

# Convert to MP3 with new Shell commands and ffmpeg

Today we provided an example to a FileMaker developer. The job is to convert the sound from a video (or audio) file to an mp3 file. As we recently got the new [Shell](#) functions, we can now use ffmpeg easily in FileMaker with a script like this:



```
Set Variable [ $shell ; Value: MBS( "Shell.New" ) ]
Set Field [ Shell::Output ; "" ]
Set Field [ Shell::Error ; "" ]
Commit Records/Requests [ With dialog: Off ]
# Where the app is:
Set Variable [ $executable ; Value: "/Applications/ffmpegX.app/Contents/Resources/ffmpeg" ]
# option: overwrite file
Set Variable [ $s ; Value: MBS( "Shell.AddArgument"; $shell; "-y" ) ]
# option: input file
Set Variable [ $s ; Value: MBS( "Shell.AddArgument"; $shell; "-i" ) ]
# Path to input file
Set Variable [ $InputFile ; Value: "/Users/cs/Desktop/movie.m4v" ]
# and output file
Set Variable [ $OutputFile ; Value: "/Users/cs/Desktop/sound.mp3" ]
# Run it!
Set Variable [ $s ; Value: MBS( "Shell.Execute"; $shell; $executable; $InputFile; $OutputFile ) ]
Set Variable [ $error ; Value: "" ]
Set Variable [ $result ; Value: "" ]
If [ MBS("IsError") ]
    Show Custom Dialog [ "Failed to run" ; $s ]
Else
    # Loop while app runs and collect messages
    Loop
        # Wait a second or till it quits
        Set Variable [ $s ; Value: MBS( "Shell.Wait"; $shell; 1 ) ]
        # And read output
        Set Variable [ $error ; Value: $error & MBS( "Shell.ReadErrorText"; $shell; "UTF-8" ) ]
        Set Variable [ $result ; Value: $result & MBS( "Shell.ReadOutputText"; $shell; "UTF-8" ) ]
        Set Field [ Shell::Error ; MBS( "Text.ReplaceNewline"; $error; 1 ) ]
        Set Field [ Shell::Output ; MBS( "Text.ReplaceNewline"; $result; 1 ) ]
        # exit when done
        Exit Loop If [ MBS( "Shell.IsRunning"; $shell ) ≠ 1 ]
    End Loop
    # We are done
    Commit Records/Requests [ With dialog: Off ]
End If
Set Variable [ $r ; Value: MBS( "Shell.Release"; $shell ) ]
```

As you see we prepare new shell and then you can loop and call Shell.Execute with the next input file. When launch works, we run a loop to wait for result and collect output from both channels, stdout and

stderr. The messages are put in the fields and when the tool is done, we exit. The script is universal and can be used to run any shell tool on Mac, Windows and Linux. Just make sure the paths are correct. The new [Shell](#) functions are coming with 7.5 release in the next weeks. You can try [the beta](#) today.

PS: Instead of wait, you can use also the FileMaker Pause Script Step to pause.