.BAT** command. This is especially useful for any machine which is not included in the replication list (**FolderDefaults.TXT**).

You can provide a list of script repository sources for the **GetScripts.BAT** command in any of the following three ways:

- Editing the **GetScripts.BAT** script directly (not recommended)
- Adding a list of UNC paths to the **ScriptSource.TXT** file in the Input folder
- Adding a valid UNC at the command line e.g. **GetScripts** **\\someserver\someshare$**

It is no longer recommended that you always have at least one good path in the batch file, since additional paths can be added to the **ScriptSource.TXT** file or at a command prompt.

Customizing the Scripts for Your Environment

The script repository was designed with a certain level of abstraction. It's not quite perfect – not yet, anyway – but it has matured to the point where complete customization can be attained by editing only a handful of files within the INPUT folder.

Most of the scripts – including all of the major or popular ones – contain a fair amount of documentation to assist with modification. Always review the internal script documentation to determine exactly what supporting files it needs in order to execute properly.

While a few scripts have their own INPUT file, there is now a common INPUT file available to all scripts called **CustomVariables.TXT**. This central configuration file can supply any variable that is needed by a script at run-time. To use it, just add/change entries in the file with the name of the script without extension (e.g. **GetInfo** for **GetInfo.BAT**), the variable name, and variable value, delimited by a semicolon. Variables which should be available for all (or many) scripts should be listed with an ampersand (**&**) as the script name.

```
Name of Script;      Variable Name;     Default Value of Variable
---------------------------------------------------------------------------
&;                   @MAILSERVER;       mail.mydomain.com
&;                   @MAILPORT;         25
GetInfo;             @SUBJECT;          Malware Alert for: %@UNKNOWN%
GetInfo;             @MY_PATCHMGMT;     Automatic Updates
GetInfo;             @MY_ANTIVIRUS;     CylancePROTECT
---------------------------------------------------------------------------
```

(Since at least 2017, the major scripts leverage central config files for their custom settings.)

The folder that contains most of the files you need to customize is the text **INPUT** folder (**C:\Scripts\Bat\Input**). This folder contains the various job configuration files and lists of systems that are needed by many of the scripts to perform their duties. It also used the host the **@Variables.BAT** file, which was used to set global script settings. This script, however, has been superseded by **SetDrive.BAT** in the main batch file folder.

- **ScriptSource.TXT** - This file is used by the following scripts: **SyncScripts, ReSyncScripts,** and **UpdateScripts.** This is where you will put the UNC path of your Repository Staging area(s), and their corresponding local paths. The first two scripts rely on the local path, while the latter uses the UNC path, and because of the relationship of the locations, it made more sense to store the information in a single location.

- **FolderDefaults.TXT** - Contains the systems you want to have the scripts copied to, along with the drive that will serve as the root for the **\Storage\Logs** and **\Storage\Backups** folders. This file is used primarily by the **ReSyncScripts.BAT** file to replicate the scripts across your environment. It is also used by the **SetDrive.BAT** script to determine where to put logs, backups and other data files generated by the scripts in this archive. If no **FolderDefault.TXT** file exists, each system will make "**%SystemDrive%\Storage**" the Storage folder location. This is usually **C:**

- **CustomVariables.*computername*.TXT** - Holds all custom script variables, including the email settings for scripts that use customized SMTP configuration such as recipients, mail servers, or any other variables. The variables are associated with the script name, so if you change the name of your script, be sure to make the corresponding change to this file's contents.

- **CustomVariables.MINI.*computername*.TXT** - Used to make allow minor configuration deviations for a specific system. It is processed after the **CustomVariables.TXT** file has been processed.

- **ArchiveInfo.*computername*.TXT** - Contains the source/destination information needed to manage zip archival and replication of files with a retention schedule. The variables are associated with specific archival or replication jobs and are called by various scripts such as **CheckArchive.BAT**, **CreateArchive.BAT**, **MoveArchives.BAT** and **RetainBackups.BAT**, among others.

- **ReplicationInfo.*computername*.TXT** – Contains the source/destination information needed to replication one file tree to another location. The variables are associated with specific replication jobs called by various scripts such as **Replicate-Files.BAT**.

- **ServerList.*computername*.TXT** - Most scripts which require a list of systems to act upon (e.g. **OpsLogs.BAT**, **Do.BAT**, and many more) will default to using this file, unless you provide another.

▪ **AC-DCs.TXT**	List of domain controllers in the environment
▪ **SmallList.TXT**	Small list of test systems used by various scripts during debug mode
▪ **SecurityList.TXT**	List of systems from which to gather EventLog and other Security info
▪ **UptimeReport.TXT**	List of systems to check for Uptime Statistics (subset of **OpsLogs.BAT**)
▪ **IPConfig.TXT**	List of systems and their IP, DNS, WINS and TimeZone info
▪ **MyServers.TXT**	List of systems that can be accessed by the **ManageUtil.BAT** script
▪ **OpenCMD.TXT**	List of accounts for executing RunAs commands via **OpenCMD.BAT**
▪ **[SysInternals].TXT**	List of registry keys for setting the SysInternals EULA approvals
▪ **[BrainWaveUtils].TXT**	List of current BrainWave console utilities that power the repository
▪ **[KnownServices].TXT**	List of accepted services that will be checked by **GetInfo.BAT**
▪ **[LongProcesses].TXT**	List of long-running processes; checked by **KillLongProcess.BAT**
▪ **[CopyExemption].TXT**	List of files to skip when using XCOPY to replicate files and folders

If you want any particular system to use a custom version of any of the files above, such as **ServerList.TXT**, then make a file called **ServerList.***computername***.TXT**. The system that matches the name in this file will be the only system that uses that file. This allows you to store all customized files centrally, replicate them globally, but still limit their execution to a specific system. (Example: only **MYSERVER01** will use **ServerList.MYSERVER01.TXT**)

Once you have edited the **FolderDefaults.TXT** and **ScriptSource.TXT** files, you will be ready to start replicating scripts around. Be sure to maintain the existing format of the files, as the data locations are important for parsing purposes.

Managing Those Lists of Systems

The default repository comes with a set of generic files which you have to populate with your own information before the scripts can get really start working for you. The biggest burden would be configuring all of the different system lists as found in the INPUT folder, such as: **AD-DCs.TXT**, **ServerList.TXT**, **FolderDefaults.TXT**, etc.

Trust me – I feel your pain. As I have relied heavily upon these scripts to manage customer networks, I have done a tremendous share of list maintenance, and finally decided to make the whole process easier. Rather than consolidate the text files and update the scripts to match (lots more work, ultimately), I settled on a different approach using Excel and a shell script.

Excel is used to generate a CSV that serves as the input for the new script which will neatly generate all the text files for you – excellent for bulk updates. You can manage the list directly as a CSV, but the Excel file (**SystemList.xlsm**) contains conditional formatting and a macro that are intended to simplify the process. The shell script (**MakeSystemLists.BAT**) is then used to generate all the necessary scripts in the right format from the CSV file. A description of the spreadsheet/csv format follows:

▪ **SERVER_NAME**	Hostname
▪ **IP_ADDRESS**	Primary IP Address of the system (optional)
▪ **ENV**	The environment the system resides in (e.g. PRODUCTION, DEV, QA)
▪ **LOGS**	Drive letter to store the **Storage$** share
▪ **ROOT**	Root share of **%SystemRoot%**
▪ **BACKUPS**	Which drive to place the **Backups$** share, if different from **LOGS**
▪ **CLASS**	Description of device type (Server, Mobile, Workstation, Storage, etc)
▪ **TYPE**	Description of hardware type (Physical or Virtual)
▪ **OS_VERSION**	Operating System Version/Edition (i.e. Windows 2016 x64)
▪ **DEVICE_PURPOSE**	Description of the device role or function

▪ **FOLDER_DEF**	List of systems receiving the scripts **(FolderDefaults.TXT)**
▪ **DC_LIST**	List of domain controllers **(AD-DCs.TXT)**
▪ **MGMT_LIST**	List of systems accessed by RDP at once **(MyServers.TXT)**
▪ **SERVER_LIST**	List of systems for many scripts **(ServerList.TXT)**
▪ **UPTIME_LIST**	List of systems for many scripts **(UptimeList.TXT)**
▪ **SEC_LIST**	List of systems for many scripts **(SecurityList.TXT)**
▪ **SMALL_LIST**	Small list of systems for testing **(SmallList.TXT)**
▪ **EXCH_LIST**	List of Exchange servers **(ExchangeList.TXT)**
▪ **EXCH_TEST**	List of test Exchange servers **(ExchangeTest.TXT)**
▪ **LOW_DISK**	List of systems for storage monitoring **(LowDisk.TXT)**
▪ **SUS_SERVERS**	List of servers for WSUS configuration **(SUS_Servers.TXT)**
▪ **SUS_CLIENTS**	List of desktop clients for WSUS configuration **(SUS_Clients.TXT)**
▪ **IN_DEFAULT**	List of systems in the default version of the above scripts
▪ **CUSTOM_LISTS**	Names of one or more customized lists from above

For the list columns, a "**Y**" entry indicates that the system will be written to the list. An entry of "**C**" will also write a given system to the list, but prefixed by a semi-colon, which comments the whole line. This is useful if you want to keep a system in a list and enable it manually at different times. Any other character ("**-**" preferred) will keep the system out of the associated file. An entry of "**C**" in the *IN_DEFAULT* column will supersede all "**Y**" entries for that system.

All of the columns from *FOLDER_DEF* to *IN_DEFAULT* <u>must</u> have some entry in them, or the **MakeSystemLists.BAT** script will generate incorrect output. To aid in this endeavor, the spreadsheet will highlight any cell that is missing data when it should not be missing any.

The documentation for this section has been enhanced and is being captured in a spreadsheet. An updated edition of this document will provide a link to the spreadsheet, but for now, there should be enough information to make real deployment progress.

Key Scripts in the Repository

▪ **SetDrive.BAT**	Determine which drive to dump log files and store other data
▪ **JobHistoryUpdate.BAT**	Obtain Most Current Job History Update for Scheduled Jobs
▪ **Do.BAT**	Run commands against a group of remote systems at one time
▪ **DNSCheck.BAT**	Determine if AD DNS servers are in sync with working replication
▪ **ConfigureNetwork.BAT**	Set IP, DNS, WINS and TimeZone info for each server
▪ **GetComputerIPs.BAT**	Obtain List of Systems from Active Directory, with IP Addresses
▪ **FindCommand.BAT**	Search for script or executable in the local script repository
▪ **FindInBatch.BAT**	Search for content within the script files on the local repository
▪ **JobHistoryUpdate.BAT**	Obtain a summary of most recent history for major scripts
▪ **MakeRepository.BAT**	Create a repository share on the local machine for distribution
▪ **SpaceInfo.BAT**	Quickly display the **StorageInfo.BAT** logs for a local or remote server
▪ **TaskConfig.BAT**	View/Update the credentials for scheduled jobs

Key Scripts in the Repository (Scheduled Jobs)

▪ **BackupDHCP.BAT**	Scheduled weekly backups of the DHCP config on DCs @ **4:30 am**
▪ **BackupDNS.BAT**	Scheduled weekly backups of the DNS config on DCs @ **4:30 am**
▪ **BackupGPO.BAT**	Scheduled weekly backups of the GPOs on DCs @ **4:15 am**
▪ **CheckIntegrity.BAT**	Perform file integrity verification of critical system files daily @ **0:30 am**
▪ **CreateArchive.BAT**	Scheduled daily/weekly/monthly backups of various logs and data files
▪ **Debug.BAT**	Obtain System Diagnostics Info of a local system
▪ **GetInfo.BAT**	Obtain Remote System Forensics for Incident Response
▪ **IPDebug.BAT**	Obtain IP Diagnostics Info of a local system
▪ **KillLongProcess.BAT**	Scheduled Job to kill any process that exceeds predetermined duration
▪ **MoveArchives.BAT**	Move zip archives or other files to the network on a retention schedule
▪ **OpsLogs.BAT**	Obtain Operational System Stats from Key Systems @ **5:00 am**
▪ **SaveLogs.BAT**	Save Event Logs every night @ **1:30 am**
▪ **Replicate-Files.BAT**	Scheduled daily backups with mirroring or non-mirroring configuration
▪ **RetainBackups.BAT**	Scheduled daily/weekly/monthly backups with retention/purging options
▪ **StorageInfo.BAT**	Capture Local Storage Consumption every night @ **2:30 am**
▪ **ZipLogs.BAT**	Schedule Monthly Compression/Archive of Various System Log Files
▪ **ZipFiles.BAT**	Schedule Monthly Compression/Archive of Other Data Files

Please see the individual scripts for more documentation.

NOTE: The **OpsLogs** and **BackupDNS** scripts are among the very few scripts in the archive that <u>must</u> be scheduled to run in a specific privileged user context, rather than the default **SYSTEM** account that they are created in. Use **TaskConfig.BAT** to automate**.**

Default Folders Used in the Script Repository

Below is an extended description of the folder structure for the Script Runtime location. The four key folders that are part of this tree are as follows:

C:\Scripts\BAT	Batch/Shell Scripts for Administration
C:\Scripts\BAT\Input	Input/Source Files for the Scripts
C:\Scripts\Utils	Administrative Executable (3rd party utilities)
C:\Scripts\Zips	Monthly zip archives of scripts and utilities

Another critical folder is the **\Storage** folder. There is always a **\Storage** folder on the system drive, but on a per server basis, you can choose to create a **\Storage** folder on any drive which has sufficient space. This is where the various backup scripts and logging scripts will place their data. For the purpose of this document, we will assume a configuration of **D:\Storage**, but it can be any local drive of your choosing and is controlled by configuring the **INPUT\FolderDefaults.TXT** file.

C:\Storage\Drivers	Feel free to place system drivers here
C:\Storage\Docs	Feel free to place system docs here
C:\Storage\i386	Recommended location for the i386 source files (slipstreamed)
D:\Storage\Installs	Feel free to place installation files for the system here
D:\Storage\Downloads	Feel free to place downloaded files here

D:\Storage\Logs	Log files for Syslog, EventLogs and Scripts
D:\Storage\Logs\Zips	Log archives; Automatically generated monthly
D:\Storage\Backups	Daily backups of SystemState (includes Mailboxes on Exchange)
D:\Storage\Backups\DHCP	Weekly backups of local DHCP Scopes on DHCP servers
D:\Storage\Backups\DNS	Weekly backups DNS Zones on DCs (or other DNS servers)
D:\Storage\Backups\GPO	Weekly backups of Group Policy Objects via the GPMC

Default Shares Used in the Script Repository

Here are the default shares that are created by the **$NewSetup.BAT** script.

Drivers$	C:\Storage\Drivers	System Drivers
SysDocs$	C:\Storage\Docs	Admin Docs
Scripts$	C:\Scripts	Admin Scripts

Storage$	D:\Storage	Root of Storage Folders
Installs$	D:\Storage\Installs	Software Install Point
Logs$	D:\Storage\Logs	System & Application Logs
Backups$	D:\Storage\Backups	System Backups
Downloads$	D:\Storage\Downloads	Download Archives

NOTE: On any given system, the **Storage$** share and its associated folders may (should) be placed on the drive or partition with the most free space. This setting is controlled by the **C:\Scripts\Input\FolderDefaults.TXT** input file and can be different on each system. Generally speaking, the goal is for this share/folder to be on drive D: or any other drive with a lot of free storage **(min 50GB free)**.

Local Installation of the Scripting Environment

Once you have replicated the scripts to all the target systems, you will need to run the following command on each of those systems to create the appropriate folders, shares and scheduled jobs, along with registry changes.

The command to run is: **C:\Scripts\Bat\$NewSetup.BAT**

This script can be run successfully via a remote shell (using **psExec** from SysInternals) for remote systems, so you won't have to visit each of them. For best results, please run this script with the full path, since your system won't know where to find these files until the script has completed. If you ever change the drive letter used for the **\Storage** folder on a specific machine, be sure to update the input files, then re-run **$NewSetup.BAT** to modify the shares and scheduled jobs on that machine.

$NewSetup.BAT should be followed by **TaskConfig.BAT** to ensure that all scheduled jobs using custom credentials are configured to run in the background, and not just interactively.

NOTE: It is essential to run the initial configuration scripts (and certain other scripts) in an elevated Command Prompt in order to successfully make all the necessary system changes for the script repository requires. There is no problem with running the script more than once, so if you forget to elevate the prompt, it is easy enough to correct.

And there you have it – everything you need to start customizing this script repository for your own environment... Be sure to check out the various scripts that are sitting in the BAT folder. You might find something useful, such as:

aCopy.BAT	**BulkRename.BAT**	**CheckIntegrity.BAT**	**Connect.BAT**
CreateArchive.BAT	**Do.BAT**	**DoPS.BAT**	**FindCommand.BAT**
OpenCMD.BAT	**GetServerIPs.BAT**	**ManageUtil.BAT**	**Replicate-Files.BAT**
TaskConfig.BAT	**JobHistoryUpdate.BAT**	**ZipFiles.BAT**	**ZipLogs.BAT**

This repository also contains some nifty executables (x86 & x64), written in FreePascal:

CCalc.EXE	**ChangeCase.EXE**	**CheckParams.EXE**	**DateInfo.EXE**
FileHash.EXE	**MakeString.EXE**	**Now.EXE**	**PrimePlus.EXE**
PrintFileInfo.EXE	**Readable.EXE**	**ReadConfig.EXE**	**SubString.EXE**

But wait! There's much more… For example, there are quite a number of scripts in this archive which can facilitate replicating data between multiple storage system, or creating backups with a retention schedule. They vary in features and flexibility, from the very flexible **RetainBackups.BAT**, to the very basic **Replicate-Files.BAT**. Take some time to explore the archive and see what value it can add to your systems administration toolbox.

Most of the scripts support the **/H** or **/?** parameters for obtaining syntax assistance.

NOTE: While the settings of many scripts can be made directly in the scripts themselves, it is highly recommended that you make the changes in the appropriate configuration file instead. For running in diagnostics mode, many scripts provide a parameter to turn this on or off. Additionally, many scripts look for the **DEBUG** environment variable set to **TRUE** or **LOG** or **COPY** to obtain diagnostics information at run time.

Input files in brackets (e.g. **[ScriptInput].TXT**) are global files which are not as likely to require customization at each location.