Table of Contents

GETTING STARTED GUIDE

Some basic help can be found within MCEBuddy itself. Hover of the text, buttons or titles to get help on what they are for.

OPEN THE ILLUSTRATED USER GUIDE (PDF or Word Document).

MCEBuddy 2.x User Guide.docx

MCEBuddy 2.x User Guide.pdf

If you are unable to open the attached guides, refer to the text version below.

The 5 Step Guide to Getting Started with MCEBuddy

1. Start MCEBuddy from the Window Start Menu and click on *Settings* on the main screen
2. Double click "Convert to MP4" on the *Conversion Tasks* pane
3. Change the *Destination Directory* to the directory you want to place the output file and click OK
4. Double click Windows Default in the *Monitor Locations* pane
5. Change the *Monitor Path* to point to the directory where you video/recording are that need to be converted, then click OK
6. Now click OK to close the settings page and then click *Start*

YOU'RE DONE!
Welcome to MCEBuddy

The Slightly Longer Guide to Getting Started with MCEBuddy

1. Start MCEBuddy from the Windows Start Menu
2. Click on *Settings*
3. In the *Conversion Tasks* section double click on the default Conversion Task (Convert to MP4) to open it or click on *Add* in the Conversion Tasks pane to create a new Task
4. Enter the *Task Name* if you're creating a new task (give it any name)
5. Select the *Profile* you want to use to convert the video. The profile determines that format of the encoding and quality of the encoding. A brief description of each profile will be given in the box below.The default setting is to convert the video in a High Quality MP4 file. If you want you can change the format or quality by selecting the appropriate profile from the drop down list.
6. Select the *Destination Directory*, this is the place where the converted file will be placed.

7. *Remove Ad's*, by default MCEBuddy is set to remove commercials using a program called Comskip. If you want to keep the commercials change this to *None* or leave it as *Comskip* if you want to remove the Commercials from the video while converting.
8. Click Ok to close the Conversion Task page and return to the Settings page.
9. If you want MCEBuddy to automatically convert all video files in a particular directory then click on "Add" in the *Monitor Locations* pane or double click on *Windows Default* in the Monitor locations pane.
10. If you selected Add, enter a name to mark this location
11. Set to *Monitor Path* to the directory in which all your video's to be converted are lying(e.g. your Recorded TV directory).
12. *Search Pattern*, by default MCEBuddy will look for *ALL* video files in that directory. Say you want it to convert only WTV files, then enter **.wtv* to in the Search Pattern box.
13. Click OK to close the Monitor Location page and return to the Settings page
14. Click OK to close the Settings page
15. Click *Start* and MCEBuddy should get started with the conversions
16. You can always Manually add additional files to convert by clicking the *Add* button on the main page or drag and drop the file into the white box on the main page.
17. Click on the *Close* button to close the window and MCEBuddy will keep running the background.
18. You can always start MCEBuddy from the Start Menu to see the current status of the conversions.

INSTALLATION ISSUES

Sometimes MCEBuddy Installer will run into various issues while installing or uninstalling the application. Listed below are some of the common error messages and resolution steps. Please make sure you **DO NOT SKIP any step and ORDER is very important**.

1. ALWAYS perform a complete uninstall of the previous version of MCEBuddy before installing a new version.
2. ALWAYS delete the previous installation of MCEBuddy (by default in the *C:\Program Files\MCEBuddy2x*) before installing a new version of MCEBuddy.

How do I do a clean install of MCEBuddy

1. Uninstall MCEBuddy from the Start Menu or the Uninstall Programs Shortcut (Control Panel)
2. Delete the MCEBuddy Service: Find ***cmd.exe*** or *command.exe* in the Start Menu -> Accessories. RIGHT CLICK on it and select "***Run as Administrator***".
   In the command prompt type ***sc stop MCEBuddy2x*** and press Enter. This should stop the MCEBuddy service.
   Now type ***sc delete MCEBuddy2x*** and press Enter. This should delete the MCEBuddy service. Do not worry if it gives an error saying service does not exist. NOTE: If you do not run as an administrator then the service deletion will fail and MCEBuddy will not install.
3. Delete the installation directory (usually C:\Program Files\MCEBuddy2x) - you will have to check where MCEBuddy was installed
4. Delete the backup configuration files stored by MCEBuddy. Open the folder *%HOMEPATH%* (this is a windows variable, just type that into your Run Prompt or Windows Explorer). Delete the following 3 files if found,*mcebuddy.conf*, *profiles.conf* and *history*
5. Reboot the computer

How do I do a manually uninstall MCEBuddy

Follow the below steps to manually remove MCEBuddy. **CAUTION: ADVANCED EXPERIENCED REQUIRED OR YOU MAY END UP MAKING WINDOWS UNSABLE**

1. Stop the *MCEBuddy2x* service from the windows service manager (*Run Services.msc*) and close the MCEBuddy GUI/Application
2. Delete the *MCEBuddy2x* folder from the *Program Files* folder
3. Start an Administrative command prompt (right click on *cmd.exe* or *command.exe* and select *Run as Administrator*), type *sc delete mcebuddy*

4. **ONLY DO THIS STEP IF YOU ARE FAMILIAR ON HOW TO OPERATE THE REGISTRY. IT CAN STOP YOUR COMPUTER FROM OPERATING IF YOU MAKE ANY MISTAKES**. Open the Registry, navigate to the following registry keys, search for and delete entries or keys as appropriate with MCEBuddy2x in them
   HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Installer\Assemblies\ (delete keys)
   HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Installer\Products\ (delete key)
   HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\Folders (delete entries)
5. Reboot the computer

MCEBuddy Engine does not start . Windows Event Viewer has the following Error System.ServiceModel.AddressAlreadyInUseException: There is already a listener on IP endpoint 0.0.0.0:23332

CAUSE

MCEBuddy wasn't uninstalled properly and did not remove the service properly.

RESOLUTION

1. Uninstall and reinstall MCEBuddy
2. If that does not work, then follow the procedures for a clean install

Service name already exists

CAUSE

Previous uninstallation was not clean and the MCEBuddy service was left behind so the new installation cannot complete until the old service is removed.

RESOLUTION

1. Delete the MCEBuddy Service: Find *cmd.exe* or *command.exe* in the Start Menu -> Accessories. RIGHT CLICK on it and select "***Run as Administrator***". In the command prompt types*c stop MCEBuddy2x* and press Enter. This should stop the MCEBuddy service. Now types*c delete MCEBuddy2x* and press Enter. This should delete the MCEBuddy service. Do not worry if it gives

an error saying service does not exist. NOTE: If you do not run as an administrator then the service deletion will fail and MCEBuddy will not install.

2. Delete the entire installation directory (by default in the *C:\Program Files\MCEBuddy2x*).
3. Now try to install the new version of MCEBuddy

While UnInstalling it gives an error, Service Does Not Exist

CAUSE

Some antivirus programs such as Norton mistakenly identify MCEBuddy as an unwanted service and delete it. This causes the uninstallation to fail.

RESOLUTION

1. Goto Control Panel -> Uninstall a program
2. Select MCEBuddy and click *Change* and Select ***Repair***
3. After the Repair is complete you can now UnInstall MCEBuddy

I have a port conflict on my local machine (port is already in use)

CAUSE

Another program is using the port used by MCEBuddy. By default MCEBuddy uses port 23332 for communication.

RESOLUTION

1. Open MCEBuddy.conf
2. In the section *[Engine]*, change the *LocalServerPort* to another port number which is not used by any other application (choose from*1* to *65534*) (try a random port number greater than 1024 if your don't know which ports are used)
3. Reboot the computer OR Restart the MCEBuddy service (*Control Panel* ->*Services* -> *MCEBuddy 2x* -> Right click and select *Restart*)

An exception occurred in the OnInstallAfter event handler of System.ServiceProcess.ServiceInstaller. --> This access conrol list is not in canonical form and therefore cannot be modified

CAUSE

MCEBuddy tries to create the temp and log directories after installation. Sometime there is a security conflict with the previous version of MCEBuddy and the new version cannot create/delete the directories.

RESOLUTION

1. Delete the MCEBuddy Service: Find ***cmd.exe*** or *command.exe* in the Start Menu -> Accessories. RIGHT CLICK on it and select "***Run as Administrator***". In the command prompt type***sc stop MCEBuddy2x*** and press Enter. This should stop the MCEBuddy service.
   Now type ***sc delete MCEBuddy2x*** and press Enter. This should delete the MCEBuddy service. Do not worry if it gives an error saying service does not exist. NOTE: If you do not run as an administrator then the service deletion will fail and MCEBuddy will not install.
2. Delete the entire installation directory (by default in the *C:\Program Files\MCEBuddy2x*).
3. Now try to install the new version of MCEBuddy

COMMON ISSUES

With newer MCEBuddy builds (2.3.15 and later) the video conversions fail randomly (sometimes they work, sometime they don't)

Cause

When hardware acceleration is enabled, the stability of the conversion depends upon the stability of the graphics drivers. Older Intel graphics drivers are unstable and cause handbrake and quicksync to crash and conversions to fail randomly.

Resolution

Upgrade to Intel graphics drivers version 10.18.10.3496 or newer to fix the issue.

I'm using the WTV profile but the WTV videos created by MCEBuddy won't play on Windows Media Player or Windows Media Center

Resolution

This issue has been resolved with 2.3.15 and later versions of MCEBuddy.

I've setup MCEBuddy to monitor folders but it won't add files/not all files show up in the queue

Cause

The monitor path is invalid/incorrect

The file search pattern for the Monitoring folder does not match the files in the directory

The filters on the Conversion Task Settings page don't match the file metadata

The files are marked readonly or are locked by another process. MCEbuddy needs exclusive access to convert files.

Resolution

Check the monitor location directory configured in the monitor locations tasks
Check the file search pattern (e.g. *.*wtv*) for the monitoring location task. It may not be setup to accept all video/video files types in the directory being monitored.
Remove the readonly option on the files, right click on the file and select properties. *Uncheck*the *ReadOnly* box.
If the file is locked by another process, you'll have to find out which process and then close the process/release the file.
Check the Filename, Showname and Network name Filters on the Conversion Task Settings page

Refer to the MCEBuddy Advanced Commands page for more details on filename and showname pattern matching.

I'm adding files to MCEBuddy but it won't show up in the queue

Cause
The files are marked readonly or are locked by another process. MCEbuddy needs exclusive access to convert files.

The filters in the Conversion Task Settings do not match the Metadata on the file

Resolution
Remove the readonly option on the files, right click on the file and select properties. *Uncheck*the *ReadOnly* box. If the file is locked by another process, you'll have to find out which process and then close the process/release the file.
Check the Filename, Showname and Network name filters in the Conversion Task Settings page. If the filters don't match the metadata on the file, the files will not be added to the queue.

I keep getting MCEBuddy Service is unavailable. Please start from windows control panel or check engine connection.

Cause
Either the service has not been started

OR there is a conflict with another port

OR ShowAnalyzer is causing MCEBuddy to hang

Resolution

Check that the service has been started. Click Start -> Control Panel -> Windows Services -> Select MCEBuddy 2.x -> Right click and click Start

If there is a conflict with another port, MCEBuddy will fail to start, check the Windows Event logs for more information and see the Installation Issues for more details.
If you have ShowAnalyzer installed; there's a bug in ShowAnalyzer which causes it to hang. The MCEBuddy User Application makes a call to check the status of ShowAnalyzer which causes it to hang and it presents itself as not able to connect to service issue.

Uninstallation, rebooting the computer and reinstalling Showanalyzer usually solves the problem. If it persists, try to uninstall ShowAnalyzer and use Comskip instead.

I am unable to access files created by MCEBuddy from a remote computer on my shared drive

Cause

Permissions or owner are set incorrectly on the parent folder which causes the files created by MCEBuddy to inherit permission that don't match the shared folder permissions.

Resolution

Ensure that the destination folder and the files are inheriting the permissions from the parent folder and the parent folder has the permissions you need for others to view the files (e.g. Read/Write/Modify for Guests or any other user). Also ensure that owner for the parent folder is set to you.

My original WTV file works fine in Media Center Media Extender devices but I am unable to play back WTV files created by MCEBuddy in Media Center Media Extender devices

Cause

Some devices don't support h.264 video in WTV files (which is created by the WTV profile)

Resolution

Use the WTV Legacy profile which uses MPEG2 in WTV for older devices like Ceton Echo and Linksys DMA2100.

The other option is to use the WTV Unprocessed profile if your original video plays back correctly on your Extender.

I am unable to connect to the MCEBuddy engine using the Remote Client

Cause

- Firewall is blocking the ports

- Engine is not running

Resolution

- On the machine where MCEBuddy engine is installed, check *Enable remote access* from *Settings -> Advanced Settings -> Expert Settings.* This will allow MCEBuddy to automatically create an exception in the Windows Firewall to allow incoming connections to the engine on the configured port. This only works with Windows Firewall, if you have a custom firewall then you'll have the configure it manually.

- Ensure the MCEBuddy engine is running

MCEBuddy sometimes creates video with no audio

Cause: Some recording have multiple audio tracks, sometimes these audio tracks do no contain any audio.
Resolution: Which audio track is selected depends up on how MCEBuddy is configured. If "*Select the best soundtrack*" is checked, then the encoder (*handbrake*, *ffmpeg* or *mencoder*) will decide the best audio track to use. Often *Handbrake* can get this wrong while *ffmpeg* is better at it. If you're using this option, try to change the *order* in your profiles.conf to have *ffmpeg* as the first encoder.
The other option is to uncheck "*Select best soundtrack*", in which case MCEBuddy will choose the best audio track with the highest number of audio channels and highest bitrate.
If you want a specific audio track, add the file to the conversion queue, right click on the file and select*MediaInfo*. This will show all the available audio tracks in the file. If the file contains impaired audio tracks then note the language of the audio track which NOT impaired. Open *Settings -> Change Conversion Task*Settings -> *Advanced Settings* -> *Audio Language*. Select the audio language of the non-impaired audio track as determined from *MediaInfo* earlier. Do not worry if more than one audio tracks (impaired and non-impaired) have the same language. MCEBuddy will choose the non-impaired audio track when an*Audio Language* has been selected.

How do I prevent MCEBuddy from reconverting my videos

There are multiple options to prevent MCEBuddy from reconverting video files:

1. If you're upgrading from a version **before 2.1.7** please save the history file manually (*C:\Program Files\MCEBuddy2x\config\history*) and restore it after upgrading. (2.1.7 and above save the history file during uninstallation and restore it on installation)
2. If you're using 2.1.5 or greater you can use this option to manually add the files not to be converted to the history file in the following manner. In Windows Explorer select all the files you want to add to the history file. Drag and Drop them into MCEBuddy window (MCEBuddy should be started and it should start converting them). Now select all the files in the conversion queue (Click on the 1st file and press *Shift* + *End*) and then click *Delete*. It will stop converting all the files and add them to the History file and they will not be reconverted unless you add them manually.
3. Create a folder called *MCEBuddyArchive*. Put all files you don't want converted into this folder, MCEBuddy will not convert any files inside this folder. Please note, when the *Archive Files* option is selected in the advanced settings page, MCEBuddy does this automatically.
4. Open the Conversion Task you're created. In the Advanced Settings is a *File Name Match* (read the documentation MCEBuddy Advanced Commands). You can specify a negative filter (with ~) to specify which files NOT to convert OR a regular filter to specify which files to convert.
5. If you're using the Monitor Locations option to automatically convert videos by monitoring a folder for new videos **AND** the Monitor Folder and Destination Folder are the **same**, change the Monitor Location *Search Pattern*. E.g. all source videos are WTV files, all destination videos are MP4. Open the Monitor Location task and set the *Search Pattern* to *.wtv, so now MCEBuddy will only pick up WTV files to convert. Or you can use a -ve filter again like [videos];~*.mp4 or *.*;~*.mp4 to convert all files except .mp4 (the [videos] or *.* is important to specify since there must be atleast one +ve filter to select files)

My audio is out of sync with my video

Cause

There are many reasons for the Audio to be out of sync with Video. It can vary from corrupted frames to country format differences.

Resolution

There are many things can do to resync the Audio with the Video.

- If you're using a FFMPEG profile, trying setting the *-ss* parameter to *-ss 30* to see if that solves the problem.
- Try setting the *<encoder>-audiodelay=skip* for the profile you're using.
- If point 2 does not work, then you can use the same parameter and replace skip with a +ve or a -ve number. This will advance or retard the audio by that many seconds. This can help bring the Audio back into sync with the Video through trial and error. Make sure you replace *<encoder>*with the profile encoder you're using. e.g. *ffmpeg-audiodelay=1.25*
- If you're using MEncoder in your profile and your audio is going out of sync ONLY when commercial removal is enabled then try to use the *mencoderedlskip=true* option to use the MP4Box

fallback method to strip out commercials. (again this will only help if commercials removal is causing the desync in audio while using MEncoder).

- If you're using *FFMPEG* or *Handbrake* and *.mp4* or *.m4v* outputs then try playing with*CutMP4Alternative=true*
- Try setting *UniversalCommercialRemover=true* if the above doesn't work.

Refer to the MCEBuddy Advanced Commands for more details on the above.

The video I'm converting succeeds but has garbled video and audio

Cause

The video is copy protected.

Resolution

Record a non-copy protected video. There is no way to convert copy protected videos. To check for Copy Protection, right click on the video -> click properties. Check the Video properties page and look for the copy protection field.

When I use the MKV profile, I am unable to play it back with Windows Media Player

Cause

There is no MKV Media Splitter for Windows Media Player

Resolution

Download and install a MKV Media Splitter, e.g. *Haali Media Splitter*

When I use the NoRecode profile, I am unable to play it back with Windows Media Player

Cause

Windows does not have a MPEG-2 video splitter installed

Resolution

Download and install a MPEG-2 video splitter. e.g. *FFDSHOW*

My video is stuttering/broken after conversion

Cause

1. Source video is corrupted or broken
2. Check the log files, if the *average rate of dropped and/or duplicate frames* are high, MCEBuddy is unable to detect the correct video frame rate which leads to stuttering video.

Resolution

For corrupted original video, check the quality of the recording signal. A low quality signal can lead to corrupted recordings.

For incorrect frame rate issues, open *mcebuddy.conf*. Under the section *[FFMpegBackupRemux]*, replace*auto* in *-r auto* with the correct frame rate of the video. (this can be done via trial and error or with the help of 3rd party software)
e.g. -r 25
e.g. -r 30*1000/1001

Refer to MCEBuddy Advanced Commands for more details.

My original video has 6 channel/multichannel audio but my converted file only has stereo/2 channel audio

Cause

MCEBuddy by default limits all the output audio to 2 channel audio (stereo) to preserve compatibility with many 3rd party players and Windows Media Player (which doesn't support multichannel audio).

Sometimes when source video files have multiple audio tracks with different number of channels, MCEBuddy downmixes the audio to preserve compatibility.

Resolution

In MCEBuddy, in the *Conversion Tasks* page, under *Advanced Settings*, enable *Multichannel Audio*. This will tell MCEBuddy to output multichannel audio if the source video has multichannel audio.
If your source video has multiple audio tracks, then in the *Conversion Tasks* page, under *Advanced Settings*, select the *Audio Language* or enter the *3 digit Audio Language code* corresponding to the audio track you want to keep. You can find out which audio tracks are there in the source file by right clicking on the file in the conversion queue from the MCEBuddy main *Status Page Window* and selecting*MediaInfo*.

System does not wake up from standby/suspend even though I've enabled the Wake option in the settings page

OR

System goes to sleep even if I have UN-Checked Allow Sleep in the Settings page

Cause

Windows (Vista and 7) sometimes by default sets the Operating System to disable Wake Up timers. this is especially true for portable systems and laptops. When the System Wakeup timers are disabled no application can wake up the system from standy/suspend mode. Also sometime computer BIOS have disabled wake timers and wake support.

Resolution

Make sure Allow Wake from USB (or equivalent power wake options) are enabled in the BIOS.

Make sure Wake Timers are enabled in the system. Follow the steps below to enable Wake Timers.

1. Open the *Control Panel* (start -> Control Panel)
2. Look for *Power Options* and open it
3. For the active plan (the plan with the dot next to it), click on *Change Plan Settings*
4. Click on *Change Advanced Power Settings*
5. Click on the + next to the *Sleep* option
6. Click on the + next to *Allow Wake Timers*
7. Set the option to *Enable* for *On Battery* and *Plugged In*
8. Click *OK*

MCEBuddy does not start and stop at the time entered in the Settings page

Cause

The wake and start time are accidentally set in a 12 hour mode

Resolution

Set the start and stop time in a 24 hour mode. i.e. 01:30 PM should be set as 13:30

All my conversions keeps failing OR files are immediately removed from the queue after starting the conversion OR random files keep failing at random points OR my conversion never finishes (it just hangs)

Cause

MCEBuddy is running out of diskspace and the conversions fail in between at random points when it runs out of disk space. This is aggrevated when multiple conversions are being used.
Antivirus software is interfering with the conversion

Resolution

a) Sometimes Antivirus software such as AVAST are known to interfere with MCEBuddy. These software lock down the file when it is accessed by MCEBuddy and this causes the conversion not to start, hang or randomly fail. Try to disable the Antivirus and see if that solves the problem.

b) MCEBuddy needs *3 TIMES* the size of the file being converted as free space on the *INSTALLATION DRIVE*. i.e. If MCEBuddy is installed on C drive and you're converting a 4GB file then you need 12GB of free space on C drive.
NOTE: MCEBuddy needs *1.5X free space* for ***EACH file being converted simultaneously***. i.e. if you have 3 simultaneous conversions and each file is 4GB in size, you need 1.5x3x4 = 18GB of free space on the installation directory!
If you do not have enough space in the installation drive, you have 2 options
1. Reduce the number of simultaneous conversions
2. Set the temp directory to point to a different drive where there is more space. Refer to MCEBuddy Advanced Commands for more details.

I pressed Start, I can see the files in the queue but Conversion task is showing Idle or Paused

Cause

The Start/Stop conversion time scheduler has been set on the settings page and the current time is not within the specified conversion period

The Pause button has been pressed

Resolution

There are the following options:

1. Wait for the conversion time period specified in the settings page and MCEBuddy will start automatically
2. Click settings and change/disable the Start/Stop time period in the settings page
3. Click on the *Resume* button on the GUI page

MCEBuddy is not able to see or work with files on a network path/share

Cause

Network path is password protected

Network path is already mapped by Windows

No username has been entered in the authentication pages of MCEBuddy settings

Network path is offline

Resolution

a) If you're using a network path/share in the *Monitor Task* to monitor a network folder or if you're using a network path/share as the destination folder in the *Conversion Task*, make sure you enter the username and password in the MCEBuddy Credential box next to the Monitor Path or Destination Path text box by clicking on the button. Even if you shared the network share without a password or anonymous or guest or everyone, **you MUST enter atleast *Guest* in the username of the credential page**. This is because MCEBuddy uses LocalSystem account and not the local user account to authenticate on the network, so the mapped drives or logged on user authentication will not be accessible by MCEBuddy.

b) If you're using the *Add File* button or dragging and dropping files/directories into the queue then you need to add the network login credentials to *Settings* -> *Expert Settings* -> *Network Authentication*. This MUST match the same credentials that have been used to access the network share via Windows Explorer or if the network share has been mapped to a drive. See point (c) below.

c) Check if the path has already been mapped by Windows. To check the mappings, open a command prompt (Start -> Run -> cmd). Type *net use* and it should show you the mapped drives. Make sure you are using the same credentials that have been used to map the network shares/drives in Windows. Windows only allows for one set of credentials per network share. If you don't need the mapped drive/paths you can disconnect them using the command *net /delete <pathname>*.

d) Make sure the network drive is connected and accessible.

Windows is VERY FINICKY. You cannot map the same Network share more than once with different credentials.

If you've connected to the remote computer using Windows Explorer or the Run command, then you've already logged in with one set of credentials. If MCEBuddy now tries to connect with a different set of credentials it will FAIL since Windows will reject it.

I'm not able to use a network path for Monitor Path or Destination Path. It gives an error or nothing happens

Cause
Network path is password protected
Network path is already mapped by Windows
No username has been entered in the authentication page of MCEBuddy settings
Network path is offline

Resolution
a) Make sure you enter the username and password in the MCEBuddy Credential box next to the Monitor Path or Destination Path text box by clicking on the button. Even if you shared the network share without a password or anonymous or guest or everyone, **you MUST enter atleast *Guest* in the username of the credential page**. This is because MCEBuddy uses LocalSystem and not the local user authentication and local system does not authenticate on the network.
b) Check if the path has already been mapped by Windows, this can create a conflict.
To check the mappings, open a command prompt (Start -> Run -> cmd). Type *net use* and it should show you the mapped drives. You'll have to disconnect the necessary paths (*net /delete <pathname>*).
Alternatively, map your path to a drive in window (*explorer* -> *tools* -> *map network drive*) and then don't enter any username and password in mcebuddy instead use the mapped drive.
c) Make sure the network drive is connected and accessible.

Windows is VERY FINICKY. You cannot MAP the same COMPUTER 2 times with different logins.

If you've connected to the remote computer using Windows Explorer or the Run command, then you've already logged in with one set of credentials. If MCEBuddy now tries to connect with the configured credentials it will FAIL since Windows will reject it.

1. Either use your own credentials and MAP the remote computer folder using Windows explorer and DO NOT enter any credentials in the Monitor Task or Conversion task pages for that remote computer. In this case MCEBuddy will use the Windows Explorer credentials.

OR

2. Enter the credentials in the MCEBuddy Monitor Task and Conversion Task page and let MCEBuddy connect to the remote computer.

Remember you CANNOT DO BOTH of the above, otherwise MCEBuddy will not be able to access the remote computer.

When I extract Closed Captions (CC) as SubTitles, the subtitles are out of sync from the start by a few seconds

Cause
By default all conversion profiles have a *-ss 3* in the video conversion parameters. While this is necessary for a smooth conversion, the *-ss 3* causes the conversion to skip the first 3 seconds of the video. This causes the subtitles generated to be out of sync by 3 seconds from the video
Resolution
In the Conversion Task settings page, next to the checkbox labled *Extract CC*, in the box labled *Offset*, enter *-3*. This will cause MCEBuddy to timeshift all the Subtitles back by 3 seconds which will compensate for the *-ss 3* shift caused by the profile settings.

I upgraded my OS or MCEBuddy or suddenly MCEBuddy seems to "hang" or have high CPU utilization while starting or adding files

Cause
Sometime a very large history file or a corrupted history file can cause MCEBuddy to hang.

Similarly other corrupted files can cause MCEBuddy to hang.

Resolution

1. Delete the history file (in the <MCEBuddy installation>\config directory (or you can click on *Show History* on the main GUI page and click *Clear History*)
2. If that does not work, do a clean install of MCEBuddy (refer to Installation Issues -> How do I do a clean install of MCEBuddy)

How do I use my custom EDL and Closed Caption/Subtitles files while converting the videos

Resolution
Place the *.EDL* and/or *.SRT* files along with the source video. MCEBuddy will automatically pick up these files (make sure *Remove Ads* is set to *Comskip* or *Showanalyzer* and/or *Extract CC* is enabled in the*Conversion Task* - MCEBuddy will use the custom EDL/SRT files before generating new ones). Also make sure they have the same name as the source video.
e.g. Source video -> MySerial.wtv
EDL File -> MySerial.edl
SRT File -> MySerial.srt

Refer to the MCEBuddy Advanced Commands for more details on Custom EDL/SRT Files.

I'm trying to convert my video but after a few minutes the converted file has no audio

Cause
TV shows sometimes change the audio formats between shows/advertisements which causes FFMPEG, Mencoder to fail.

Resolution
Calculate (in seconds) the amount of time the previous show/advertisements are recorded before the start of the main show, add a few seconds to it for safety and put that value in the *Start Trim* box under*Conversion Tasks -> Advanced Settings*.
E.g. There is 2 minutes and 30 seconds of video before the main show starts, put *155* in the *Start Trim*box.
Similarly calcuate the number of seconds after the show ends of video that is recorded (next show/advertisements) and put that value in the *End Trim* box.
This will make MCEBuddy trim the start and end of the video and leave just the show for conversions which should solve the problem of changing audio formats.

PROFILE CUSTOMIZATION

MCEBuddy Profiles Basics

MCEBuddy profiles are in the profiles.conf file. One can create new profiles or customize existing ones to tweak them and match you requirements which very often differ from country to country.

The basic intent of the profiles is to create the command line parameter to pass onto the encoders, commercial cutting routines etc.

MCEBuddy uses the following format for building a command line parameter to pass to the encoder:*GeneralParameters* + <*InputFile*> + *VideoParameters* + *AudioParameters* + <*OutputFile*>

Basic Format of a Profile:

[<Profile Name>]

e.g. [Convert to MP4]

Name of the profile as it will show up in MCEBuddy

Description=<Profile Description>

e.g. Description=Convert a file to MP4 format

Short description of what the profile does

order=<copy/ffmpeg/mencoder/handbrake>

e.g. order=ffmpeg,mencoder

Specify the order in which to use encoders (primary to backup) to convert the file. You can specify one, two or three encoders from the options listed above each separated by a comma. MCEBuddy will use the first encode to try to convert the file, if that fails then it will try the second and so on. In the above example it will try FFMPEG and if that fails then it will try MEncoder.
copy is a special case, where there is no recoding/encoding, the remuxed file (TS) incase of a WTV or DVRMS source video or the original source file is just copied to the output as the converted file without any processing.
NOTE: Each encoder in the order specified REQUIRES a <*encoder*>*-general*, <*encoder*>*-video*, <*encoder*>*-audio*, <*encoder*>*-audioac3* and <*encoder*>*-ext* parameter. If any of these parameters are missing the encoder will fail (except for *copy* encoder). See below for details.

<encoder>-general=<general parameters>

e.g. ffmpeg-general=-threads 0 -async 1

GeneralParameters: For each encoder in the order above the general encoding parameter needs to be specified for the encoder. The general parameters are the first in the command line parameters.

<encoder>-video=<video parameters>

e.g. ffmpeg-video=-ss 3 -vcodec copy

VideoParameters: For each encoder in the order these specific what video parameters are to be passed to the encoder. The video parameters follow the inputs file name while MCEBuddy builds the command line.

<encoder>-audio=<audio parameters>

e.g. ffmpeg-audio=-acodec copy

Audio Parameters: For each encoder in the order these specific what audio parameters are to be passed to the encoder. These are used if the inputs audio is NOT ac3. The audio parameters follow the video parameters while MCEBuddy builds the command line.

<encoder>-audioac3=<ac3audio parameters>

e.g. ffmpeg-audioac3=-acodec copy

AC3Audio Parameters: For each encoder in the order these specific what audio parameters are to be passed to the encoder if the inputs audio IS AC3. MCEBuddy will use EITHER the audio or the audioac3 parameters while building the command line depending on the inputs file audio format. The ac3audio parameters follow the video parameters while MCEBuddy builds the command line.

<encoder>-ext=<extension>

e.g. ffmpeg-ext=.avi

For each encoder in the order enter the extension of the output filename. Depending upon the extension the appropriate encoders, commercial cutting program and other parameters will be selected by MCEBuddy. Supported output formats include *.wtv, .wmv, .mkv, .avi*, *.m4v*, *.mp4*. *.mpg* and *.ts*

Note: As of v2.3.12 MCEBuddy supports almost all formats, however there are limitations on which encoder is being used, e.g. *MEncoder* works best

with *.avi*, *Handbrake* with *.mp4* and *.m4v* while *FFmpeg*can support nearly all formats.

Optional Parameters for the Profiles

<encoder>-unsupported=<extension + (optional) video codec + (optional) audio codec> (v2.3.3+)

e.g. mencoder-unsupported=ts+h264,wtv+h264+ac3

For each encoder in the order one can specify an unsupported combination of container/video codec/audio codec each separated by a+. Multiple conditions are separated by a , If the source video contains any of these combinations the conversion will be forcefully failed for the particular encoder and it will continue trying to convert using the next encoder specified in the *order* parameter. This can be useful when you're trying to separate encoder and file formats, e.g. you want to use ffmpeg for WTV HD recordings and MEncoder for everything else.

2pass=<true/false>

e.g. 2pass=true

If set to true then MCEBuddy will do a 2 pass during the conversion else 1 pass

<encoder>-remuxto=<extension>
e.g. mencoder-remuxto=.mp4

For each encoder in the order one can specific of the output file needs to be remuxed into a different format at the end after the first extension is created. E.g. ffmpeg will work best when creating an avi file. It can then be remuxed into mp4. MCEBuddy will choose the appropriate remuxer based on the input and output extension Supported extensions are *.wtv*, *.avi*, *.m4v*, *.mp4* and *.mkv*
Note: As of v2.3.12 MCEBuddy supports almost all formats.

RenameExt=<extension>
e.g. RenameExt=.mpeg

This is used to rename the file extension just before copying the file to the output directory.

<encoder>-UsingHardwareEncoding=<true/false> (v.2.3.15)+
e.g. handbrake-UsingHardwareEncoding=true

When *true*, this flag will tell MCEBuddy NOT to detect/auto adjust the hardware encoder codec or any hardware encoding related settings as this profile has been already optimized for hardware encoding.

Currently MCEBuddy only supports auto hardware enabling/tuning for handbrake, so this setting only applied to handbrake profiles. FFMpeg and MEncoder profiles needs to be manually written/optimized to handle hardware encoding.

Refer to the MCEBuddy Advanced Parameters page MCEBuddy Advanced Commands to learn about advanced Profiles tweaking

ADVANCED CONCEPTS

Advanced concepts to use MCEBuddy

MCEBuddy GUI Startup Options

MCEBuddy support the following options while starting the GUI (MCEBuddy.GUI.exe). Include these parameters by right clicking on the application short cut in the Start Menu and add them to the target path.

/startmin - Start the GUI minimized
/startengine - Force the engine to start (after connection with engine is established)
e.g. "*C:\Program Files\MCEBuddy2x\MCEBuddy.GUI.exe*" */startmin /startengine*
Note: The parameters will come after the double quotes close and a space between each parameter

MCEBuddy Command Line Interface (CLI)

MCEBuddy provides a CLI with limited capability to interact with the MCEBuddy engine as an alternative to the GUI. This can be useful when used with custom commands and scripts. Run the*MCEBuddy.UserCLI.exe* without any parameters to get a list of supported commands and usage details.

Tuning Comskip Commercial Detection

Comskip can be fine tuned to better detect and cut commercials for each country/region. There is a file names comskip.ini in the comskip directory in the MCEBuddy installation directory (by default C:\Program Files\MCEBuddy2x). The details of the tuning parameters can be found at:http://www.kaashoek.com/files/tuning.htm

If the video appears to get cut for about 3 seconds after the commerical ends (i.e. the commercial skipping cuts 3 seconds into the actual video post the end of the commerical), this could be because of the *-ss* settings in the *<encoder>-video* section of the profile. the *-ss 3* command is required to get rid of corrupted video at the beginning to avoid audio sync issues. Remove the *-ss 3* from the video parameters to fix this issue. Refer to the Audio Sync Issues section at the end of this document for details on the -ss command.

Multiple Audio Tracks in Converted File (v2.3.12+)

This only works with FFMPEG based profiles and if the source file has multiple audio tracks. Handbrake and MEncoder do not support converting/copying multiple audio tracks.

Select *<Default>* as the Audio Language in the conversion options AND add *-map 0:a -map 0:v* to the*ffmpeg-video* section for the FFMPEG profile and it will copy/convert all audio tracks.

Show Analyzer (v2.1.2+)

MCEBuddy supports ShowAnalyzer version 1.0 and greater. Once you download and install ShowAnalyzer it will show up as an option on the conversion profile settings page under commerical detection.

Filename and Showname Pattern Matching (v2.2.3+)
MCEBuddy supports multiple wildcard name matching to match shownames and filenames using the *;*operator. E.g. once can specify **.wtv;*.avi*.mpg* as the name selection criteria or *ncis*;house*.mpg* to match all names that start with ncis and house (ending with a .mpg).
If you want to specify a name avoid list, i.e. select all/abc files EXCEPT xyz, then prefix the name selection criteria with a ~ operator. E.g. ncis*;~ncis miami* will select all files that start with ncis but NOT those starting with ncis miami. E.g. *.*;~*.mp4;~*.avi will select all files except those starting with .mp4 and .avi
Name matching is case insensitive.

To select all default video types use the expression *[video]* which will match the following expression**.dvr-ms;*.wtv;*.asf;*.avi;*.divx;*.dv;*.flv;*.gxf;*.m1v;*.m2v;*.m2ts;*.m4v;*.mkv;*.mov;*.mp2;*.mp4;*.mpeg;*.mpeg1;*.mpeg2;*.mpeg4;*.mpg;*.mts;*.mxf;*.ogm;*.ts;*.vob;*.wmv;*.tp*
For advanced users MCEBuddy also supports REGEX expressions. To provide a regex expression prefix the regex matching pattern with *regex:* and then the regex pattern (this is a very advanced concept).
Multiple Conversion Tasks Customized by ShowName or FileName (v2.2.3+)
Refer to Filename and Showname Pattern Matching above to understand how to create patterns for filename and shownames.

You can create multiple conversions tasks in MCEBuddy. Each conversion task will process each file in the queue. Order of conversions tasks do not matter when creating filename and showname filters. If you want to create custom conversion tasks for different files/shows you can do so with the help of the *Filename* or *Showname* pattern matching filters in the Conversion Tasks Settings page, under Advanced Settings.
For e.g. if you want to create 3 tasks, one that processes all NCIS files, one that processes all Star Trek files and one that processes all other files (i.e. default). To do so you will first create 3 conversion tasks.

In the first conversion task, where you want to process NCIS, under *Filename* or *Showname* filters (depending on what you want to filter based on, the name of the file or the name of the show taken from the metadata), you will enter *NCIS** . This tell the conversion task to process only those files starting with NCIS.
In the second conversion task, where you want to process Star Trek files, under the *Filename* or*Showname* filter you will enter *Star Trek** .
In the third conversion task, you want to process ALL other files (ie default), then you will enter the following **;~NCIS*;~Star Trek** . The first * tells the conversion task to process ALL files, the~*NCIS** tells

the task NOT to process files starting with NCIS and _~Star Trek*_ tells the task NOT to process files starting with Star Trek, i.e. process all files except those which are being processed by the first two tasks.

Custom File Renaming Pattern (v2.1.6+)

MCEBuddy allows you to create your own custom file name and directory structure using the metadata stored in the file and downloaded from the internet.

The following commands are currently supported by MCEBuddy

- *%originalfilename%* - Name of the source file (without the path or extension)
- *%showname%* - Showname / Title
- *%episodename%* - Episode Name / Subtitle
- *%season%##* - Season No
- *%episode%##* - Episode No
- *%network%* - Network channel on the show was recorded (v2.1.10+)
- *%rating%* - Parental rating (v2.3.14+)
- *%premiereyear%* - Premiere Air Year (v2.3.14+)
- *%premieremonth% - Premiere* Air Month (v2.3.14+)
- *%premieremonthshort% - Premiere* Air Month Name Abbreviation (v2.3.14+)
- *%premieremonthlong% - Premiere* Air Month Full Name (v2.3.14+)
- *%premiereday% - Premiere* Air Day (v2.3.14+)
- *%premieredayshort% - Premiere* Air Day of Week Abbreviation (v2.3.14+)
- *%premieredaylong% - Premiere* Air Day of Week Full Name (v2.3.14+)
- *%airyear%* - Original Air Year
- *%airmonth%* - Original Air Month
- *%airmonthshort%* - Original Air Month Name Abbreviation (v2.3.11+)
- *%airmonthlong%* - Original Air Month Full Name (v2.3.11+)
- *%airday%* - Original Air Day
- *%airdayshort%* - Original Air Day of Week Abbreviation (v2.3.11+)
- *%airdaylong%* - Original Air Day of Week Full Name (v2.3.11+)
- *%airhour%* - Hour of air date (v2.3.6+)
- *%airhourampm%* - Air date hour in AM/PM (v2.3.12+)
- *%airminute%* - Minute of air date (v2.3.6+)
- *%recordyear%* - Record Year
- *%recordmonth%* - Record Month
- *%recordmonthshort%* - Record Month Name Abbreviation (v2.3.11+)
- *%recordmonthlong%* - Record Month Full Name (v2.3.11+)
- *%recordday%* - Record Day
- *%recorddayshort%* - Record Day of Week Abbreviation (v2.3.11+)
- *%recorddaylong%* - Record Day of Week Full Name (v2.3.11+)
- *%recordhour%* - Record Hour (v2.1.10+)
- *%recordhourampm%* - Record Hour in AM/PM (v2.1.12+)
- *%recordminute%* - Record Minute (v2.1.10+)
- *%ismovie%*<RenamePatternIfTrue,RenamePatternIfFalse> - If recording is a movie rename using True pattern, else rename using False pattern (v2.3.12+)

- *%issport%*<RenamePatternIfTrue,RenamePatternIfFalse> - If recording is a sport show rename using True pattern, else rename using False pattern (v2.3.15+)
- *%airingdbid%* - sageDbId from the SageTV Airing tag (v2.4.1+)
- *%mediafiledbid%* - sageDbId from the SageTV MediaFile tag (v.2.4.1+)
- \ - Directory Separator

E.g.: *TVShows\%showname%\Season %season%\%showname% - S%season%##E%episode%## - %episodename% - %airyear%_%airmonth%_%airday%*
Will produce: TVShows\CSI\Season 1\CSI - S01E02 - Best of times - 2012_01_06.<ext>

E.g.: *Recording\%ismovie%<Movie\%showname%,TVShows\%showname%\Season %season%\%showname% - S%season%##E%episode%##>-Converted*
Will produce:

Recording\Movies\Star Wars-Converted.<ext> (for a movie)

OR

Recording\TVShows\CSI\Season 1\CSI - S01E02-Converted.<ext> (for non movies)

The extension is added automatically depending upon the profile specifications.

The number of #'s at the end for Episode and Season indicates the number of digits in the written number. E.g. *%season%##* will give the output at *01* while *%season%###* will give the output at *001*. #'s are optional, leaving them out will write the number without any formatting. i.e. *%season%* will give *1* as the output.

Audio Sync/Missing Issues

Sometimes the audio and video go out of sync due to corruption at the start of the video. To solve this problem add the parameter ***-ss 10*** at the BEGINNING of the *<encoder>-video* parameter in the profile. DO NOT put this parameter in the general parameters section. (do not forget the - before the ss and make sure there is a space between ss and 10)
E.g. ffmpeg-video=-ss 10

This indicates that ffmpeg will skip the first 10 seconds of the video/audio after decoding it (if you put this in the general parameters section then the encoder will skip over first 10 which DOES NOT solve the problem, the video/audio need to be first decoded and then discarded to be put back in sync).

You can play with the number to find the right mix, it can be as low as 3 or as high as 30 depending upon the corruption. You'll be surprised how many videos are corrupted at the start and FFMPEG is very sensitive to corruption as far as audio sync goes. You often cannot see the corruption while play the video (it will look okay), these are at the frame level and often ignored by players.

Also try to play with the AudioDelay parameter (given above). Set to *skip* to see if that resolves the issue. If the Audio sync is still off, try setting the AudioDelay number to +ve or -ve to advance or retard the audio sync and see how it behaves. It is a trial and error process.

If you're using *FFMPEG* or *Handbrake* as the encoder with your profile **AND** your output extension is*.MP4* or *.M4V* **AND** your audio is going out of sync ONLY when you remove commercials (i.e. the Audio is in sync without commercials removal), try to use an alternative commerical stripping by using the command *CutMP4Alternative=true* in your profile.

If that doesn't work try using *UniversalCommercialRemover=true* in your profile. See above for more details.

Sometimes the Audio gets cut out completely in between a video. Try using *PreConversionCommercialRemover=true* in your profile. See above for more details.

Using Custom Filenames and Custom Metadata (v2.3.12+)

If you're using a WTV or DVRMS file, then MCEBuddy will extract the metadata from them directly.

Otherwise, MCEBuddy looks for a XML metadata file along with the original source video (with the *same source video filename*). It uses this file to extract Metadata information for non WTV/DVRMS files.

Apart from *XML* it can also extract metadata from *PROPERTIES* (SageTV) files, *ARG* (ArgusTV) files and*TIVO* files (MAK needs to be configured).

If it cannot find any custom metadata files, till try to extract the embedded metadata if the file supports it. E.g. MP4, M4V, MKV, WMV etc.

If none of the above metadata extraction techniques work, MCEBuddy can also extract metadata (Showname and Original Broadcast Date and Time, which can then be used to get download additional metadata from the internet) from the filename when set in NextPVR/nPVR format:

SHOWNAME_AIRDATE_AIRTIME.<ext>

Where:

AIRDATE (optional) - *YYYYMMDD*

AIRTIME (optional) - *HHMMHHMM* (Start-HHMM, End-HHMM)

e.g.: CSI_20140423_14301530.ts

(v2.4.1+) It can also extract the IMDB Id from the filename if the filename is given in the format

MOVIE_IMDBID.<ext>

OR

SHOWNAME_IMDBID.<ext>

where *IMDBID* is the IMDB Id for the Movie or Episode. The Showname for a series is taken from*SHOWNAME*.

Note: This is the Episode IMDB Id (specific for that exact Episode only) and NOT the Show IMDB Id.

E.g. Movie_tt0103064.ts OR Battlestar Galactica_tt0519763.ts

Also, if MCEBuddy find a XML file created during the conversion process (e.g. from Comskip), it will copy the XML file to to the destination directory along with the converted file.

Using Custom EDL Files (v2.3.12+)

MCEBuddy looks for a EDL file along with the original source video (with the *same source video filename*). If it finds the EDL file with the source video it copies the EDL file to the output directory along with the converted file.

MCEBuddy gives preference to the Custom EDL file over Comskip/ShowAnalyzer generated EDL files.

Using Custom SRT Files (v2.1.8+)

MCEBuddy looks for a SRT file along with the original source video OR in the temp working directory (with the *same source video filename*). If it finds the SRT file in either place it copies the SRT file to the output directory along with the converted video (with preference given to a SRT file in the temp working directory).

If *ExtractCC* is enabled in the *Conversion Task* -> *Advanced Settings* it will trim the Custom SRT file to be in sync with the EDL file (commercial removal). MCEBuddy gives preference to the Custom SRT files over the CC generated SRT file.

Comskip can be set to generate SRT files (*output_srt=1*) or CCExtractor can be used using from *Custom Commands* (if the ExtractCC from the GUI fails to work) to generate SRT files during the conversion process. Once generated, MCEBuddy will copy them to the output directory along with the converted file.

MultiThreading Support (v2.1.2+)

MCEBuddy by default calculates how many threads are required for the programs to run optimally. However this can be overridden manually by specifying the threads in the general parameters in the profiles file for each profile and encoder type.

Advanced parameters to tweak the conversion

(apply to each profile in profiles.conf)

(v2.3.14+ Profile parameters, when specified, override any conversion task options - e.g. SkipCropping and AutoDeInterlace)

Inserting Special Commands (v2.3.12+)
You can get MCEBuddy to insert special commands in the 4 sections of the profile

- *<encoder>-general*
- *<encoder>-video*
- *<encoder>-audio*
- *<encoder>-audioac3*

The following special commands will be replaced by MCEBuddy at runtime:

- *<source_without_ext>* - Source filename without extension
- *<source>* - source filename
- *<converted_without_ext>* - Output filename without extension
- *<converted>* - Output filename

E.g. mencoder-general = -sub "<converted_without_ext>.srt" -ss 3

will be replaced at runtime with (assuming the output file is c:\temp\test file.avi):

mencoder-general = -sub "c:\temp\Test File.srt" -ss 3

NOTE: MCEBuddy does not put quotes around the replacement parameters, some command expect quotes to be sure to put quotes where required.

ForceWTVStreamsRemuxing=true (v2.3.12+)
This parameter tells MCEBuddy to use DirectShow to extract the audio and video streams from the WTV file and remux them into a TS file. This has the advantage that uses Windows codecs, is fast and also support encrypted/Copy Protected content. The disadvantage is that it only support one audio and video stream in the WTV file.

NOTE: By default Streams remuxing is used for DVR-MS files and also as the last option for WTV files if all else fails.

AllowAllCopyRemuxing=true (v2.3.15+)
This parameter tells MCEBuddy to internally allow remuxing all video codec formats into a TS format without converting it to MPEG2 video first. This can be used when one wants to change the container format while retaining the original video without recoding it. E.g. from WMV to MP4 or WTV to MKV or WTV to TS, or even WMV to WMV and remove commercials etc.

NOTE: When using this option it possible that other underlying programs (like comskip, ccextractor) etc may not support the video codec being copied and may fail to function properly.

AllowH264CopyRemuxing=true (v2.3.12+)

This parameter tells MCEBuddy to internally allow remuxing H264 video into a TS format without converting it to MPEG2 video first. This can be used when one wants to change the container format while retaining the original H264 video without recoding it. E.g. from WTV to MP4 or WTV to MKV or WTV to TS, or even WTV to WTV and remove commercials etc.

For more details on this refer to this thread http://mcebuddy2x.codeplex.com/discussions/431449
NOTE: (v2.3.14+) By default this is enabled, also keep in mind that the default version of *Comskip* that comes with MCEBuddy is *SLOW* to detect H264 commercials. Use the *Donator* version of Comskip (http://www.comskip.org/) to speed up the H264 commercial detection.
(v2.3.12 - v2.3.13) By default this is disabled. The limitation on this is that default version of *Comskip* that comes with MCEBuddy *does not* support H264 commercial detection, however *ShowAnalyzer can be used*in it's place or use the *Donator*version of Comskip (http://www.comskip.org/) for H264 commercial detection.

UseWTVRemuxsupp=true (v2.3.11+)
This parameter tells MCEBuddy to use Remuxsupp FIRST to remux WTV files before trying other remuxers. This can help with some videos (rare) that not remuxed properly with FFMPEG or other remuxers but work with Remuxsupp. Usually Remuxsupp does not work well with many videos.
NOTE: This only helps in certain situations where files remuxed by FFMPEG are not able to be converted by MEncoder.

ForceEDL=true (v2.1.4+)
This is only effective when Comskip is enabled and Custom EDL files are not provided. When the commercial removal option is set to Comskip it tells MCEBuddy to force use the EDL file instead of the EDLP file from Comskip. EDL/EDLP files indicate the sections (timestamps) of the video to cut (commercials), however the timestamps differ slight for each video format, some use EDL and others EDLP. If your video/show is getting cut before/after commercials by a few seconds then try using this parameter.
NOTE: By default MCEBuddy uses EDL for .TS files and EDLP for all others

ForceEDLP=true (v2.1.4+)
This is only effective when Comskip is enabled and Custom EDL files are not provided. When the commercial removal option is set to Comskip it tells MCEBuddy to use the EDLP file instead of the EDL file from Comskip. EDL/EDLP files indicate the sections (timestamps) of the video to cut (commercials), however the timestamps differ slight for each video format, some use EDL and others EDLP. If your video/show is getting cut before/after commercials by a few seconds then try using this parameter.
NOTE: By default MCEBuddy uses EDL for .TS files and EDLP for all others

FixedResolution=true (v2.1.4+)

This tells MCEBUddy not to change source video resolution while converting (keep source resolution). When this parameter is set, MCEBuddy ignores the *Max Width* slider on the *Conversion Task -> Advanced Settings* page.
(v2.3.13+) When this is set, it also fixes the bitrate to what is specified in the profile. Normally MCEBuddy will adjust the bitrate as specified in the profile (which is optimized for 720 pixel width) to compensate for the change in converted video resolution (up or down) as limited by the *Max Width* slider. If the user does not want to limit the resolution then they should increase the *Max Width* slider in the *Conversion Task Settings -> Advanced Settings p*age all the way to the right.

SkipCropping=true (v2.1.4+)
This tells MCEBuddy not to autodetect crop information and skip cropping the video. This helps when the video edges are being cut off or if you have a video with no black bars on the sides to be removed, it will speed up the conversion process. Auto cropping can also sometimes skew the aspect ratio, so setting this will make MCEBuddy skip cropping.

ffmpeg-SubtitleBurn=true (v2.3.15+)
This parameter only works with ffmpeg profiles. If this is enabled, MCEBuddy looks for a valid SRT (subtitle) file and if it finds one during the conversion process, it will use that SRT file to burn the subtitles into the video during conversion. This only works if the video is being encoded (i.e. copy codec is NOT being used). This only works if there is a valid SRT file. So ensure that you have a SRT file along with the original video (MCEBuddy will copy it) or you have enabled *Extract Closed Caption* in Conversion Task settings. Once the subtitles have been successfully burnt the SRT file is deleted and not copied to the output.

<encoder>-AudioDelay=xxx/auto/skip (v2.1.2-2.2.18, 2.2.19+)
This parameter is set for each encoder type, e.g. ffmpeg, mencoder or handbrake
When xxx is set to *skip* then it tells MCEBuddy to skip auto correction of audio delay (sync). This is useful for some file formats such as AVI where audio gets skewed on auto correction.
If this parameter is set to a numerical value, then this parameter is used to manually specify the audio delay correction to correct audio sync, *xxx* is a +ve or a -ve number in seconds. If the value is equal to*auto*, it adds or subtracts the delay specified to the auto calculated audio delay for the video. A value of 0 means that MCEBuddy will skip any delay correction.
By default this this parameter is set to *skip*.
E.g. mencoder-AudioDelay=0.85

2ChannelAudio=true (v2.1.5 - v2.1.9, v2.2.13+)
When this parameter is set MCEBuddy limits the output audio to 2 channels irrespective of number of inputs channels
This can be useful to when specialized devices/software can read only 2 channels

By default MCEBuddy sets the output channels to the number of input channels
(As of 2.1.10, this can also be set in the Conversion Task GUI - profile overrides GUI)

MEncoderEDLSkip=true (v2.1.4+)
This parameter tells MCEBuddy not to use the EDL command with MEncoder to remove commercials from the video during conversion (which can save time but sometime cause the audio to go out of sync in rare cases).
If this parameter is set then MCEBuddy will remove the commercials AFTER the conversion is completed by using MP4Box to cut the commercials. Please do not set *CutMP4Alternative=true* when using this parameter.
This can be used to if converting with Mencoder is causing your Audio to go out of sync while removing commercials (it has no impact if commercial stripping is disabled) and none of the other parameters (*-ss 30* or *mencoder-audiodelay=skip*) do not work.
UniversalCommercialRemover=true (v2.3.11+)
When is parameter is set it tells MCEBuddy to use the Universal Commercial Remover after the conversion is complete (where as the default are very specific commercials remover functions for each file type, but they can sometimes fail as they are very sensitive to errors and sync issues). This helps when the standard profile is failing or causing audio issues during the commercial removal stage. While this cutter can leave some artifacts where the video is cut, however it works on ALL file types and will work successfully without any audio sync issues.

The other advantage of *UniversalCommercialRemover=true* is that it support files with multiple audio tracks and preserves them where as other mechanisms may or may not support depending up the file type.
(v2.3.12+) Setting *UniversalCommercialRemover=true* will force all commercials to be cut post conversion, unless *PreConversionCommercialRemover=true* is set. This can be used an option if commercial removal are failing or if a new file format is being used which is not natively supported by MCEBuddy.
PreConversionCommercialRemover=true (v2.3.12+)
When is parameter is set it tells MCEBuddy to use the Universal Commercial Remover to remove the commercials before the conversion (rather than the usual case of removing commercials after conversion). This helps when the conversion is failing during the commercial removal stage. While this cutter can leave some artifacts where the video is cut, however it works on ALL file types and will work successfully without any audio sync issues. This option is independent of *UniversalCommercialRemover*. If the preconversion commercial removal fails, it will automatically fall back to the post conversion commercial remover.
CommercialMergeTool=ffmpeg/avidemux (v2.3.14+)

There are two tools used to merge video segments after removing the commercials. This parameter can be set to force use of the tool. By default MCEBuddy will determine the best tool to use, however if the conversion is hanging or if you want this can be changed. The tools are *ffmpeg* and *avidemux.avidemux* is great for single audio tracks and for most general purposes but it can sometimes cause MCEBuddy to hang due to bugs in it. *ffmpeg* is a very stable tool, supports multiple audio tracks but sometimes it leaves artifacts when merging segments at the frame the segments are merged. Occasionally *ffmpeg* can fail the merging but will not hang MCEBuddy.
e.g. *CommercialMergeTool=ffmpeg*

CutMP4Alternate=true (v2.1.4+)

When is parameter is set it tells MCEBuddy to use an alternative mechanism for remove commercials from *MP4* and *M4V* files after the conversion is complete. By default MCEBuddy uses MP4Box to remove commercials during the last step, if this parameter is set it will use MEncoder to remove commercials. This helps sometime when the Audio Goes out of sync with the video after using Comskip/ShowAnalyzer but is in sync without Comskip. NOTE: This parameter is only effective when using
using *FFMPEG* or*Handbrake* encoders. When using mEncdoder as the encoder, this parameter has NO effect as the commercial will always be stripped during the conversion itself (unlike FFMPEG and Handbrake, who's commercials are stripped out AFTER the conversion is complete).

CommercialSkipCut=true (v2.1.4+)

If this parameter is set, it tells MCEBuddy to do the Commercial Scan but NOT to cut the commercials. Instead it will copy the generated EDL file to the output directory along with the converted file.
Comskip generates 2 types of files, EDL and EDLP. Use the *ForceEDL* and *ForceEDLP* command to specify which file to use. See above for more details.

NOTE: When this parameter is set, the Closed Caption/SRT files generated will NOT be trimmed to match the EDL file.

CopyPropertiesFile=true (v2.3.15+)

If this parameter is set, it tells MCEBuddy to copy the SageTV properties file if found along with the original video. The properties file is copied along with the converted file to the output directory.

CopyLogFile=true (v2.3.14+)

If this parameter is set, it tells MCEBuddy to copy the log file generated by comskip during commercial detection. The log file is copied along with the converted file to the output directory.

AutoDeinterlace=true (v2.3.14+)

If this parameter is set it tells MCEBuddy to automatically detect the scan type (interlaced, progressive or telecine) of the video and override the profile parameters to optimize the quality of the video.

Running Custom Commands (v2.1.8+)

MCEBuddy can provide the user with an option to run a single command at the following points in this order during the conversion process:

Refer to the Conversion Process Overview to understand the exact location of each Custom Command.

1. (v2.3.14+) At the very beginning of the conversion process just BEFORE extracting the metadata from the source file (before the remuxing or commercial detection or conversion). This custom command can executed with the following parameters. To do so **the first 4 parameters need to be defined** for the profile for which a pre conversion custom command needs to be run.
NOTE: Since this is run ***before*** the MetaData extraction, the only information available is about the original/source filenames/paths and conversion parameters like profiles/task name etc. There is no information about the recording metadata.
PreMetaCustomCommandPath=<Full Path/Relative of the executable>
PreMetaCustomCommandParameters=<Optional parameters to be passed - see below for list>
PreMetaCustomCommandHangPeriod=<0 or +ve number>
PreMetaCustomCommandCritical=<true or false>
PreMetaCustomCommandUISession=<true or false>
PreMetaCustomCommandShowWindow=<true or false>
PreMetaCustomCommandExitCodeCheck=<true or false>

2. (v2.3.13+) At the beginning of the conversion process just after extracting the metadata from the source file (before the remuxing or commercial detection or conversion). This custom command can executed with the following parameters. To do so **the first 4 parameters need to be defined** for the profile for which a pre conversion custom command needs to be run.
PreCustomCommandPath=<Full Path/Relative of the executable>
PreCustomCommandParameters=<Optional parameters to be passed - see below for list>
PreCustomCommandHangPeriod=<0 or +ve number>
PreCustomCommandCritical=<true or false>
PreCustomCommandUISession=<true or false>
PreCustomCommandShowWindow=<true or false>
PreCustomCommandExitCodeCheck=<true or false>

3. (v2.3.15+) After remuxing, closed caption extraction and commercial detection is complete, just before cutting the commercials. This custom command can executed with the following parameters. To do so **the first 4 parameters need to be defined** for the profile for which a pre conversion custom command needs to be run.
PreCommercialRemovalCustomCommandPath=<Full Path of the executable>
PreCommercialRemovalCustomCommandParameters=<Optional parameters to be passed - see below for list>
PreCommercialRemovalCustomCommandHangPeriod=<0 or +ve number>
PreCommercialRemovalCustomCommandCritical=<true or false>
PreCommercialRemovalCustomCommandUISession=<true or false>

PreCommercialRemovalCustomCommandShowWindow=<true or false>
PreCommercialRemovalCustomCommandExitCodeCheck=<true or false>

4. (v.2.1.8+) At the end of the conversion process just after the file is renamed and before it is moved to the destination directory. To do so **the first 4 parameters need to be defined** for the profile for which a post conversion custom command needs to be run.
CustomCommandPath=<Full Path/Relative of the executable>
CustomCommandParameters=<Optional parameters to be passed - see below for list>
CustomCommandHangPeriod=<0 or +ve number>
CustomCommandCritical=<true or false>
CustomCommandUISession=<true or false>
CustomCommandShowWindow=<true or false>
CustomCommandExitCodeCheck=<true or false>

5. (v.2.3.14+) At the very end of the conversion process after all the files are moved to the destination directory. To do so **the first 4 parameters need to be defined** for the profile for which a post conversion custom command needs to be run.
PostCustomCommandPath=<Full Path/Relative of the executable>
PostCustomCommandParameters=<Optional parameters to be passed - see below for list>
PostCustomCommandHangPeriod=<0 or +ve number>
PostCustomCommandCritical=<true or false>
PostCustomCommandUISession=<true or false>
PostCustomCommandShowWindow=<true or false>
PostCustomCommandExitCodeCheck=<true or false>

In the *PreMetaCustomCommandParameters, PreCustomCommandParameters, CustomCommandParameters* and *PostCustomCommandParameters* you can use the following parameters:

- *%convertedfile%* - will be replaced with the full name and path of the (converted file for *PreCustom*) (converted and renamed file for *Custom*) (moved/destination converted file for *PostCustom*) as part of the parameters passed to the custom program.
- *%sourcefile%* - will be replaced with the name and path of the original source video.
- *%remuxfile%* - will be replaced with the name and path of the intermediary .TS file generated in the temp working directory if the source video is a WTV or DVR-MS file. Please note this will be blank if the source video is not a WTV or DVR-MS file. The following data is extracted from the Source Video metadata when available
- *%srtfile%* - Full path to SRT file, if it exists, otherwise blank (v2.3.12+)
- *%edlfile%* - Full path to EDL file, if it exists, otherwise blank (v2.3.12+)
- *%taskname%* - Name of the task being used (v2.3.13+)
- *%profile%* - Name of profile being used (v2.3.13+)
- *%workingpath%* - Full path to the temp folder where the converted file is kept (v2.3.12+)
- *%destinationpath%* - Full path of the destination folder where the converted file is kept (v2.3.15+)
- *%originalfilepath%* - Full path of the source file

- *%originalfilename%* - Name of the source file (without the path or extension)
- *%showname%* - Title of the show
- *%episodename%* - Subtitle of the show
- *%episodedescription%* - Description of the show
- *%season%*## - Season no
- *%episode%*## - Episode no
- *%bannerfile%* - Full path to downloaded banner file
- *%bannerurl%* - URL to banner file
- *%imdbmovieid%* - IMDB.com Movie Id
- *%movieid%* - MovieDB.com Movie Id
- *%seriesid%* - TVDB.com Series Id
- *%genre%* - Genre
- *%ismovie%* - True or False if the video is a movie
- *%issport%* - True or False if the video is a sport show (v2.3.15+)
- *%premiereyear%* - Premiere Air Year (v2.3.14+)
- *%premieremonth% - Premiere* Air Month (v2.3.14+)
- *%premieremonthshort% - Premiere* Air Month Name Abbreviation (v2.3.14+)
- *%premieremonthlong% - Premiere* Air Month Full Name (v2.3.14+)
- *%premiereday% - Premiere* Air Day (v2.3.14+)
- *%premieredayshort% - Premiere* Air Day of Week Abbreviation (v2.3.14+)
- *%premieredaylong% - Premiere* Air Day of Week Full Name (v2.3.14+)
- *%airyear%* - Year of air date
- *%airmonth%* - Month of air date
- *%airmonthshort%* - Month of air date abbreviation (v2.3.11+)
- *%airmonthlong%* - Month of air date full name (v2.3.11+)
- *%airday%* - Day of air date (v2.3.6+)
- *%airdayshort%* - Day of air date abbreviation (v2.3.11+)
- *%airdaylong%* - Day of air date full name (v2.3.11+)
- *%airhour%* - Hour of air date (v2.3.6+)
- *%airhourampm%* - Hour of air date in AM/PM (v2.3.12+)
- *%airminute%* - Minute of air date (v2.3.6+)
- *%recordyear%* - Year of record date
- *%recordmonth%* - Month of record date
- *%recordmonthshort%* - Month of record date abbreviation (v2.3.11+)
- *%recordmonthlong%* - Month of record date full name (v2.3.11+)
- *%recordday%* - Day of record date (v2.3.11+)
- *%recorddayshort%* - Day of week from record date abbreviation (v2.3.11+)
- *%recorddaylong%* - Day of week from record date full name (v2.3.11+)
- *%recordhour%* - Hour of record date (v2.3.12+)
- *%recordhourampm%* - Hour of record date in AM/PM (v2.3.12+)
- *%recordminute%* - Minute of record date
- *%network%* - Network channel on the show was recorded
- *%rating%* - Parental rating (v2.3.14+)
- *%airingdbid%* - sageDbId from the SageTV Airing tag (v2.4.1+)
- *%mediafiledbid%* - sageDbId from the SageTV MediaFile tag (v.2.4.1+)

PreCustomCommandParameters CustomCommandParameters are CASE SENSITIVE.
e.g. *CustomCommandPath = C:\Test 1\test.exe*
e.g. *CustomCommandParameters = /i "%convertedfile%" /o "%sourcefile%" -t*
In the above example, *%convertedfile%* will be replaced by the full path and filename of the converted file. The *%sourcefile%* will be replaced with the full path and name of the original source video.
If your CustomCommandXXX line starts with a double quote and ends with a double quote, then you MUST enclose the ENTIRE line in an additional double quote otherwise MCEBuddy will not be able to read it.

e.g. if you want *CustomCommandParameters = "%convertedfile%" "D:\Media\TV Shows"*
then you must write it as:

CustomCommandParameters = ""%convertedfile%" "D:\Media\TV Shows""
(v2.1.4 - v2.3.12) MCEBuddy encloses the parameters in DOUBLE QUOTES.

(v2.3.12+) MCEBuddy **does NOT** enclose the parameters in DOUBLE QUOTES. Some applications required the parameters to be in DOUBLE QUOTES, so YOU need to ADD the DOUBLE QUOTES around the parameters in the as shown in the example above.
(v2.3.15+) MCEBuddy can accept relative paths (relative to the directory MCEBuddy is installed in) for the*CustomCommandPath*, prior to which you must be full path paths only.
The number of #'s at the end for Episode and Season indicates the number of digits in the written number. E.g. *%season%##* will give the output at *01* while *%season%###* will give the output at *001*. #'s are optional, leaving them out will write the number without any formatting. i.e. *%season%* will give *1* as the output.
NOTE: While running a custom command, the output is redirected to the log file with a debug log status. Also note that not all parameters will have valid values for *PreMetaCustomCommandParameters and PreCustomCommandParameters* since this command is run at the beginning of the conversion process (e.g. *%convertedfile%*, *%remuxfile%* will return blank since they don't exist yet or *%airday%* will return blank for *PreMetaCustomCommandParameters* since the data has not been extracted yet).
CustomCommandHangPeriod represents the number of SECONDS MCEBuddy will wait during which if no output is detected and logged to the log file and it determines the process to be hung and kills it. Set it to *0* to DISABLE hang detection. Please note that if hang detection is disabled, MCEBuddy will wait endlessly for the process to finish before proceeding (or the task is deleted/stopped).
CustomCommandCritical, when this is set to *true*, MCEBuddy will fail the entire conversion if the custom command is invalid or if the process is terminated (due to hang detection). If it is set to *false* then MCEBuddy will continue with the conversion processing irrespective of the custom command failure/success UNLESS the converted file has been renamed or deleted. By default the value is *false*.
CustomCommandUISession (v2.3.15+) is set to true if you want your program to run in UI space (Session 1) instead of Kernel/Service space (Session 0) (this does NOT create a UI interface, just runs the program in UI space). This is useful ONLY if you're running a program that accesses hardware enabling API's (such as QuickSync, CUDA, DirectX etc) and since MCEBuddy runs as a service in kernel space and some operating systems do not allow direct access to hardware API's from a kernel space. Use this parameter ONLY WHEN REQUIRED. It has other implications, like, Unicode filenames will not work and MCEBuddy mapped network shares are not available (since this is a UI session and MCEBuddy maps the shares in Service session) etc.

CustomCommandShowWindow (v2.3.15+) is set to true if you want MCEBuddy to show the window for the program it is running, false to hide the window.
CustomCommandExitCodeCheck (v2.4.1+) is set to true if you want MCEBuddy to check the Exit Code of the custom command application. If the return code is 0 (default) it is considered a success, if not 0 (+ve or -ve) then it is considered a failure. If the Exit Code fails MCEBuddy will stop further processing and fail the conversion.
NOTE: MCEBuddy will fail the conversion process if the custom command deletes or renames the converted file (*%convertedfile%*). Also note, Custom Command will fail if it cannot find the file specified (avoid using executables files on network drives since MCEBuddy uses the Windows Service account to access network drives and sometimes these are not mapped correctly due to credential issues)

Advanced parameters to configure MCEBuddy

(apply to mcebuddy.conf)

NOTE: As of v2.3.13 all configurable parameters (except FFmpegBackupRemux) in MCEBuddy.conf can be accessed through the GUI, *Settings -> Expert Settings* page. You do NOT need to edit the MCEBuddy.conf file.HOW TO MAKE CHANGES TO MCEBUDDY.CONF
Start the MCEBuddy application from the Start Menu. **Click on *Stop* to stop the MCEBuddy engine. This is VERY important**, if you don't press *Stop*, the changes you make to mcebuddy.conf will be overwritten. Open mcebuddy.conf in Notepad, make the required changes and save the file. Now click *Start* on the MCEBuddy application.
DO NOT REPLACE the file mcebuddy.conf directly with an old version, it will break MCEBuddy. Open, edit and save it. The old version is preserved for your reference only.

Custom eMail Notification Subject Lines (v2.3.13+)

The following parameters can be edited to create custom subject lines for eMail notifications for events. If there is nothing specified then MCEBuddy will use the default subject lines.

eMailSuccessSubject=<Custom subject line for a successful conversion>
eMailFailedSubject=<Custom Subject Line for a failed conversion>
eMailCancelledSubject=<Custom Subject Line for a cancelled conversion>
eMailStartSubject=<Custom Subject Line on the start of a conversion>
eMailDownloadFailedSubject=<Custom Subject Line when downloading series information fails>
Each custom subject line can contain User Custom Commands that will be replaced by MCEBuddy at runtime. Refer to the *Running Custom Commands* for a liist of custom commands that can be used in the subject line.
e.g. *eMailSuccessSubject=Show %showname% for file %originalfilename% conversion successful using task %taskname%*

NOTE: Not always will all the data be available in the custom commands, for e.g. when download information has failed or on a start event the video metadata like episode name and converted file name may not be available.

CustomComskipPath=xyz (v2.1.12+)
This parameter can be set if you want a use a custom version of Comskip, e.g. a Donator version or special/old version, that is lying in a different directory than the one that ships with MCEBuddy.

e.g. *CustomComskipPath=D:\Donator MCEBuddy\comskip.exe*
NOTE: The path must be to the executable comskip AND the directory should contain ALL the files necessary for Comskip to run (comskip.ini, comskip.dictionary etc). When using this parameter the default comskip.ini that ships with MCEBuddy is NOT used rather it expects the comskip.ini to be present in the same directory as the custom comskip.exe.

If a special comskip.ini has been specified in the *Conversion Task* advanced settings page then that special comskip.ini that WILL be used rather than the one lying in the custom comskip.exe directory.

IgnoreCopyProtection=false (v2.1.12+)
By default, this value is *false*. If this is set to *true*, then MCEBuddy will ignore the Copy Protection flag on the recording if it encounters it during conversion. It will only record an error in the conversion log and continue the conversion. If the recording is really copy protected then the final video will be look garbled. This flag can help because sometimes videos are incorrectly flagged by recording software/tuners. When this is set to *false*, if MCEBuddy encounters an Copy Protection flag it will stop the conversion after logging an error.

UseRecycleBin=false (v2.1.12+)
By default, this value is *false*. If this is set to *true*, then MCEBuddy will use the Recycle Bin when deleting the original video recording and related files (like EDL, XML etc).

DownloadBanner=true (v2.1.12+)
By default, this value is *true*. If this is set to *false*, then MCEBuddy will not download the Banner file (artwork) for the movie or tv series. This can be used when you want to download the series information (check *Download Series Details* from the *Conversion Task* page) but not download the artwork / banner file.

CPUAffinity=<binary mask integer> (v2.1.12+)
This parameter can be set to set the CPU affinity mask. This limits MCEBuddy to 1 or more processors only as specified in the bitmask. While this number is written as an integer, the integer represent a binary mask with each bit representing a processor. HINT: Convert the integer to a binary representation, each bit in the binary number presents a CPU processor.

E.g.:

CPUAffinity=1 will select the 1st CPU processor

CPUAffinity=2 will select the 2nd CPU processor

CPUAffinity=3 will select the 1st and 2nd CPU processor

CPUAffinity=4 will select the 3rd CPU processor

CPUAffinity=5 will select the 1st and 3rd CPU processor

CPUAffinity=6 will select the 2nd and 3rd CPU processor

CPUAffinity=7 will select the 1st, 2nd and 3rd CPU processor

TempWorkingPath=xyz (v2.1.2+)

This parameter is used to manually specific the temp working directory for MCEBuddy, i.e. xyz. By default MCEBuddy uses a directory called *working* within the MCEBuddy installation directory (e.g. C:\Program Files\MCEBuddy2x\working) (it will make multiple directories if using multiple simultaneous conversions with a number at the end for each parallel conversion, e.g. working0, working1 etc). This has to be a complete working path with rights to be able to create, modify and delete files/directories else the conversion might fail. This parameter should come under the section *[Engine]* in the conf file.

E.g. *TempWorkingPath=D:\Temp\MCEBuddy*

PollPeriod=xxx (v2.1.2+)

This parameter specifies how often MCEBuddy should check the *Monitor Locations* for new files. xxx is defined in *number of seconds*. This parameter should come under the section *[Engine]* in the conf file.

E.g. *PollPeriod=150*

NOTE: By default MCEBuddy uses 300 seconds as the interval to check for new files.

LocalServerPort=xxx (v2.2.16+)

This parameter specifies the TCP/IP port of the installed MCEBuddy server engine on the local machine. MCEBuddy registers this port with the any UPnP enabled routers on the network by default to enable remote connections from outside the network (NAT). This is the MCEBuddy GUI application should connect to for the local machine. The default port used by MCEBuddy is *23332*. This parameter should come under the section *[Engine]* in the conf file.

E.g. *ServerPort=23332*

SubtitleSegmentOffset=x.x (v2.3.12+)

This parameter is used when the subtitles go progressively out of sync after commercials are removed from the video. This happens because when a video is cut, the video is aligned with the next GOP and I Frame sequence which can be a few seconds away from the actual cutting point. Hence each time the video is cut the subtitles get progressively out of sync with the video. This setting is used to bring the subtitles back into sync with the video. Each time the video commercial is cut out, the subtitles are time shifted by this amount (+ve or -ve in seconds), hence bringing the subtitles back into sync with the video segments.

E.g. *SubtitleSegmentOffset=5.4*
HangPeriod=xxx (v2.2.4+)
This parameter specifies how long MCEBuddy will wait for a console output from the application (e.g. handbrake or ffmpeg etc) before it determines that the application has hung and terminates it. xxx is defined in *number of seconds*. This parameter should come under the section *[Engine]* in the conf file. Use*0* to disable hang detection (i.e. MCEBuddy will wait forever for the process to exist normally).
E.g. *HangPeriod=500*
NOTE: By default MCEBuddy uses 300 seconds as the interval to determine a hung application

FFMpegBackupRemux (v2.1.12+)
This is a special section in the configuration file that stores the command parameters for the ReMux operations which are carried out using FFMPEG if the main ReMuxSupp application fails/not used to remux the video. First the *CopyRemux* is used, if that fails then the *SlowRemux* is used. *CopyRemux* is used first for MPEG2 video files to stream copy the video without recoding the video. If the original video is not MPEG2 or *CopyRemux* fails then *SlowRemux* parameters are used to remux the video. THESE SHOULD NOT BE CHANGED UNLESS THE BACKUP REMUX IS COMPLETELY FAILING. Each of these entries will have a number after them like *CopyRemux0*, *CopyRemux1*, *SlowRemux0*, *SlowRemux1* etc. These numbers indicate successive remux parameters to be tried if the previous one fails, i.e. if*SlowRemux0* fails then MCEBuddy will look for *SlowRemux1* and if that fails then it looks for *SlowRemux2*etc.
You can also specify *-i <source>* in the remux parameters as a placeholder for the input filename which MCEBuddy will replace at runtime. This can be useful if you want to specify parameters before the input file in the FFMPEG parameter command.
E.g. CopyRemux0 = -fflags +genpts -i <source> -map 0:a -map 0:v -vcodec copy -acodec copy

MCEBuddy automatically detects the frame rate of the video through the use of *-r auto* in the FFMPEG Remux parameters in *CopyRemux* and *SlowRemux*. This can be disabled by removing *-r auto*. It can also be overridden by manually specifying the frame rate to use (which can be useful if the average rate of dropped and/or duplicate frames is very high leading to stuttering videos).
E.g. -r auto
E.g. -r 25
E.g. -r 30*1000/1001

MCEBuddy monitors the *Average Rate of Dropped* frames and *Average Rate of Duplicate* frames, the key is average rate and not absolute value. If this exceeds a threshold then the auto detected frame rate is

incorrect and needs to be corrected. These thresholds can be specified in the parameters*RemuxDropThreshold* and *RemuxDuplicateThreshold*. If that threshold is exceeded you may need to manually specify the frame rate (see above).
E.g. RemuxDropThreshold=3.0
E.g. RemuxDuplicateThreshold=3.0

[FFMpegBackupRemux]
CopyRemux0=<FFMPEG Remux parameters>
SlowRemux0=<FFMPEG Remux parameters>
SlowRemux1=<FFMPEG Remux parameters>
RemuxDropThreshold=<Average Rate of Dropped Frames threshold>
RemuxDuplicateThreshold=<Average Rate of Duplicate Frames threshold>

DEVERLOPERS GUIDE

This page is meant for developers who want to check out the repository for building MCEBuddy on their own or contributing to the development of MCEBuddy.

The code is hosted on a Team Explorer repository.

MCEBuddy requires *Visual Studio 2010* to build. It supports both 32bit and 64bit builds. When using Visual Studio we recommend you use*Team Foundation Server/Team Explorer* as the Source Control system for checking out/in the code. Though*AnkhSVN* is also supported for SVN access.

Installing the Build and Source Control Environment

1. Install Visual Studio 2010
2. Install AnkhSVN (if you're using SVN)
3. Open Visual Studio
4. Click *Tools* -> *Options* -> *Source Control* -> *Plug-in Selection* and choose *Visual Team Foundation Server (*or *AnkhSVN)* as the source control.

Team Explorer Access

1. Open Visual Studio
2. Click Team -> Connect to Team Foundation Server
3. Click Servers
4. Click Add
5. Server Name: *tfs.codeplex.com*
6. Path: *tfs*
7. Port number: *443*
8. Protocol: *https*
9. Project Collection: *TFS21*
10. Project Name: *mcebuddy2x*

11. Username: *<get your username from CodePlex>*
12. Password: *<same as your CodePlex website password>*
13. Click Ok -> Select MCEBuddy2x

AnkhSVN Access

If you're only checking out (Read Only) the repository follow these steps.

1. Open Visual Studio
2. Click File -> Subversion -> Open from subversion
3. Click on top right *Add Url Repository* (globe with + sign)
4. Enter *https://mcebuddy2x.svn.codeplex.com/svn* and click OK
5. Select MCEBuddy.sln and click open
6. Enter the directory you want to check out the code into

For Read/Write access (checkout/checkin) use the following steps (you will need developer access to MCEBuddy CodePlex repository):

1. Download plink.exe from http://the.earth.li/~sgtatham/putty/latest/x86/plink.exe
2. Place the plink.exe in your windows installation directory (c:\windows)
3. Open the following file using Notepad *%APPDATA%\Subversion\config*
4. Add or Replace the following line in the *[tunnels]* section (replace login and pwd with your CodePlex login and password)*ssh = c:\\windows\\plink.exe -batch -l <LOGIN> -pw <PWD>* (**note** the double backslash)
5. Save and close the file
6. Open Visual Studio
7. Click File -> Subversion -> Open from subversion
8. Click on top right *Add Url Repository* (globe with + sign)
9. Enter *https://mcebuddy2x.svn.codeplex.com/svn* and click OK
10. Select MCEBuddy.sln and click open
11. Enter the directory you want to check out the code into
12. Your username is your CodePlex login username
13. Your password is your CodePlex password

After checking out the code, please MAKE SURE YOU READ the ReadMe Developers.txt file to understand how to build and maintain MCEBuddy

CONVERSION PROCESS

A quick and dirty view of the high level conversion process

(as of 2.3.15)

- Prior to conversion process start, the queue manager does a check using the configured filters for
  - Monitor taskname matching to linked conversion tasks
  - Metadata matching (without downloading additional from internet)
    - Filename match
    - Showanme match
    - Network/Channel match
    - Showtype match
- **PreMetaCustomCommand**
- Extract metadata (including reading XML, TiVO etc) and download from the internet
- Check filters for skipping file reprocessing
- **PreCustomCommand**
- Check disk space
- Save source SRT, EDL, EDLP files
- Remux to MPEGTS or Copy source file to temp (if not remuxing)
- Trim video
- Scan with comskip/showanalyzer (if EDL/EDLP file is not saved earlier)
- Extract (if SRT file is not saved earlier)
- Analyze video info/details
- **PreCommercialRemovalCustomCommand**
- Remove advertisements (pre-conversion if configured)
  - Trim closed captions (if EDL exists)
- Reanalyze video info/details and get cropping details
- Convert the file (fallback to various encoders if configured)
-
  - Fix audio delay
  - Remux to required format
- Remove the commercials if there weren't removed earlier
  - Trim closed captions (if EDL exists)
- Add subtitles and chapters to the file
- Write metadata to file

- Write XML file metadata
- Rename the file
- **CustomCommand**
- Move the final file to destination
- Add the final file to the iTunes/WMP library
- Move the remaining files (if found) to the destination (XML, SRT, EDL, EDLP, NFO, LOG)
- **PostCustomCommand**
- Queue manager will clean up and move/delete the original files
- DONE

NOTE: The items that are indented only take place if the Rename and Move files only (i.e. no conversion/processing) is NOT checked.

LATEST VERSION PAGE

This page is used to communicate the latest version (between the BEGIN and END lines)

It is also used to communicate any critical announcements. Make sure it is only 1 line between the announcement tags (the : signifies the start and end of the tag, the text should come between the :'s). With each announcement, one Link can be associated. (the annoucement and link should be put between the :'s on each line respectively). The link must be hyperlinked (i.e. must have a html link code associated with it).

FORMAT for communication latest version
Latest Version:<platform x86 or x64>:<version string major.minor.build>:<release or beta>:EOV

BEGIN VERSION COMMUNICATION
Latest Version:x86:2.3.13:Release:EOV
Latest Version:x64:2.3.13:Release:EOV
END VERSION COMMUNICATION

BEGIN ANNOUNCEMENT
Announcement:Donate to support MCEBuddy:Announcement
Link:https://www.paypal.com/cgi-bin/webscr?cmd=_xclick&business=mcebuddy2x%40yahoo%2ecom&lc=US&item_name=Help%20MCEBuddy&no_shipping=1&tax=0:Link
DonationLink:https://www.paypal.com/cgi-bin/webscr?cmd=_xclick&business=mcebuddy2x%40yahoo%2ecom&lc=US&item_name=Help%20MCEBuddy&no_shipping=1&tax=0:DonationLink
FacebookLink:https://www.facebook.com/plugins/like.php?href=http://www.facebook.com/MCEBuddy:FacebookLink
END ANNOUNCEMENT

WRITING A REMOTE CLIENT FOR MCEBUDDY

The MCEBuddy Engine uses XML SOAP 1.1 to communicate with the clients. This allows users to write custom Remote Clients using the publicly exposed interfaces and classes.

This guide is a starting point on how to write a remote client for MCEBuddy. It is an evolving document and feedback is openly welcome.

1. Download the source code (refer to Documentation on connecting to CodePlex) for *MCEBuddy2x*
2. Include the namespace *MCEBuddy.Globals* in your project. This namespace contains all the key objects that will be required to configure and connect to the MCEBuddy engine
3. The public interfaces to the MCEBuddy engine is provided by the *ICore* interface
4. Open the file *Status.cs* in the Project *MCEBuddy.GUI*, refer to the method *TryConnect* to get started with how to remotely connect to the engine.