WINDOWS FILESYSTEM CLEANUP SCRIPT

Filesystem cleanup procedure on VDL NedCar IT Windows operating systems.

VNB-Cleanup files

Contents

Windows Server – File system cleanup procedure.

Introduction.

The file system of a Windows computer system needs a regular cleanup of temporary/unneeded files created by the Windows operating system or any applications running on it. If these files or folders are not removed by their applications/services the filesystem will reach critical capacity levels and may cause performance issues.

A set of Powershell modules and script have been created in order to maintain the filesystem by removing files and folders specified by a set of rules. The script is known as the 'VNB-Cleanup files' script.

Installation requirements.

The minimum requirements for this Powershell script (and required Powershell modules) to run are:
- Windows Management Framework 3.
- Powershell 3.0.
- NTFS file system supporting filesystem object attribute 'LastWriteTime'.
- WinRAR 5 for file archiving purposes.
- The script must run in UAC Administrator mode.

This script or its modules will not run on Windows 2000 or lower. Microsoft Windows Server 2003 and Windows Server 2008 is only supported when WMF 3 is installed.

Installation.

Installation of the Cleanup script and modules is part of the VDL NedCar Windows server standard configuration. The Windows Active Directory domain group policy '`GPO NCSTD FilesEnv`' makes sure the latest version of the Powershell modules and the main cleanup script are installed on any system where the policy is active. The script is installed at '[SystemDrive]\Scripts\Acties' and named 'cleanup.ps1'. Together with the script all other necessary files are located in the same folder where the script resides.

The domain GPO '`GPO NCSTD FilesEnv`' also installs the latest version of the server configuration scripts. The server configuration scripts are responsible for configuring the Windows Task Scheduler with the required task 'VNB-Cleanup files' that executes the cleanup script. The installation is therefore completely automated and does not need any user action.

Determining the age of a file or folder.

Every NTFS file or folder has three different date and time attributes.

Creation date /time. This is set at the time of creation of a file or folder and it will not change after that.
Modified date/time. This is set to the time when a file is modified. Be aware that a folder modified date may not change even though the contents of the folder is changed. The folder modified date is changed when folder attributes or permissions are changed.
Accessed date/time. This attribute will change whenever a file or folder is accessed.

The cleanup script uses the Modified date/time attribute as its selection criterion.

Triggering the cleanup script with the Windows task scheduler.

The cleanup script is triggered to run with scheduled task 'VNB-Cleanup files' that is set to run once a day at 5:00 AM. This task should not be changed in any way as its name and configuration is controlled by the server configuration scripts. A future update of these scripts will overwrite the default task causing the loss of any customization.

The default scheduled task will run the cleanup wrapper script '[SystemDrive]\Scripts\Acties\cleanup.cmd' at 5:00 AM with all the rules defined in '[SystemDrive]\Scripts\Acties\cleanup.xml' and all custom cleanup configuration files defined in '[SystemDrive]\Scripts\Cleanup\CleanupSet.xml'.
Extending the cleanup with additional/custom configuration files is explained in the chapter 'Configuring the cleanup script'.

Customizing the triggering of the script.

If for any reason the cleanup task must run at different times, intervals of with a custom configuration at a moment not suited for inclusion of the regular run at 5:00, a new scheduled task must always be created. Do not change the default task 'VBN-Cleanup files' in any way.

Triggering the default cleanup configuration for additional times/intervals.

If for any reason it is not sufficient to run the script once a day at 5:00 AM one should create a new scheduled task, configure the appropriate run times and call the wrapper script '[SystemDrive]\Scripts\Acties\cleanup.cmd'.

Triggering a custom cleanup configuration within the default trigger at 5:00 AM.

The default cleanup can easily be extended by adding custom cleanup configuration files. This does not require additional scheduled tasks but only the inclusion of the custom configuration in the cleanup configuration set. The names of these custom configuration files are declared in the cleanup set '[SystemDrive]\Scripts\Cleanup\CleanupSet.xml'.

Triggering a custom cleanup configuration outside the default trigger at 5:00 AM.

In this case it is required to create a new scheduled task, configure the appropriate trigger and call the cleanup script '[SystemDrive]\Scripts\Acties\cleanup.ps1' with the addition of a command line parameter called 'CleanupXMLInputFile'.

Task example:
The following script and parameter information must be used when triggering the script with a custom scheduled task.

```
Action program/script: powershell
Arguments:  -file 'C:\Scripts\Acties\cleanup.ps1' -CleanupXMLInputFile '<custom configuration file path>'
```

Configuring the cleanup script.

System wide cleanup configuration.

The script uses a set of XML configuration files that control which files or folders are removed or what archives are created. The main XML file '[SystemDrive]\Scripts\Acties\cleanup.xml' is used for Windows system temporary file cleanup. This configuration file should not be changed. Changing this file is pointless as it is refreshed every 3-4 hours within the Active Directory GPO refresh routine. Due to the fact that it is a domain wide updated file, the configuration rules should only apply to files or folders that are usually found on all domain systems.

Cleanup configuration XML structure.

The cleanup script uses XML files for input and controlling the behavior of the script. The following is an example of such a cleanup configuration XML file.

```
<?xml version="1.0"?>
<?xml-stylesheet type='text/xsl' href='style.xsl'?>
<cleanup description="File system cleanup"
       version="08-12-2005"
       type="PROD"
       logfile="%SYSTEMDRIVE%\Logboek\Cleanup\Cleanup-system.log"
       append="false"
>
       <fsobject type="FOLDERS" path="%SYSTEMROOT%\TMP"  age="14" include="*" />
       <fsobject type="FILES"   path="%SYSTEMROOT%\TMP"  age="1" />
       <fsobject type="FOLDERS" path="%SYSTEMROOT%\TEMP" age="14" include="*" />
       <fsobject type="FILES"   path="%SYSTEMROOT%\TEMP" age="1" />
       <fsobject type="ALLCONTENT" path="%SYSTEMDRIVE%\TMP"  age="90" include="*" />
       <fsobject type="ALLCONTENT" path="%SYSTEMDRIVE%\TEMP" age="90" include="*" />

       <!-- Remove old logs without tampering logs in Config folder -->
       <fsobject type="FILES_NORECURSE" path="%SYSTEMDRIVE%\Logboek" age="90"
keep="100" />
       <fsobject type="FILE" path="%SYSTEMDRIVE%\Logboek\archive.zip" age="30" />

       <!-- Compress logs folder -->
       <fsobject type="COMPRESSNTFSFOLDER" path="%SYSTEMDRIVE%\Logboek" age="0" />


       <!-- IIS service log files cleanup -->
       <fsobject type="FILES" path="%SYSTEMDRIVE%\inetpub\logs\LogFiles" age="90"
keep="100" />
</cleanup>
```

Within the tag **<cleanup>** the following attributes are defined:

Attribute	Required	
Description	No	Any value allowed.
Version	No	Any value allowed.
Type	Yes	Any other value than PROD, PRODUCTIE, PRODUKTIE, PRODUCTION, PRODUKTION will cause the script to execute in debug mode. During debug mode the script will only log its action but will not execute any file/folder removal or archive actions.
Logfile	No	Must be a full pathname to the log that will be written to. If not defined the script will log to [SystemDrive]\Logboek\Cleanup\Cleanup-undefined-logname.log with Append mode set to False.
Append	No	If set to TRUE, will append text to an already existing log. If not set or all other values than TRUE will overwrite the previous log.

Within the tag **<cleanup>** the additional tag **<fsobject>** can be defined. This object has a number of attributes:

Attribute	Required	Default value	
Type	Yes		Cleanup type. See explanation below.
Path	Yes		Full pathname. The path may contain system environment variables. These will be expanded at runtime.
Age	Yes		Age in days. The value must be numerical, and represents the age in a number of days.
Keep	No	0	Number of files or folders to keep even if they are older than <Age> days.
Include	No	*	Selection of files or folders to include in the cleanup by using a filesystem wildcard definition. The default value will include all files/folders.
Exclude	No		Selection of files or folders to exclude in the cleanup by using a filesystem wildcard definition. The default value is to not exclude any file/folder.
Description	No		You can describe the cleanup.
Comment	No		You can add comment to this cleanup.

The script will execute the cleanup per FSObject rule. Each rule has a Type definition. The following are valid values for Type.

Type value	
FILE	Deletes a single file
FILES or FILES_RECURSE	Deletes files in a folder and all its subfolders with recursion.
FILES_NORECURSE	Deletes files in a folder but not its subfolders. (no recursion)
Type value	
FOLDER	Deletes a single folder and all its contents.
FOLDERS or SUBFOLDERS	Deletes only subfolders and all their content.
ALL or AllCONTENT	Deletes all files in a folder and deletes all subfolders in a folder. (Basically a combination of FILES and FOLDERS)
Type value	
ZIPFILES	Creates a zip archive called archive.zip in a folder. The procedure moves all the files into the archive without subfolder recursion.
ZIPFILESFOLDERS	Creates a zip archive called archive.zip in a folder. The procedure moves all the files and folders into the archive with subfolder recursion.
ZIPSUBFOLDERSONLY	Creates a zip archive called archive.zip per subfolder in a folder. The procedure moves all the files and folders per subfolder into the archive.
ZIPHISTORY or ZIPSINGLEFILES	Creates a zip archive per subfolder in a subfolder called 'history' with subfolder recursion.
Type value	
COMPRESSNTFSFOLDER	Compress a folder and its contents (with recursion) with NTFS filesystem compression. Compression can only occur if NTFS file compression is supported and switched on. (See: fsutil behavior query disablecompression)
DECOMPRESSNTFSFOLDER	Decompress a folder and its contents (with recursion) with NTFS filesystem compression.

Extending the default cleanup configuration.

Besides the main XML file '[SystemDrive]\Scripts\Acties\cleanup.xml' which is used for Windows system temporary file cleanup, it is possible to extend the default cleanup run with additional (custom) configuration files.
By declaring a set of configuration files in '[SystemDrive]\Scripts\Cleanup\CleanupSet.xml' the cleanup can be customized. These additional cleanup configurations are executed one-by-one after the default cleanup has completed.

```
<?xml version="1.0"?>
<?xml-stylesheet type='text/xsl' href='style.xsl'?>
<cleanup description="Set of cleanup XML files" version="1.0">
       <CleanupObj path="D:\AP_P\cleanup\apollo.xml"
       <CleanupObj path="D:\AP_XY\cleanup\cleanme.xml"
       <CleanupObj path="D:\TEST\cleanup\whatamess.xml"
</cleanup>
```

The <cleanup> tag basically only has one attribute tag called <CleanupObj> which refers to a standard Cleanup configuration file. The [path] of the custom cleanup configuration file is restricted to reside on the local file system. The custom cleanup file path must always start with local volume names and may not contain a UNC path.

Cleanup performance.

Be aware of the fact that when extending the default cleanup run, all additional cleanup configurations are executed in sequence starting with the default cleanup configuration first. A complete run with many rules could take quite a long time depending on how many cleanup rules per configuration file need to be executed and how many files and folders need to be checked. Each potential file or folder must be queried and sorted for its modified time attribute and on large filesystems with many files and folders this may take quite a while.

When not to use NTFS compression.

NTFS compression on folders and their files should be used sparingly. Compression takes place per file and may cause a significant performance impact. NTFS compression is therefore not suited for I/O intensive file systems.

Common cleanup configuration rule examples.

The following are a number of examples how the cleanup script can be used to perform common file system maintenance.

Example 1: This will remove the file C:\Windows\Logboek\Config\Test.log if it is older than 60 days.

```
<fsobject type="FILE" path="%SYSTEMDRIVE%\LOGBOEK\Config\Test.log"
age="60" />
```

Example 2: This will remove all files in C:\Windows\TMP and its subfolders older than 1 day.

```
<fsobject type="FILES" path="%SYSTEMROOT%\TMP" age="1" />
```

Example 3: This will remove all files with wildcard "*.tmp" in C:\Users and its subfolders older than 100 days.

```
<fsobject type="FILES" path="C:\Users" age="100" Include="*.tmp"/>
```

You may use an Include and Exclude wildcard in the same rule.

Example 4: This will remove all subfolders in C:\Windows\Temp older than 14 days.

```
<fsobject type="FOLDERS" path="%SYSTEMROOT%\TEMP" age="14"
include="*" />
```

Example 5: This will remove all files and subfolders in C:\inetpub\logs\LogFiles older than 90 days. The 100 newest files and subfolders in that folder will not be removed.

```
<fsobject type="ALL" path="%SYSTEMDRIVE%\inetpub\logs\LogFiles"
age="90" keep="100" />
```

Archive configuration rule examples.

Example 1: This will move files older than 60 days in C:\Windows\Logboek and all its subfolders in an archive called C:\Windows\Logboek\archive.zip. The folder structure is retained inside the archive.

```
<fsobject type="ZIPFILESFOLDERS" path="%SYSTEMDRIVE%\LOGBOEK"
age="60" />
```

Example 2: This will move the files older than 60 days of each subfolder in the folder C:\Windows\Logboek in an archive per subfolder. The folder structure is retained inside each individual archive.

```
<fsobject type="ZIPHISTORY" path="%SYSTEMDRIVE%\LOGBOEK" age="60"
/>
```