|  |
| --- |
| Podcast Utilities |
| Podcast Utilities |
| User Guide |
|  |
| **Derek Wilson** |
| **July 2011**  **Version 1.3.4** |

|  |
| --- |
| How to use and configure Podcast Utilities |

# Contents

[1. Contents 2](#_Toc298682736)

[1. What is Podcast Utilities? 4](#_Toc298682737)

[1.1. Benefits 4](#_Toc298682738)

[2. Installing Podcast Utilities 4](#_Toc298682739)

[2.1. Prerequisites 4](#_Toc298682740)

[2.2. Copying the distribution 4](#_Toc298682741)

[2.3. Distribution contents 5](#_Toc298682742)

[3. DownloadPodcasts 6](#_Toc298682743)

[3.1. Typical Usage 6](#_Toc298682744)

[3.2. Command Line 6](#_Toc298682745)

[4. SyncPodcasts 6](#_Toc298682746)

[4.1. Typical Usage 6](#_Toc298682747)

[4.2. Command Line 6](#_Toc298682748)

[5. PurgePodcasts 6](#_Toc298682749)

[5.1. Typical Usage 6](#_Toc298682750)

[5.2. Command Line 6](#_Toc298682751)

[6. GeneratePlaylist 7](#_Toc298682752)

[6.1. Typical Usage 7](#_Toc298682753)

[6.2. Command Line 7](#_Toc298682754)

[7. Using the utilities 8](#_Toc298682755)

[7.1. Downloading to a PC and then synchronising to a device 8](#_Toc298682756)

[7.1.1. Setting up 8](#_Toc298682757)

[7.1.2. Key elements in the control file 8](#_Toc298682758)

[7.1.3. Workflow 10](#_Toc298682759)

[8. Control File Format 11](#_Toc298682760)

[8.1. Example Control File 11](#_Toc298682761)

[8.1.1. Global Section 12](#_Toc298682762)

[8.1.1.1. sourceRoot 12](#_Toc298682763)

[8.1.1.2. destinationRoot 13](#_Toc298682764)

[8.1.1.3. playlistFilename 13](#_Toc298682765)

[8.1.1.4. playlistFormat 13](#_Toc298682766)

[8.1.1.5. freeSpaceToLeaveOnDestinationMB 13](#_Toc298682767)

[8.1.1.6. sortfield 13](#_Toc298682768)

[8.1.1.7. sortdirection 13](#_Toc298682769)

[8.1.1.8. maximumNumberOfConcurrentDownloads 14](#_Toc298682770)

[8.1.1.9. feed section 14](#_Toc298682771)

[8.1.1.10. format 14](#_Toc298682772)

[8.1.1.11. downloadStrategy 14](#_Toc298682773)

[8.1.1.12. namingStyle 14](#_Toc298682774)

[8.1.1.13. maximumDaysOld 15](#_Toc298682775)

[8.1.1.14. deleteDownloadsDaysOld 15](#_Toc298682776)

[8.1.2. Podcast Section 15](#_Toc298682777)

[8.1.2.1. folder 15](#_Toc298682778)

[8.1.2.2. pattern 16](#_Toc298682779)

[8.1.2.3. number 16](#_Toc298682780)

[8.1.2.4. sortfield 16](#_Toc298682781)

[8.1.2.5. sortdirection 16](#_Toc298682782)

[8.1.2.6. feed section 16](#_Toc298682783)

[8.1.2.7. url 16](#_Toc298682784)

[8.1.2.8. format 16](#_Toc298682785)

[8.1.2.9. downloadStrategy 16](#_Toc298682786)

[8.1.2.10. namingStyle 16](#_Toc298682787)

[8.1.2.11. maximumDaysOld 16](#_Toc298682788)

[8.1.2.12. deleteDownloadsDaysOld 16](#_Toc298682789)

# What is Podcast Utilities?

PodcastUtilities are a set of utilities designed to help manage digital media on removable drives.

## Benefits

There are many sophisticated media management programs available for organising digital media however the key differences that Podcast Utilities has are

* Portable  
  Podcast Utilities is a “portable” application, in that it does not need to be installed and can be run from any drive type, local network or removable flash drive
* Configurable  
  Almost any synchronisation solution can be configured, for example copy 3 of this podcast, all of these podcasts, in this order, and leave a certain amount of space on the device. The number of concurrent downloads can easily be configured.
* Lightweight  
  The application does not use large amounts of memory in a system tray icon all the time its not being used, it only uses system resources when its being run.
* Scriptable  
  Podcast Utilities lends itself to being run as a scheduled task or from a script. The API is “headless” and can easily be called and used from other applications
* Open Source  
  The source is open and available
* Free  
  There is no cost to using Podcast Utilities for any use.
* No need for an iTunes account  
  Podcasts promoted on iTunes can be downloaded however there is no need to have an iTunes account.

# Installing Podcast Utilities

Podcast Utilities does not require any special installation, just copy the files from the dirstibution ZIP into any folder and run any of the utilities. The utilities can be run from local hard disks, network shares or removable media such as flash drives.

## Prerequisites

The following are required to install PodcastUtilities

1. A machine running Windows XP / Vista / 7
2. .NET Framework 3.5

## Copying the distribution

Unzip all the files from the distribution ZIP into any folder.

You will need to create your own control XML file for your specific media files, an example file is provided in the distribution ZIP along with this document.

## Distribution contents

The distribution contains the utilities in a binary form for running on Windows. The utilities (exe files) and their supporting dlls can be copied to any folder. Before you can use the utilities you will need to setup your control file. An example control file is included in the distribution.

If you are a developer and want to write code using PodcastUtilities then you need to add a reference to PodcastUtilities.Common.dll, this is the only file you need from the distribution though you may find it useful to refer to the Documentation.chm file for documentation on the API. PodcastUtilities.Common.dll is designed to be used “headless”, it makes no calls to win forms, web forms or console output.

The source for the console applications: DownloadPodcasts, SyncPodcasts and PurgePodcasts provides an example of using the PodcastUtilities.Common assembly.

# DownloadPodcasts

DownloadPodcasts is the command to be used to download media from all the feed urls specified int he control file. You need to have internet connectivity for this command to work.

## Typical Usage

Use this command to download podcasts either directly to a media playing device attached to your computer or to download media to your computer and then use SyncPodcasts to copy a subset to your media player.

## Command Line

The command line to run DownloadPodcasts is

DownloadPodcasts <controlfile>

Where <controlfile> is the filename of the XML file to control how DownloadPodcasts will work. For example if your control file is called MyPhone.XML then the command line would be

DownloadPodcasts MyPhone.xml

# SyncPodcasts

SyncPodcasts is the command that will copy digital media from a source location to a destination.

## Typical Usage

SyncPodcasts is typically used when you have a large number of digital media files and you want to transfer a subset of these files to a removable drive. Only files that are not present in the destination are copied, any files that have been removed in the source are also removed in the destination.

## Command Line

The command line to run SyncPodcasts is

SyncPodcasts <controlfile>

Where <controlfile> is the filename of the XML file to control how SyncPodcasts will work. For example if your control file is called MyPhone.XML then the command line would be

SyncPodcasts MyPhone.xml

# PurgePodcasts

PurgePodcasts is the command that will remove old podcasts that have been downloaded

## Typical Usage

PurgePodcasts is typically used to remove old downloads. The number of days it takes to be considered to be old is configured in the control file on a fed by feed basis, you can also disable the purging in any feed.

## Command Line

The command line to run PurgePodcasts is

PurgePodcasts <controlfile>

Where <controlfile> is the filename of the XML file to control how PurgePodcasts will work. For example if your control file is called MyPhone.XML then the command line would be

PurgePodcasts MyPhone.xml

# GeneratePlaylist

GeneratePlaylist is the command that will create a playlist file from a folder structure

## Typical Usage

GeneratePlaylist is typically used when you have a large number of digital media files that you have copied to a location and you would like to generate a playlist for all the files.

## Command Line

The command line to run GeneratePlaylist is

GeneratePlaylist <controlfile>

Where <controlfile> is the filename of the XML file to control how GeneratePlaylist will work. For example if your control file is called MyPhone.XML then the command line would be

GeneratePlaylist MyPhone.xml

GeneratePlaylist is the action performed by SyncPodcasts after all the copying has been done.

# Using the utilities

This section covers how the utilities might be used together to manage podcasts

## Downloading to a PC and then synchronising to a device

You want to download podcasts using a windows PC (either downloading to the hard disk or a memory stick) and then sync to a device.

### Setting up

To setup for this environment copy all the files from the ZIP to a folder on your hard disk or on a memory stick.

Do remember that removable media such as memory sticks can be referred to by different drive letters on different machines, however you can use the “Computer Management” console to permanently set the drive letter for a removable media device on a given computer. You can launch the Computer Management console by typing “compmgmt.msc” into a command prompt



In Windows 7 select Action -> All Tasks -> Change Drive Letters and Paths

### Key elements in the control file

The global section of the control file looks like this

<podcasts version="1.0">

<global>

<sourceRoot>.\media</sourceRoot>

<destinationRoot>W:\Podcasts</destinationRoot>

<playlistFilename>podcasts.wpl</playlistFilename>

<playlistFormat>wpl</playlistFormat>

<freeSpaceToLeaveOnDestinationMB>1000</freeSpaceToLeaveOnDestinationMB>

<sortfield>name</sortfield>

<sortdirection>asc</sortdirection>

<maximumNumberOfConcurrentDownloads>20</maximumNumberOfConcurrentDownloads>

<feed>

<maximumDaysOld>31</maximumDaysOld>

<format>rss</format>

<namingStyle>pubdate\_etitle</namingStyle>

<downloadStrategy>high\_tide</downloadStrategy>

</feed>

</global>

* All downloads will be store in a subfolder call media.
* When synchronising my device will appear as drive W:
* I use WPL format playlists
* Leave at least 1GB on the download drive and the destination device
* Download 20 at a time
* Only download episodes published in the last month
* Name the files using the published date and the episode title
* After I have listened to an episode on my device, I want to be able to delete the episode on my PC and have the delete synchronised across to the device so I use “high\_tide” as a strategy, which is also why I want to have the published date in the file name

Feeds are genrally in two groups. There are feeds where I want to listen to every episode and they are formatted this this

<podcast>

<feed>

<url>http://www.hanselminutes.com/hanselminutes\_MP3Direct.xml</url>

</feed>

<folder>Hanselminutes</folder>

<pattern>\*.mp3</pattern>

<number>-1</number>

</podcast>

* I do not specify deleteDownloadsDaysOld so PurgePodcasts will never delete an old episode I need to manually delete episodes I have listened to but I do not want to miss an episode so this is fine.
* I have used “high\_tide” strategy so when I have listened to an episode I delete it on my PC and SyncPodcasts will delete the episode on my device.
* I want to copy all the epicodes to my device so I set number to -1. For some feeds I limit the number to be copied to 3 for example, then SyncPOdcasts will copy the oldest 3 episodes to my device and as I delete them on my PC will keep the device topped up until there are no more available.
* You may need to change pattern if the feed uses a different format for its files.

Then there are feeds that I subscribe to that I just want to keep the last n episodes on tap but I may not get round to listening to them. These feeds are configured like this

<podcast>

<feed>

<url>http://downloads.bbc.co.uk/podcasts/worldservice/wbc/rss.xml</url>

<deleteDownloadsDaysOld>31</deleteDownloadsDaysOld>

</feed>

<folder>World Book Club</folder>

<pattern>\*.mp3</pattern>

<number>3</number>

</podcast>

* deleteDownloadsDaysOld is set to 31 so I PurgePodcasts will delete any episodes older than that.
* high\_tide strategy is used to the old episodes will not be downloaded again
* number is set to 3 so SyncPodcasts will ensure that I have the last 3 podcasts published in the last month available on my device, as time goes on these podcasts are automatically replaced with the current month’s episodes, a bit like TiVo for your podcasts.

### Workflow

I tend to get my podcasts once or twice a week. The workflow is

1. Attach my device to the PC
2. Run PurgePodcasts to automatically remove any old podcasts
3. Manually delete any episodes I have listened to
4. Run DownloadPodcasts to get all newly published podcasts to my PC
5. Run SyncPodcasts to get the desired subset of podcasts onto my device

# Control File Format

All the utilities make use of a control file to control how they operate. The format of the control file is XML, so all <elements> must have a closing element </elements>The root element of the file is <podcasts>

## Example Control File

The control file is an XML file and must be a legal XML file, in that elements must have a start and an end. The elements for example <sourceRoot> must be specified in the correct upper and lower case as shown here. This is an example of a control file

<podcasts version="1.0">

<global>

<sourceRoot>.\downloads</sourceRoot>

<destinationRoot>W:\Podcasts</destinationRoot>

<playlistFilename>podcasts.wpl</playlistFilename>

<playlistFormat>wpl</playlistFormat>

<freeSpaceToLeaveOnDestinationMB>2000</freeSpaceToLeaveOnDestinationMB>

<sortfield>name</sortfield>

<sortdirection>asc</sortdirection>

<maximumNumberOfConcurrentDownloads>20</maximumNumberOfConcurrentDownloads> <feed>

<maximumDaysOld>31</maximumDaysOld>

<namingStyle>pubdate\_etitle</namingStyle>

<downloadStrategy>high\_tide</downloadStrategy>

<format>rss</format>

</feed>

</global>

<podcast>

<feed>

<url>http://downloads.bbc.co.uk/podcasts/fivelive/tms/rss.xml</url>

<deleteDownloadsDaysOld>31</deleteDownloadsDaysOld>

</feed>

<folder>Test Match Special</folder>

<pattern>\*.mp3</pattern>

<number>-1</number>

</podcast>

<podcast>

<feed>

<url>http://www.hanselminutes.com/hanselminutes\_MP3Direct.xml</url>

</feed>

<folder>Hanselminutes</folder>

<pattern>\*.mp3</pattern>

<number>-1</number>

</podcast>

<podcast>

<folder>From Our Own Correspondent</folder>

<pattern>\*.mp3</pattern>

<number>5</number>

</podcast>

</podcasts>

If this control file was used with the SyncPodcasts utility then it would copy all the mp3 podcasts in “Test Match Special” and “Hansleminutes” folders followed by the first five mp3 files in the folder “From Our Own Correspondant” The source files are in a folder called downloads in the current folder and the destination folder is in “w:\Podcasts”. A WPL playlist will be generated and at least 2GB of space will be left on the drive.

If it was used with DownloadPodcasts then it would download all the episodes of Test Match Special and Hansleminutes published in the last 31 days. It would attempt to download 20 episodes at a time.

### Global Section

There is only one global section and it has settings that can only be set once or default values that can be overridden for each podcast

#### sourceRoot

This is the path to the source media files. It can be a relative or absolute path. The sourceRoot is ued by DownloadPodcasts as the root folder to download episodes to, SyncPodcasts uses it as the place to read media from.

If it is a relative path (for example “.\media”) it is relative to the current working directory, usually this is the current directory if you are running the utility from a cmd prompt.

An absolute path (for example “m:\media”) will remain unaffected by where the utility is run from. Do remember that removable media can be referred to by different drive letters on different machines, however you can use the “Computer Management” console to permanently set the drive letter for a removable media device on a given computer. You can launch the Computer Management console by typing “compmgmt.msc” into a command prompt



In Windows 7 select Action -> All Tasks -> Change Drive Letters and Paths

#### destinationRoot

SyncPodcasts uses this as the path that we shall copy to, GeneratePlaylists uses it as the root folder to find media in. This must be an absolute pathname for example p:\media. GeneratePlaylist

#### playlistFilename

SyncPodcasts and GeneratePlaylist uses this as the name of the file to write the playlist that contains all the media in the destination. The playlist contains all files in the destination not just the files copied in this run.

#### playlistFormat

The format the playlist should be in. Possible values are wpl or asx, (upper or lower case).

WPL format is a windows playlist for Windows Media Player and Windows Mobile based devices, ASX is a format used by many media players. Generate the playlist format that works with your player.

#### freeSpaceToLeaveOnDestinationMB

The space in MB to leave in the device, for example 2000 is 2GB.

When using SyncPodcasts copying will stop when there is less that this space left, however if there is less that this space left when SyncPodcasts is run then files will not be deleted to free up space.

When using DownloadPodcasts downloading will stop when there is less that this space left.

#### sortfield

The field to be used to sort the media files. If we want to copy the “first” 5 files of a given podcast then this setting is used to determine which files are first.

If the value of this setting is creationtime (upper or lower case) then the files are sorted by the date and time the file was created otherwise any other value means they are sorted by name.

Many podcasts incorporate a numbering system into the filenames so that using the filename will work fine however sometimes they either have no number or it is a random id in those cases you may want to sort the files on the bases of the date the file was downloaded.

If you use the DownloadPodcasts utility to download files then you can use the namingStyle setting to control the filename to ensure that the filename can be sorted by publishing date.

This setting can be overridden on a podcast by podcast basis.

#### sortdirection

This field is also used to control the selection of media files, possible values are desc or asc , (upper or lower case) for descending or ascending sort order.

This setting can be overridden on a podcast by podcast basis.

#### maximumNumberOfConcurrentDownloads

This element is only used by DownloadPodcasts and controls how many downloads are running at the same time. Your computer or the server that holds the podcasts may limit the number of connections it will accept.

#### retryWaitInSeconds

This element can be used to resolve any file locking issues when using slow flash drives. If you are running a large number of concurrent downloads in the same feed then all the downloads will attempt to update the high tide mark in the state file. You may see a warning message saying that the state file was locked and the downloader will retry. If after a number of retries the file is still locked then an error will be thrown.

This element is the number of seconds to wait before retrying writing to the state file. You may find a higher number for example 15 or 20 seconds will help on slow flash drives, you may also find that reducing the number of concurrent downloads will help.

#### feed section

This section is only used by DownloadPodcasts and PurgePodcasts

#### format

The only supported podcast feed format is RSS

#### downloadStrategy

This is the mechanism that will be used to download episodes in a feed

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| all | All episodes in the feed will be downloaded, if a feed keeps all the episodes that have been published in it then all the episodes will be downloaded, if you delete an download then it will be replaced when DownloadPodcasts is next run |
| latest | Only the latest episode in a feed is downloaded |
| high\_tide | When first run on a new feed all the episodes are downloaded, subsequently only new episodes are downloaded. This is achieved by recording the last episode successfully downloaded in a state.xml file and only downloading episodes that have been published since this high tide date.  You can reset the high tide value by deleting the state.xml file and all episodes will then be downloaded again. |

#### namingStyle

This is the style to use to name the file downloaded from a feed. The possible values are

|  |  |
| --- | --- |
| **Value** | **Meaning** |
| etitle | The episode title, as specified in the feed will be used. Any characters that cannot be used in a filename will be replaced with an underscore. |
| pubdate\_etitle | Published date in the YYYY MM DD HH MM (to enable the files to be sorted by published date) followed by the episode title, as specified in the feed. Any characters that cannot be used in a filename will be replaced with an underscore. |
| pubdate\_url | Published date in the YYYY MM DD HH MM (to enable the files to be sorted by published date) followed by the filename element from the end of the URL |
| pubdate\_title\_url | Published date in the YYYY MM DD HH MM (to enable the files to be sorted by published date) followed by the feed title (the folder used to store the episodes) followed by the filename element from the end of the URL |
| pubdate\_folder\_title\_url | The files are named in the same manner as pubdate\_title\_url and then the files are placed in a folder named after the month the episode was published in the form YYYY\_MM. Effectively this will group podcasts from the same month into a separate folder. |
| <anything else> | The filename element at the end of the URL for this episode |

#### maximumDaysOld

This can be used to prevent old episodes from being downloaded. The age of a podcast is calculated by comparing the current date time with the published date time in the feed.

#### deleteDownloadsDaysOld

This element is only used by PurgePodcasts and is used to remove old podcasts episodes that were published more than the specified number of days ago.

Please note that PurgePodcasts can only determine the date an episode was published if you use a namingStyle (see above) that records the publish date in the filename, otherwise the date the file was created on your local disk will be used.

This element is optional, if it is omitted or has a value of zero then purging will not be done for the specified feed.

### Podcast Section

After the global section there can be any number of podcast sections. Each section describes one folder of media files to be synced or podcasts to be downloaded.

Each section must have a value for folder, pattern and number and can optionally override sortfield and sortdirection.

If you want to use DownloadPodcasts to download the episodes then you will need a feed/url element, you can optionally override any other element in the feed section of the global section.

#### folder

SyncPodcasts uses this as the folder relative to the source that will be used to copy from and relative to the destination to copy to.

For example if the sourceRoot is “C:\Podcasts” and the destinationRoot is “X:\Media” and folder is “fred” then we will copy from “C:\Podcasts\fred” to “X:\Media\fred”

GeneratePlaylists uses this as the folder to find media to add to the playlist.

It is also used by DownloadPodcasts as the folder relative to the sourceRoot to download files to and also as the title of the podcast feed if the title is to be included in the episode file name using the namingStyle element.

#### pattern

SyncPodcasts uses this as the filename pattern to look for when copying files, for example \*.mp3 will match all the mp3 files.

#### number

SyncPodcasts uses this as the number of files to copy. -1 means that all file in the source folder will be copued.

#### sortfield

This is the same as the setting in the global section, if it is missing then the one from the global section is used

#### sortdirection

This is the same as the setting in the global section, if it is missing then the one from the global section is used

#### feed section

This section is only used by DownloadPodcasts and PurgePodcasts

#### url

This is the url to the podcast feed, often this will be displayed as “RSS Feed” on a web site. If this is not present then DownloadPodcasts will ignore the podcast element.

#### format

This is the same as the setting in the global section, if it is missing then the one from the global section is used

#### downloadStrategy

This is the same as the setting in the global section, if it is missing then the one from the global section is used

#### namingStyle

This is the same as the setting in the global section, if it is missing then the one from the global section is used

#### maximumDaysOld

This is the same as the setting in the global section, if it is missing then the one from the global section is used

#### deleteDownloadsDaysOld

This is the same as the setting in the global section, if it is missing then the one from the global section is used