WTV-MetaRenamer (version 0.23)

Contents

Important Windows 7 SP1 Information

Please note that if you are using a non-English version of Windows 7 and you have installed Service Pack 1, you **WILL** need to edit the configuration file. Please read the section *WTV-MetaRenamer XML File* to understand what you need to do.

If you are not using Windows 7 Service Pack 1, or you are using the English version, the default configuration file will allow the script to operate safely.

Summary of changes from previous version

- New feature to skip protected recordings. If this is a feature you want to use, you need to edit the WTV-MetaRenamer XML file and setting **skip_if_protected** to true. By default, the script will **not** skip protected recordings in order to preserve previous behaviour.

Introduction

Windows Media Center records programmes in a proprietary container with an extension of WTV. Recordings have somewhat cryptic filenames based on the broadcast programme name, the channel broadcasting the programme and the date & time of recording.

This filename format is not particularly helpful if you want to retain recordings as part of a structured library, where the contents will typically be something like:

```
<Programme name>
       <Season number>
              <Recordings>
```

and the recordings have filenames of a format like “<Programme name> - S<season>E<episode> - Episode title”.

The purpose of WTV-MetaRenamer is to examine the files in a nominated directory and use what metadata it can find to query against The TV Database so that the correct programme name, season number and episode number can be determined and the file renamed appropriately.

The code and documentation is maintained on CodePlex: http://wtvmetarenamer.codeplex.com/.

Prerequisites

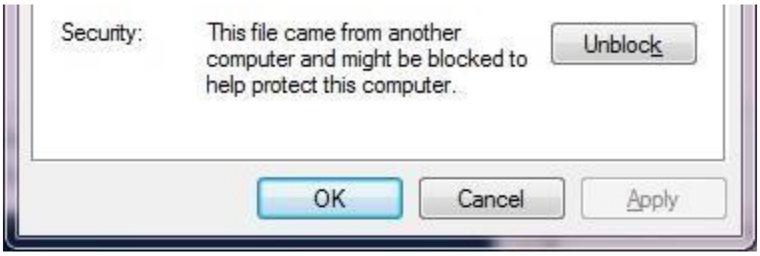
WTV-MetaRenamer requires the following:

- PowerShell v2 or later. Please see http://support.microsoft.com/kb/968929 for more details.

Once PowerShell has been installed, you will need to change the default execution policy in order to allow you to run unsigned scripts. This will require administrative rights on the computer concerned. To change the policy, execute the following command in the PowerShell command window:

```
Set-ExecutionPolicy RemoteSigned
```

This will allow you to run unsigned scripts locally but require remote scripts to be signed. You will also need to *unblock* the script. This is done by right-clicking on the PS1 file, choosing **Properties** and then clicking on the **Unblock** button and then click on **OK**:



Security: This file came from another computer and might be blocked to help protect this computer.

You will also need to unblock the file Ionic.Zip.DLL using the same steps described above.

If you experience problems or you are not concerned with running signed scripts, you can use this command instead:

```
Set-ExecutionPolicy Unrestricted
```

This may be required in order to get the script working on Windows 8.

Installation

At the simplest level, you can place the contents of the WTV-MetaRenamer Zip file into the same folder that contains the WTV files you want to rename.

If you prefer to store scripts in a separate folder, you can do so. You need to have the file Ionic.Zip.DLL (included as part of the download) in the same folder. There is also an XML file that is used to configure the script without having to edit the script itself. The file is, by default, called WTVMetaRenamer.XML, but you can specify the path and filename for the XML file as a commandline parameter to the script. This allows the script to be executed with different operational parameters just by pointing to a different XML file. The script does not modify the contents of the configuration XML file.

The XML **folder** that is part of the Zip file you've downloaded from the WTV-MetaRenamer project can be placed anywhere you like, but the script **does** write to this folder so you need to bear that in mind when considering where to place it.

WTV-MetaRenamer XML File

This file has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
   <default_language>[string]</default_language>
   <xml_cache>[path to folder]</xml_cache>
   <recordings>[path to folder]</recordings>
   <accept_single_bme>[true | false]</accept_single_bme>
   <move_to>[path to folder]</move_to>
   <move_to_single_folder>[true | false]</move_to_single_folder>
   <move_to_season_folders>[true | false]</move_to_season_folders>
   <season_folder_name>[string]</season_folder_name>
   <specials_folder_name>[string]</specials_folder_name>
   <season_number_format>[string]</season_number_format>
   <episode_name_format>[string]</episode_name_format>
   <multi_episode_format>[string]</multi_episode_format>
   <create_series_folder_if_missing>[true | false]</create_series_folder_if_missing>
   <delete_if_dest_exists>[true | false]</delete_if_dest_exists>
   <rename_if_dest_exists>[true | false]</rename_if_dest_exists>
   <move_episode_if_unmatched>[true | false]</move_episode_if_unmatched>
   <skip_if_protected>[true | false]</skip_if_protected>
   <min_age>[number of days]</min_age>
   <change_char from="[one or more characters]" to="[zero or more characters]" />
   <ignore_series>[series ID]</ignore_series>
   <only_series>[series ID]</only_series>
   <create_undo_logs>[true | false</create_undo_logs>
   <create_processing_logs>[true | false]</create_processing_logs>
   <log_path>[path to folder]</log_path>
   <move_unmatched_series>[path to folder]</move_unmatched_series>
   <move_unmatched_episodes>[path to folder]</move_unmatched_episodes>
   <move_duplicate_episodes>[path to folder]</move_duplicate_episodes>
   <move_ignored_series>[path to folder]</move_ignored_series>
   <attribute_title>[string]</attribute_title>
   <attribute_date_created>[string]</attribute_date_created>
   <attribute_subtitle>[string]</attribute_subtitle>
   <attribute_program_description>[string]</attribute_program_description>
   <attribute_recording_time>[string]</attribute_recording_time>
   <attribute_broadcast_date>[string]</attribute_broadcast_date>
   <attribute_protected>[string]</attribute_protected>
   <convert_to_dvrms>[true | false]</convert_to_dvrms>
   <delete_wtv_after_conversion>[true | false]</delete_wtv_after_conversion>
   <move_wtv_after_conversion>[path to folder]</move_wtv_after_conversion>
   <commercial_metadata_path>[path to folder]</commercial_metadata_path>
</config>
```

The **`default_language`** value is explained in the section **Language Handling** below.

The **`xml_cache`** value allows you to specify where the XML folder has been stored. If there is no value (as shown in the XML file included in the download), the script will expect to find it in the same folder as where it is being run from.

The **`recordings`** value allows you to specify where the WTV files are that you want renamed. If there is no value, the script will look in the same folder as where it is being run from.

The **accept_single_bme** value is a Boolean (you can specify 0, 1, false, true, no, yes) and if TRUE will cause the script to rename episodes if the BestMatchEpisode function can only find one episode that is a good match. Please use this with care as even single matches can sometimes be the wrong match if the metadata is useless! The default value (i.e. if nothing is specified) is false.

The **move_to** value must only be present in the XML file if you want the recordings to be moved elsewhere when the script has determined the new filename. The value is the path to the folder that you want the recordings moved to. If you want the recordings just renamed, this value must **not** be present.

It is possible to have more than one **move_to** value in the configuration file. This is useful if you have got a library of TV recordings that are spread over multiple disks.

By default, when the script **moves** recordings rather than renaming them, it tries to do so by adopting the <series>/<season>/<file> structure. If, however, you just want all renamed files to be moved to a single folder without any structure to it, set **move_to_single_folder** to be true.

If you want recordings to be stored in separate folders for each series, but you want the script to be able to move unmatched episodes (explained below), you need to set **move_to_season_folders** to be false. This is because unmatched episodes don't, by definition, know which season they below to.

The series folder is looked for in each of the **move_to** locations specified. The script is looking for a folder with a series name that matches that returned by TheTVDB. If none is found, the script will automatically create a series folder in the first location specified.

Once the series folder has been found or created, the script looks for a season folder for the episode. If no such season folder exists, the script will create one automatically. There are two values that control the text and format used – **season_folder_name** and **season_number_format**.

season_folder_name specifies a string and this is the name you are using to label season folders. If no value is specified, the script will use **Season**, but you can specify whatever string you are using on your system.

specials_folder_name specifies a string that is used if the episode number for a recording is 0. If no value is specified, the script will use **Specials**, but you can specify whatever string, e.g. **Season 0** if you preferred each season to be numbered.

season_number_format specifies a **format** string that PowerShell uses to convert a number to a string. If none is specified, the default is "0#" which means that any season number less than 10 will be prefixed with "0".

episode_name_format specifies how PowerShell will rename the files. The default is **{0} – S{1}E{2} – {3}**. Powershell replaces {0} with the series name, {1} with the season number, {2} with the episode number and {3} with the episode name. Some users prefer to use full-stops instead of dashes, and no spaces, in which case the value to specify would be **{0}.S{1}E{2}.{3}**. Note that you must **not** put quotes around the value.

multi_episode_format specifies how the script will name a recording that contains 2 episodes. The default is **{0}-{1}** and the script replaces {0} with the first episode number and {1} with the second episode number. The resulting string is then put into {2} in the **episode_name_format** output. So, by default, multiple episodes will be Eep1-ep2. If you want them to be, say, Eep1-Eep2, you need to set **this** value to be **{0}-E{1}**.

create_series_folder_if_missing is a Boolean (you can specify 0, 1, false, true, no, yes) and has a default value of TRUE. The script will always create the series folder under the **first** path specified in the **move_to** setting. If that isn't always going to be the correct location, set this to false and manually create the series folder.

delete_if_dest_exists is a Boolean (you can specify 0, 1, false, true, no, yes) and has a default value of FALSE. If set to true and the script decides that a recording already exists – either when renaming or moving – then this duplicate will be deleted. The default operation is not to delete the recording but to leave it untouched and the name unaltered.

rename_if_dest_exists is a Boolean (you can specify 0, 1, false, true, no, yes) and has a default value of FALSE. If set to true and the script decides that a recording already exists when moving recordings, this duplicate will be renamed according to the determined programme name, season and episode number. The default operation is not to rename the recording but to leave it untouched and the name unaltered. The intended value of this recording is if you want to manually check whether or not a duplicate recording is of better quality then the currently stored recording. By renaming the file, it says you looking at the metadata and then checking TheTVDB to work out what the season and episode details are.

move_episode_if_unmatched is a Boolean (you can specify 0, 1, false, true, no, yes) and has a default value of FALSE. If set to true and the script cannot match the episode's metadata against that retrieved from TVDB, the recording will be moved – without being renamed – into the appropriate series folder. Note that **move_to_season_folders** must be set to FALSE for this to work.

skip_if_protected is a Boolean (you can specify 0, 1, false, true, no, yes) and has a default value of FALSE. If set to true and the file has the Protected flag set, the script will not carry out any further processing and will leave the file completely untouched.

min_age specifies the number of days that the recording must have existed for before the script performs any operations on the file. If no value is provided, all recordings are considered.

change_char allows you to get the script to map on or more characters to another one. You can have multiple **change_char** entries in the XML file. Each entry has two values specified – from and to. The **from** value is a string of one or more characters. The script looks for each of these characters in the new episode filename to see if they need to be changed. The **to** value is a string of ZERO or more characters. If there are any characters in the **from** value that match, they get changed into the **to** value. If the **to** value is empty (i.e. “”), the matching characters are removed.

Some examples:

<change_char from=”*” to=”x”> will change any occurrence of an asterisk to a lower-case x.

`<change_char from=”&amp;” to=”and”>` will change any ampersand characters to the word **and**.

`<change_char from=”?/\” to=””>` will remove any occurrences of question marks, forward or backward slash characters.

The second example is an important one to understand. There are certain characters that cannot be represented just as themselves. The key ones that might appear in programme names are **& “ < >** - which are represented in the configuration file with `&amp;` `&quot;` `&lt;` and `&gt;` respectively.

If you want to use the character mapping feature but aren’t sure if you have got the configuration entries correct, you might want to have a single recording to work on initially and use the undo rename script to reverse the rename process if the script doesn’t get it right.

`ignore_series` specifies the ID of a series which, if matched, will cause the script to ignore any recordings for that series. You can have multiple entries in the XML file, e.g.:

```
<ignore_series>72227</ignore_series>
<ignore_series>80348</ignore_series>
```

`only_series` specifies the ID of a series which, if matched, will cause the script to process recordings for that series, and only that series. You can have multiple entries in the XML file, e.g.:

```
<only_series>72227</only_series>
<only_series>80348</only_series>
```

The default is for both lists to be empty, which means that the script will process all recordings. If you want to control which recordings the script processes, you can use **one** of the lists. In other words, you can specify values for the `ignore_series` list **OR** the `only_series` list, but not both. If you try to specify both, the script will stop running. `create_undo_logs` and `create_processing_logs` control whether or not the undo logs and the processing logs are created, respectively. The undo log is a PowerShell script which, if run, will reverse the steps taken, with the exception of any files that the script deleted. The processing log is a text file that show the processing steps taken by the script. This file will always include the verbose output, regardless of the verbose setting in the script. `log_path` allows you to specify where the log files are created. The default is the same folder as the recordings. `move_unmatched_series`, `move_unmatched_episodes`, `move_duplicate_episodes` and

`move_ignored_series`, if set to folder paths, cause the script to move any programmes that are either in an unmatched series, or that cannot be precisely matched on the episode name, or are a duplicate of existing recordings, or are in an ignored series respectively.

`attribute_title`, `attribute_date_created`, `attribute_subtitle`, `attribute_program_description`, `attribute_recording_time`, `attribute_broadcast_date` and `attribute_protected` only require values if (a) you are using Windows 7 Service Pack 1 or later, and (b) you are not using the English version of the operating system. These attributes must be set to the localised equivalent of “Title”, “Date Created”, “Subtitle”, “Program Description”, “Recording time”, “Broadcast date” and “Protected” accordingly so that the script can locate these file attributes. If you are not sure what the localised versions are, please contact the script author.

`convert_to_dvrms`, if set to true, will cause the script to convert the WTV recording into the older DVR-MS format. The conversion process does **not** normally delete the WTV recording, unless

`delete_wtv_after_conversion` is set to true. Alternatively, you can specify a folder path for **`move_wtv_after_conversion`** and the WTV file will be moved to that folder once the DVR-MS file has been created.

`commercial_metadata_path` allows you to specify a location for the script to check for XML commercial data files. By default, the script will look in the recordings directory.

Running WTV-MetaRenamer

To run the script, it is recommended that you open a PowerShell command window, change to the folder holding the script and then run it with a command such as:

```
./wtv-metarenamer.ps1
```

The script takes four optional parameters:

- **`-configurationfile`** can be used to specify the location of the XML configuration file. This is useful if you want WTV-MetaRenamer to behave differently under certain circumstances and therefore want the flexibility of multiple configuration files without having multiple copies of the PowerShell script.
- **`-singlefile`** can be used to specify the name of a single file to be processed. Please note that this must just be the leafname; the script will determine the recordings directory from the configuration file or default setting.
- **`-interactive`** can be used to instruct WTV-MetaRenamer to list multiple matches when they are encountered so that you can specify which one you want the script to use for further processing. This is used when there are multiple matches for a series or for an episode name. For all occurrences, it is possible to press RETURN by itself to skip that entry – the behaviour is then as if the –interactive switch had not been used, i.e. the script moves on to the next item.
- **`-verbose`** turns on additional logging output in the PowerShell console.
- **`-whatif`** stops the script from actually doing anything but outputs what it would do instead. This is useful for safely testing the script against recordings.

It is possible to run the script from the Explorer window by right-clicking and choosing “Run with PowerShell”. Note that the default action of double-clicking the file is to open it in the PowerShell editor – this is a safety decision taken by the PowerShell team.

Running the script directly from Explorer is not advised as the script does product output to tell you what has been done or of any problems encountered. The current release does not store this output in a log file, send it by email or do anything else with it.

For each WTV file found, the script will indicate that the file is being processed and start by trying to match the series name. Some programmes will not have a unique series name. If this occurs, the script will display the matching names and the series ID from TVDB. For example:

```
Processing Agatha Christie's Marple_ITV1_2010_08_30_21_03_19.wtv More than
one series matches series name 'Agatha Christie's Marple':
ID: 78895 ; Name: Agatha Christie's Marple
ID: 92391 ; Name: Agatha Christie's Great Detectives Poirot and Marple
```

Note that this output is normally logged in the processing log. It will only be displayed in the console if the script is running with –interactive or if the configuration file has set **`create_processing_logs`** to false.

If the script is able to **precisely** match the series name against the results returned from TvDB (as it can in the above example) then it will do so.

However, if there isn't a precise match (e.g. if the programme name is **Merlin** then there will be multiple matches), the script will not assume a match and will either list all the matches for you to later edit the SeriesList.XML file or, if the **–interactive** switch has been specified, will list them for you to make the choice straightaway. If **`move_unmatched_series`** has been set to a folder path, these recordings will be moved to that folder.

If you have recordings that you want the script to move but there is no metadata available from TVDB, it is possible to create a manual entry in the SeriesList.xml file that follows the same format as other series but has a **negative** series ID. The value used is not important but the fact that it is negative tells the script not to try and do any metadata matching.

Once the script has matched the recording against a series, it will attempt to determine which season and episode are correct for the recording. A number of tests are conducted against the metadata stored in the recording file. Depending on the accuracy or otherwise of the metadata, there may be some situations where the script cannot make a determination:

1. If the programme has an episode name that appears more than once for that programme. For example, Columbo has two episodes called "It's All in the Game" in TVDB.
2. Where there isn't a precise match of the text that the script can find against the names used in TVDB. In this situation, the script will try to find the best matches. If there is one such "best match" and **`accept_single_bme`** is set to true then the recording will be renamed using that match.

Programmes that do have a good match for the metadata against TVDB are renamed, and a PowerShell script created to provide an "unrename" opportunity to reverse the actions taken by the script.

If there are multiple possible matches and the **`-interactive`** switch has been specified, the possible matches will be listed and you will be able to choose the one you think is the appropriate match in order to have the recording renamed accordingly.

If a recording cannot be matched and **`move_unmatched_episodes`** has been set to a folder path, the recording will be moved to that folder.

Language Handling

The TvDB site supports multiple languages for programme information. However, quite a few programmes have the same programme name regardless of the language used. It is therefore necessary to configure the script appropriately if you have recordings in a language other than English.

The language can be set in two places:

1. The main configuration file
2. The SeriesList XML file

If the language is set in the main configuration file, this is considered to be the **default** language. What this means is that when the script decides the series ID for a given programme, if the SeriesList file does **not** specify a language, the default language is used.

If the language is set for a specific series in the SeriesList file, that is used as the language whenever recordings for that series are found, regardless of the language specified in the SeriesList file.

So, if you record mainly in (for example) French, you would set the default language to **fr**. If you then have some programmes that are recorded in English, you would edit the series entry to add:

```
<Language>en</Language>
```

Important note: the script does **not** write the <Language> entry when adding a series to the SeriesList file. It is necessary to add this manually.

Matching episodes by date

Each recording always has one date stored against it – the date when the recording was made. Some recordings will also store a second date called “broadcast date”. In theory, this **should** be the date that this particular episode was first broadcast. However, if you don’t reside in the country of origin for the given series, the broadcast dates are unlikely to match those held by TheTVDB.

For example, if you live in the UK, Spooks is broadcast by the BBC and so the broadcast dates will match. However, if you live outside the UK, the likelihood is that the broadcast dates for Spooks will not match.

Since date matching is somewhat unpredictable, it can only be enabled on a per-series basis, and this requires manual editing of the SeriesList XML file.

SeriesList XML File

The SeriesList XML file is used by WTV-MetaRenamer to provide a clear relationship between a series name and a series ID. It can be used to provide clarity in the event of ambiguity for recordings.

The structure of the file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<Data>
  <Series>
    <seriesid>72227</seriesid>
    <SeriesName>Two and a Half Men</SeriesName>
  </Series>  </Data>
```

Under normal circumstances, WTV-MetaRenamer will automatically add to this file as new series are discovered from recordings. In the event that the script finds multiple matches, you will need to manually edit the file in, say, Notepad and add a fresh <Series> section. Taking the "Miss Marple" series as an example, WTV-MetaRenamer found two matches. The match we want is series 78895, so the following lines need to be added to the file, before the last line which reads </Data>:

```
   <Series>
      <seriesid>78895</seriesid>
      <SeriesName>Agatha Christie's Marple</SeriesName>
   </Series>
```

You must match the series name text precisely otherwise the script will not match against the entry in the XML file and will continue to report multiple matches from the database. The series name text **must** match that used by the **broadcaster** and **not** TvDB. It is **strongly** recommended that interactive mode is used with the script to add new series to the file to ensure that the correct name and spelling is used.

As noted above, if you want to configure the behaviour of the script so that a specific series is treated as being in a language other than the "default" language used by the script (which is set in the main configuration file), it is necessary to manually add a **`<language>XX</language>`** line to the appropriate series, where XX is the two letter code for the language.

If you want to use either broadcast date or recording date matching for a given series, it is necessary to manually add a **`<MatchBroadcastDate>true</MatchBroadcastDate>`** line or a **`<MatchRecordingDate>true</MatchRecordingDate>`** line to the appropriate series. It is **strongly** recommended that the first run of the script after making this change is done manually with the **–whatif** parameter so that you can check the date matching success rate.

Scheduling the script

If you want to run the script automatically on a scheduled basis, this can be done running the file “wtv-metarenamer.bat” which is in the Zip archive. You can then use the Windows Task Scheduler to run the file on a regular basis.

Please remember that any output from the script will be lost if you use this method, but the “Undo Rename” script file will still be created.

If you create a shortcut file that specifies:

"C:\Program Files\WTV-MetaRenamer\WTV-MetaRenamer.bat" -verbose –singlefile

(noting that the path to WTV-MetaRenamer may need adjusting) then you can drag and drop individual recordings onto the shortcut file and it will trigger running WTV-MetaRenamer.

Thanks to Scott Bussinger for the suggested improvements to the bat file.