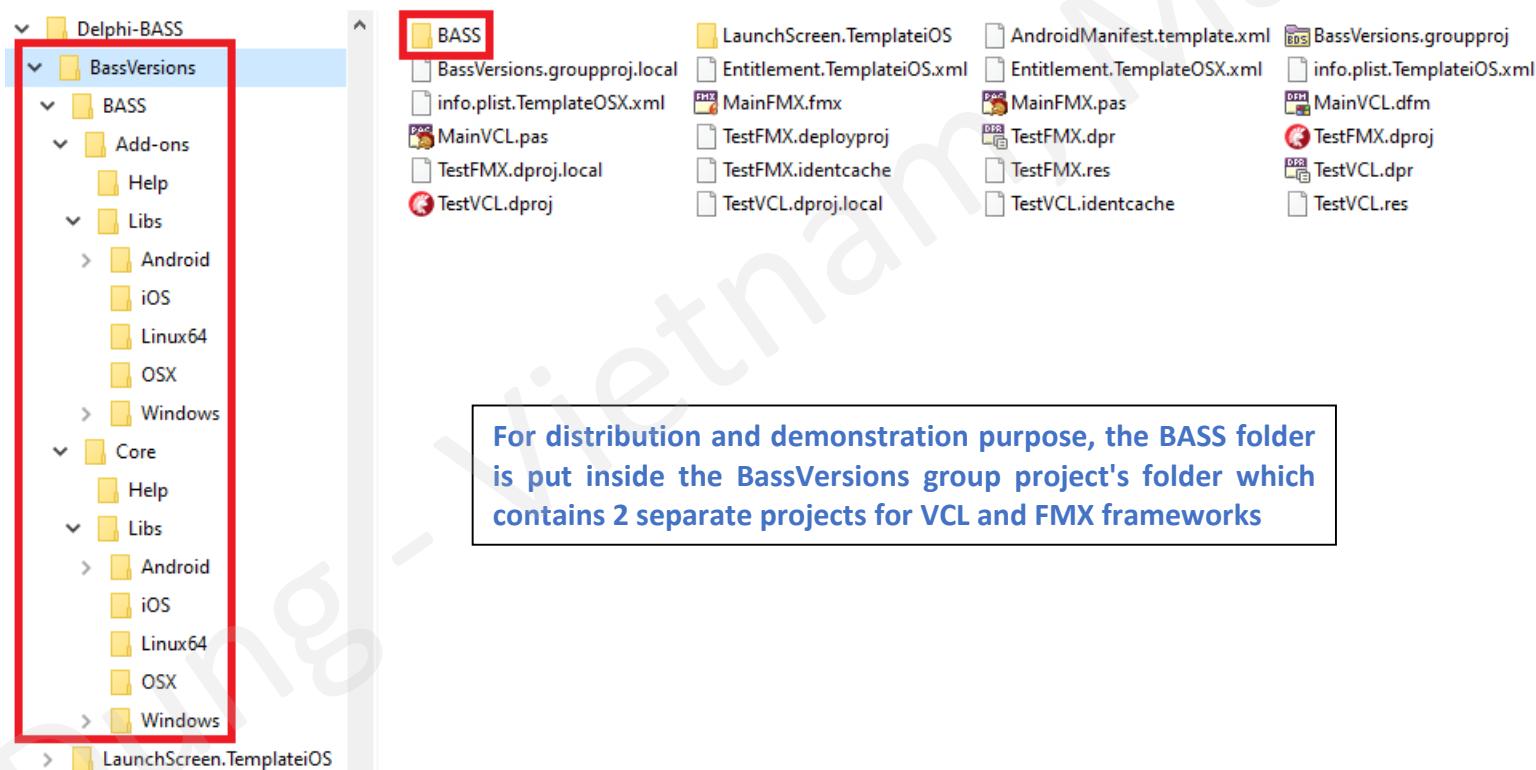


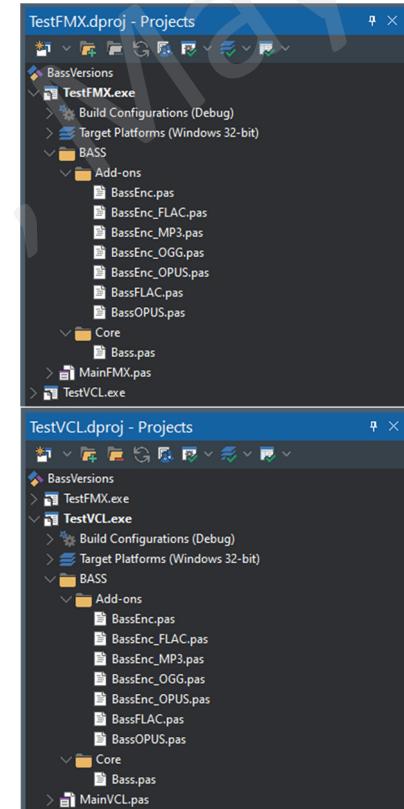
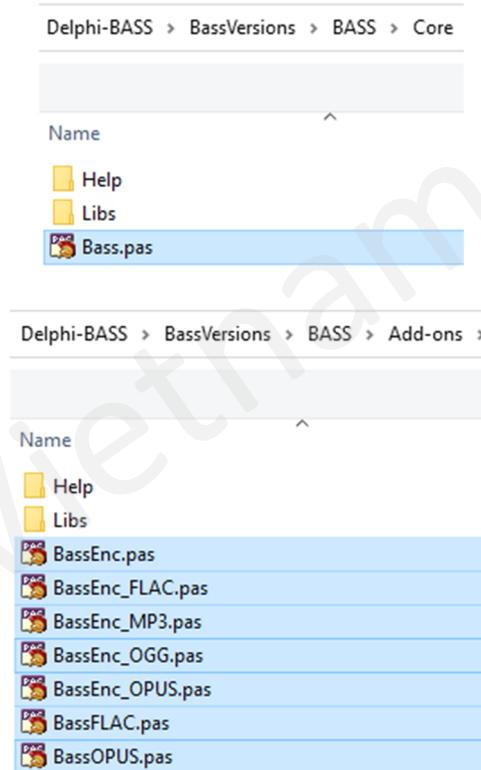
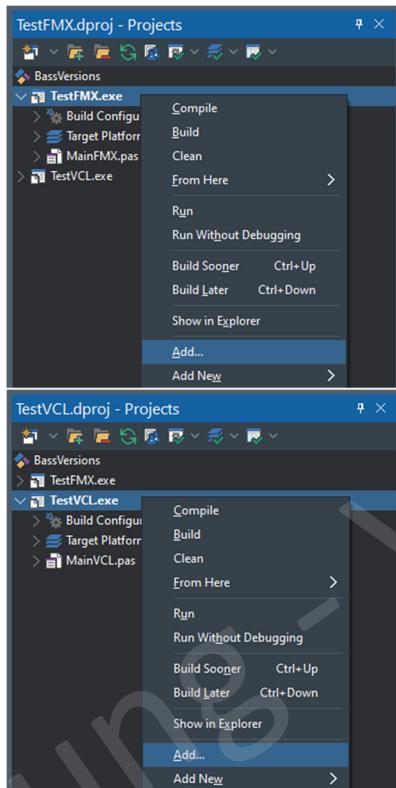
What are necessary files and Where to put them ?

The BASS folder contains all binaries of BASS and add-on audio libraries for all platforms (Windows Intel32/64, MacOS Intel64, Linux Intel64, Android ARM32/64 and iOS ARM64) that Delphi's FMX and VCL frameworks support, and Delphi's wrapper units for all of them:



How to use the Delphi's wrapper units ?

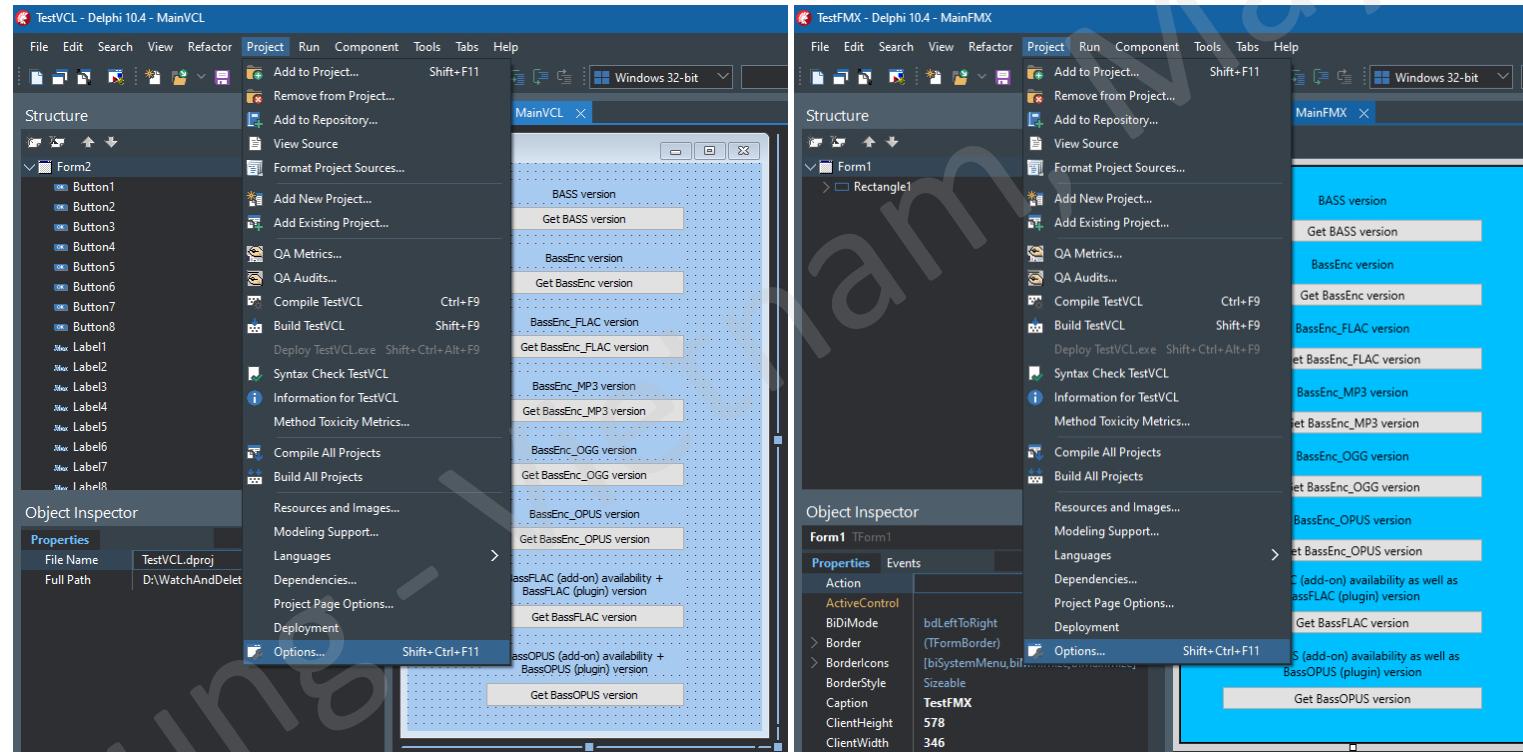
Add **only** desired wrapper unit(s) into a Delphi project as needed:

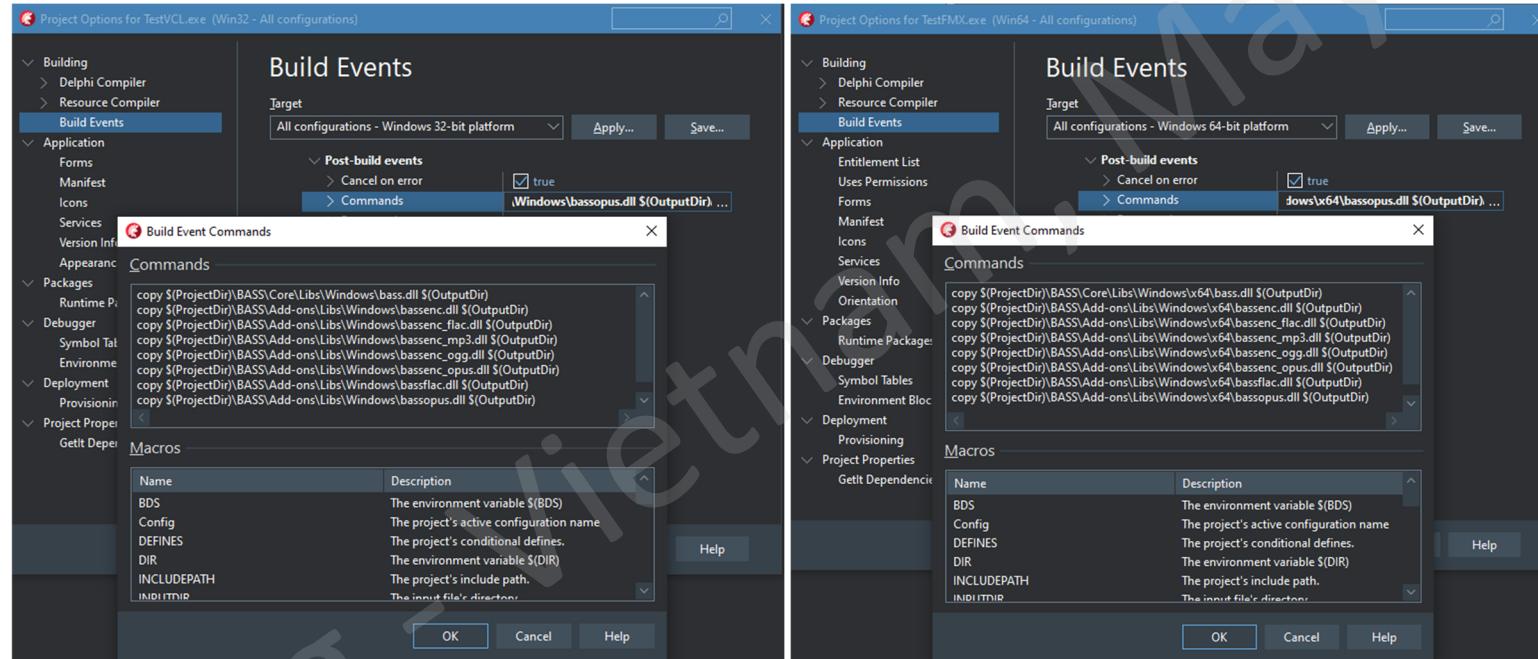


Windows Intel32/64

=====

Use Build Events to deploy **only** necessary DLL(s) for the desired wrapper unit(s) above:

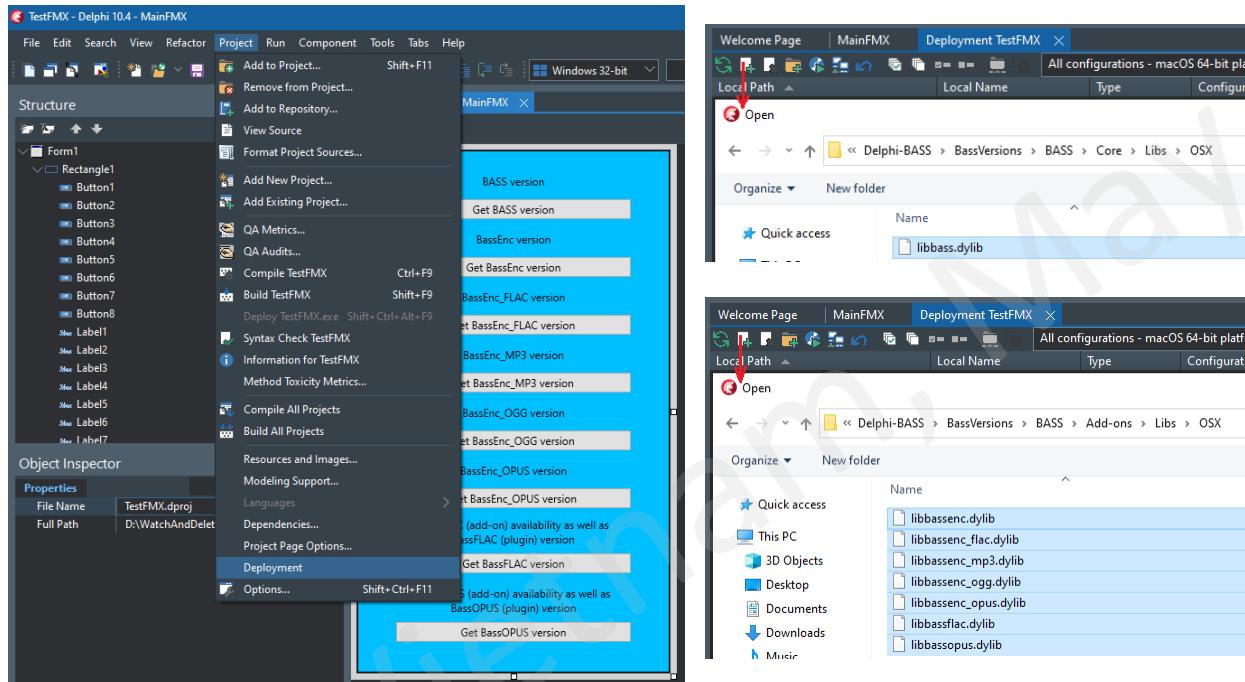




How to use (4 of 12)

MacOS Intel64
=====

Use Deployment to deploy **only** necessary DYLIB(s) for the desired wrapper unit(s) above:

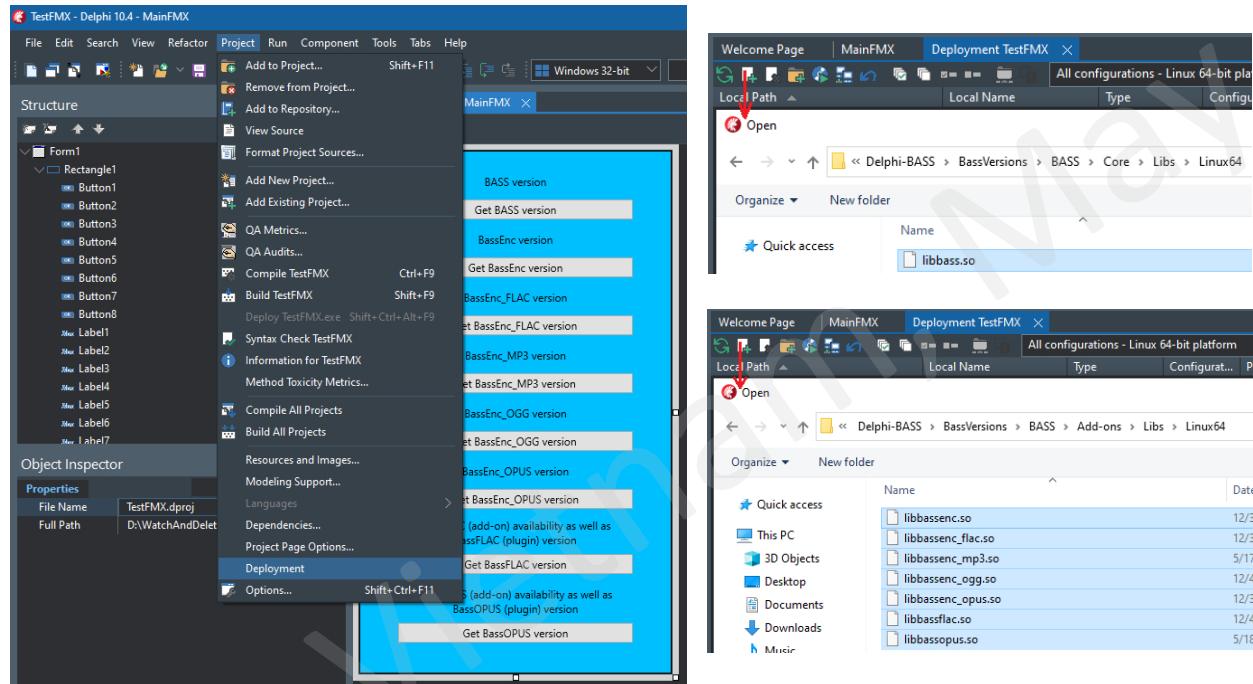


Local Path	Local Name	Type	Configur...	Platforms	Remote Path	Remote Name
\$BDS\bin\	delphi_PROJECTICNS.icns	ProjectOSRes...	Debug	[OSX64]	Contents\Resources\Sta...	TestFMX.icns
\$BDS\bin\	delphi_PROJECTICNS.icns	ProjectOSRes...	Release	[OSX64]	Contents\Resources\	Test.icns
BASS\Add-ons\Libs\OSX\	libbassenc_mp3.dylib	File	Release	[OSX64]	Contents\MacOS\	libbassenc_mp3.dylib
BASS\Add-ons\Libs\OSX\	libbassopus.dylib	File	Release	[OSX64]	Contents\MacOS\	libbassopus.dylib
BASS\Add-ons\Libs\OSX\	libbassflac.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassflac.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_opus.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassenc_opus.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_ogg.dylib	File	Release	[OSX64]	Contents\MacOS\	libbassenc_ogg.dylib
BASS\Add-ons\Libs\OSX\	libbassopus.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassopus.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_ogg.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassenc_ogg.dylib
BASS\Add-ons\Libs\OSX\	libbassenc.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassenc.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_flac.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassenc_flac.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_opus.dylib	File	Release	[OSX64]	Contents\MacOS\	libbassenc_opus.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_flac.dylib	File	Release	[OSX64]	Contents\MacOS\	libbassenc_flac.dylib
BASS\Add-ons\Libs\OSX\	libbassflac.dylib	File	Release	[OSX64]	Contents\MacOS\	libbassflac.dylib
BASS\Add-ons\Libs\OSX\	libbassenc_mp3.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbassenc_mp3.dylib
BASS\Core\Libs\OSX\	libbass.dylib	File	Release	[OSX64]	Contents\MacOS\	libbass.dylib
BASS\Core\Libs\OSX\	libbass.dylib	File	Debug	[OSX64]	Contents\MacOS\	libbass.dylib
Output\Binary\FMX\OSX\Debug\	TestFMX.info.plist	ProjectOSInfo...	Debug	[OSX64]	Contents\	info.plist

How to use (5 of 12)

Linux Intel64

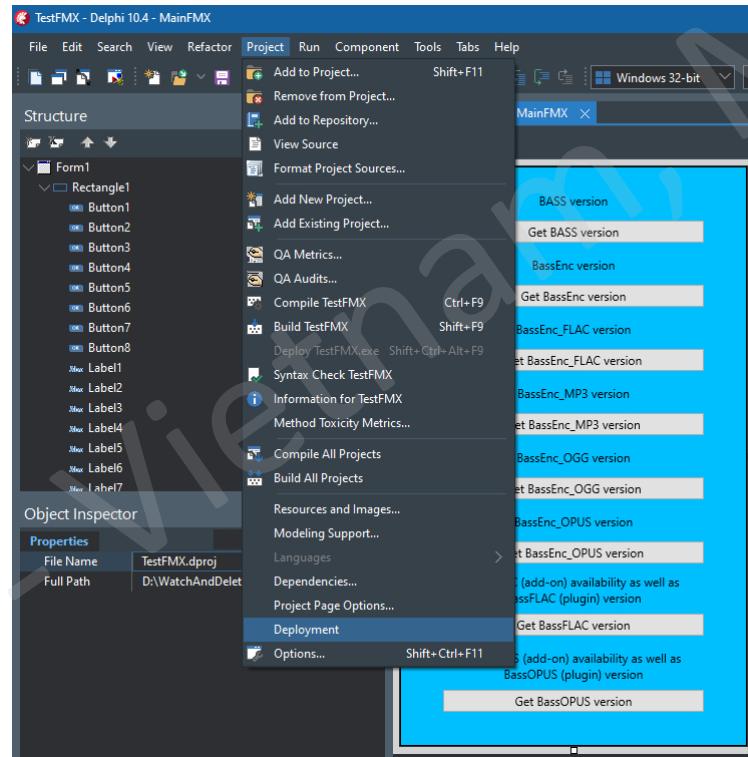
Use Deployment to deploy **only** necessary SO(s) for the desired wrapper unit(s) above:



All configurations - Linux 64-bit platform						
Local Path	Local Name	Type	Configur...	Platforms	Remote Path	Remote Name
\BASS\Add-ons\Libs\Linux64\	libbassenc.so	File	Release	[Linux64]	\	libbassenc.so
\BASS\Add-ons\Libs\Linux64\	libbassenc.so	File	Debug	[Linux64]	\	libbassenc.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_flac.so	File	Debug	[Linux64]	\	libbassenc_flac.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_ogg.so	File	Release	[Linux64]	\	libbassenc_ogg.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_opus.so	File	Release	[Linux64]	\	libbassenc_opus.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_mp3.so	File	Debug	[Linux64]	\	libbassenc_mp3.so
\BASS\Add-ons\Libs\Linux64\	libbassflac.so	File	Debug	[Linux64]	\	libbassflac.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_flac.so	File	Release	[Linux64]	\	libbassenc_flac.so
\BASS\Add-ons\Libs\Linux64\	libbassopus.so	File	Debug	[Linux64]	\	libbassopus.so
\BASS\Add-ons\Libs\Linux64\	libbassflac.so	File	Release	[Linux64]	\	libbassflac.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_opus.so	File	Debug	[Linux64]	\	libbassenc_opus.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_ogg.so	File	Debug	[Linux64]	\	libbassenc_ogg.so
\BASS\Add-ons\Libs\Linux64\	libbassopus.so	File	Release	[Linux64]	\	libbassopus.so
\BASS\Add-ons\Libs\Linux64\	libbassenc_mp3.so	File	Release	[Linux64]	\	libbassenc_mp3.so
\BASS\CoreLibs\Linux64\	libbass.so	File	Debug	[Linux64]	\	libbass.so
\BASS\CoreLibs\Linux64\	libbass.so	File	Release	[Linux64]	\	libbass.so
\Output\Binen\FMX\Linux64\Deb...	TestFMX	ProjectOutput	Debug	[Linux64]		TestFMX

Android ARM32/64

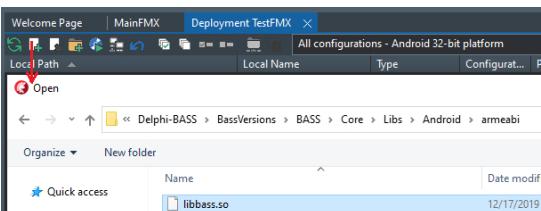
Use Deployment to deploy **only** necessary SO(s) for the desired wrapper unit(s) above:



How to use (7 of 12)

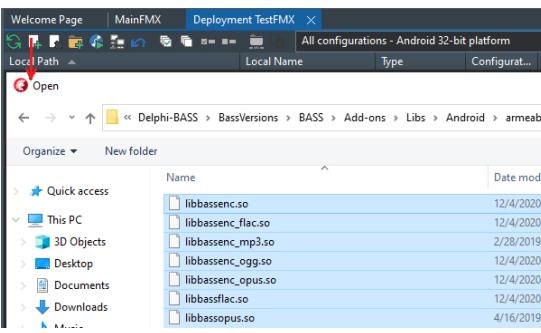
The following screenshots illustrate the steps to copy BASS library files from the local file system to the deployment output directory.

Step 1: Copy libbass.so



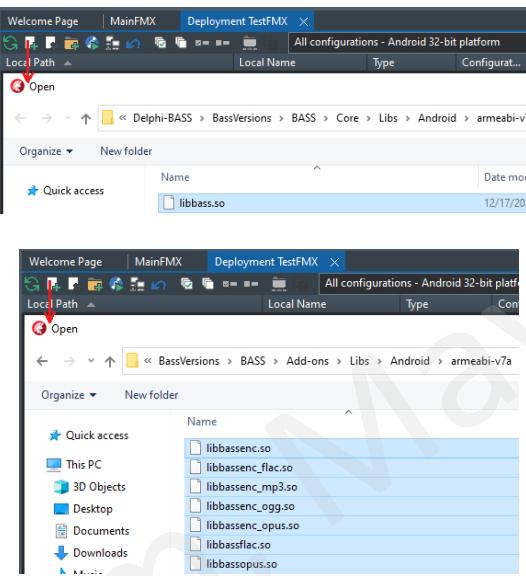
Local Path view showing the file `libbass.so` selected in the `BASS > Core > Libs > Android > armeabi` folder. A red arrow points to the "Open" button in the toolbar.

Step 2: Copy BASS Add-ons Libraries



Local Path view showing multiple BASS add-on libraries selected in the `BASS > Add-ons > Libs > Android > armeabi-v7a` folder. A red arrow points to the "Open" button in the toolbar.

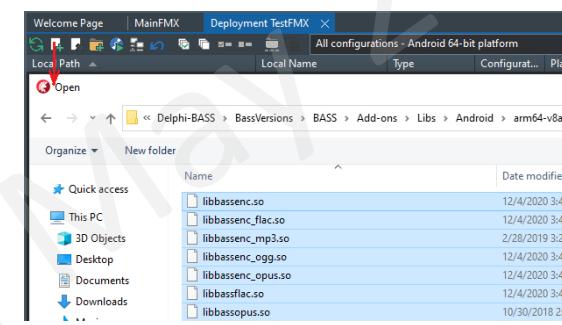
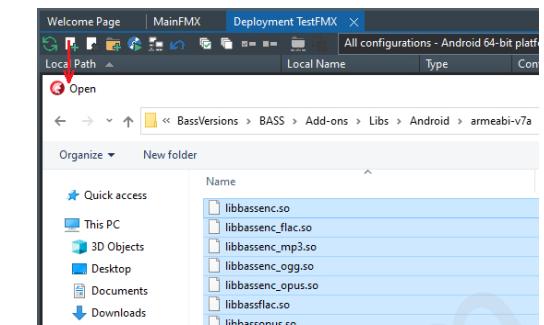
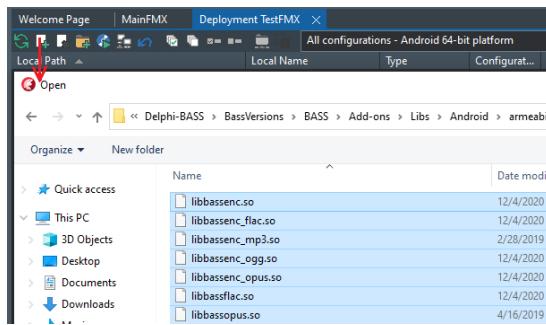
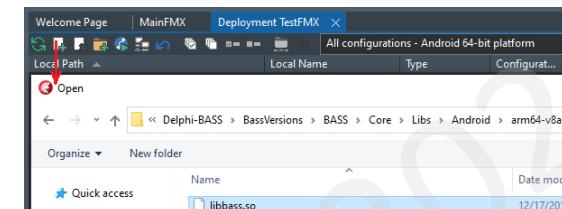
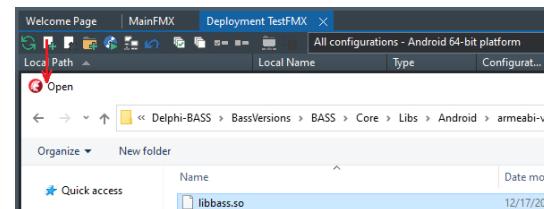
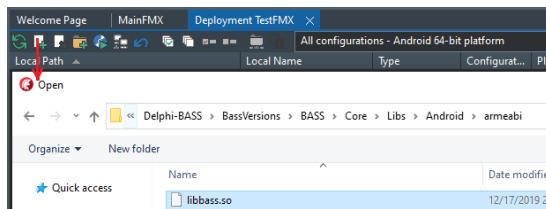
Step 3: Deployment View



Deployment view showing the selected files listed in the grid. The files are grouped by platform: `[Android]`, `[iOS]`, and `[Universal]`. A red box highlights the list of files.

Local Path	Local Name	Type	Configuratio...	Platforms	Remote Path	Remote Name
\$ (BDS)\lib\android\debug\mips\	lnative-activity.so	AndriodLibra...	Release	[Android]	library\lib\mips\	libTest.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_ogg.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_ogg.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassflac.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassflac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_flac.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbassenc_flac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassflac.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbassflac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbassenc.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_flac.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbassenc_flac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_mp3.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbassenc_mp3.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_opus.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_opus.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_mp3.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_mp3.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_opus.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_opus.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_flac.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_flac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_mp3.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_mp3.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_opus.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_opus.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_flac.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_flac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_mp3.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_mp3.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_opus.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_opus.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_flac.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_flac.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_mp3.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_mp3.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_opus.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_opus.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_ogg.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_ogg.so
BASS>Add-ons\Libs\Android\armeabi-v7a\	libbassenc_mp3.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbassenc_mp3.so
BASS\Core\Libs\Android\armeabi-v7a\	libbass.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbass.so
BASS\Core\Libs\Android\armeabi-v7a\	libbass.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbass.so
BASS\Core\Libs\Android\armeabi-v7a\	libbass.so	File	Release	[Android]	library\lib\armeabi-v7a\	libbass.so
BASS\Core\Libs\Android\armeabi-v7a\	libbass.so	File	Debug	[Android]	library\lib\armeabi-v7a\	libbass.so
output\Android\Debug\	classes.dex	AndroidClass...	Debug	[Android]	classes\	classes.dex

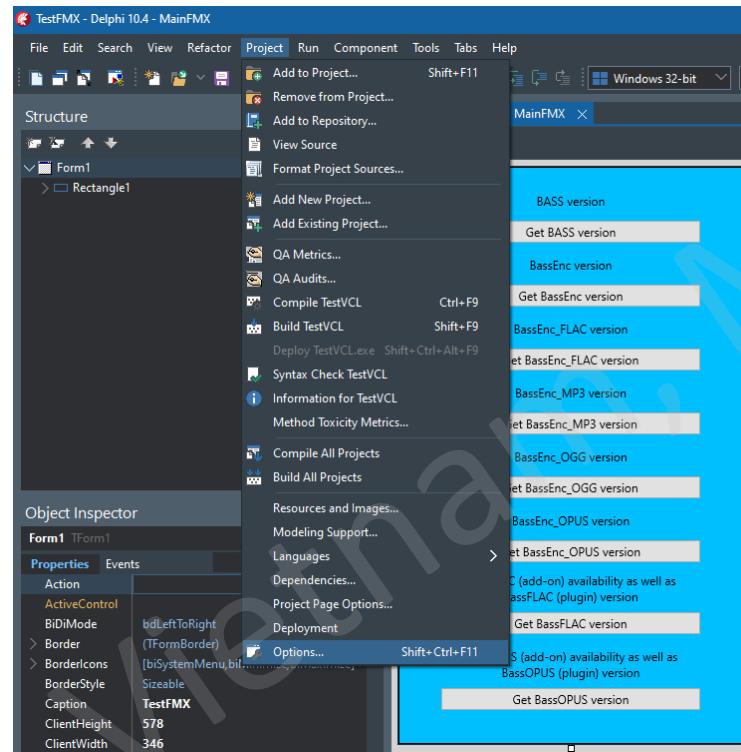
Android



How to use (8 of 12)

Android

Specify necessary permission(s) in Uses Permissions (Example: Record audio):

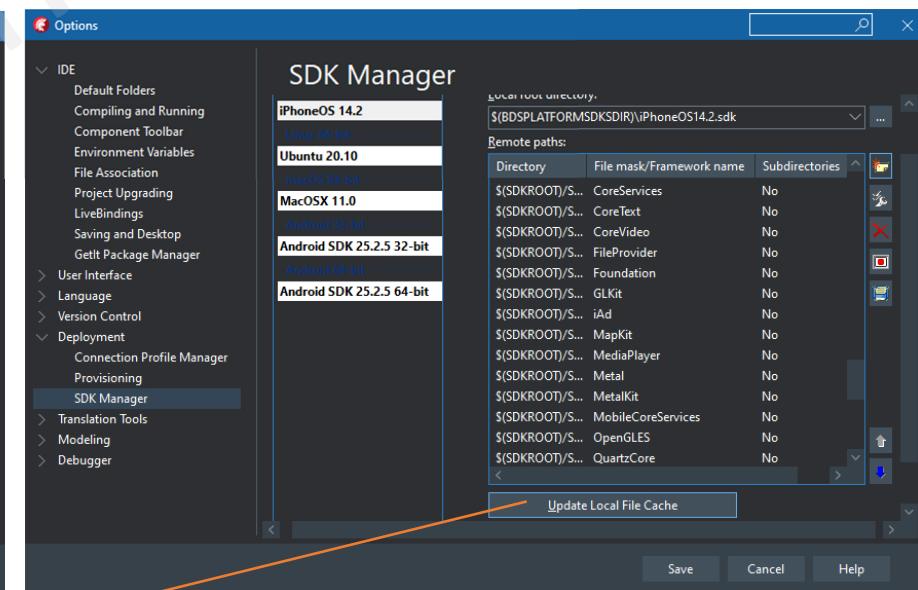
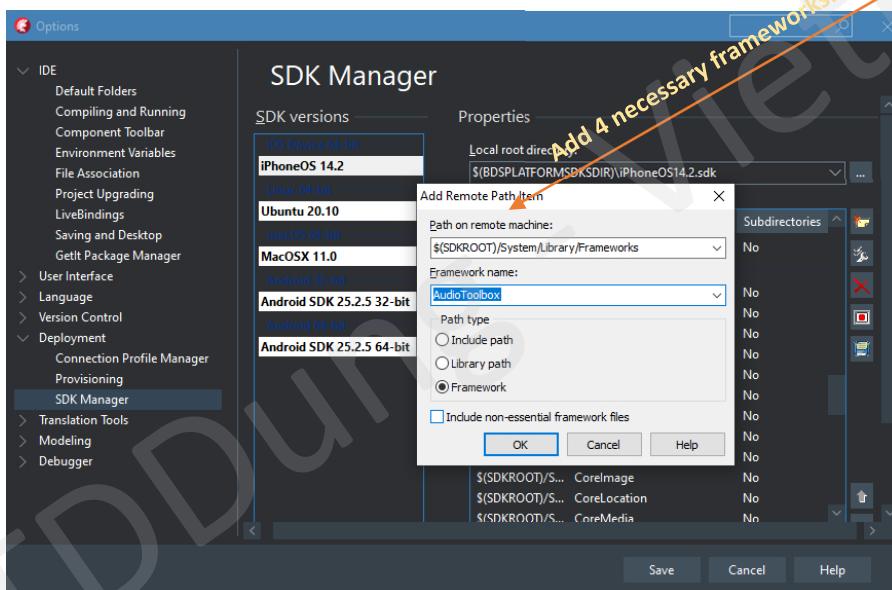
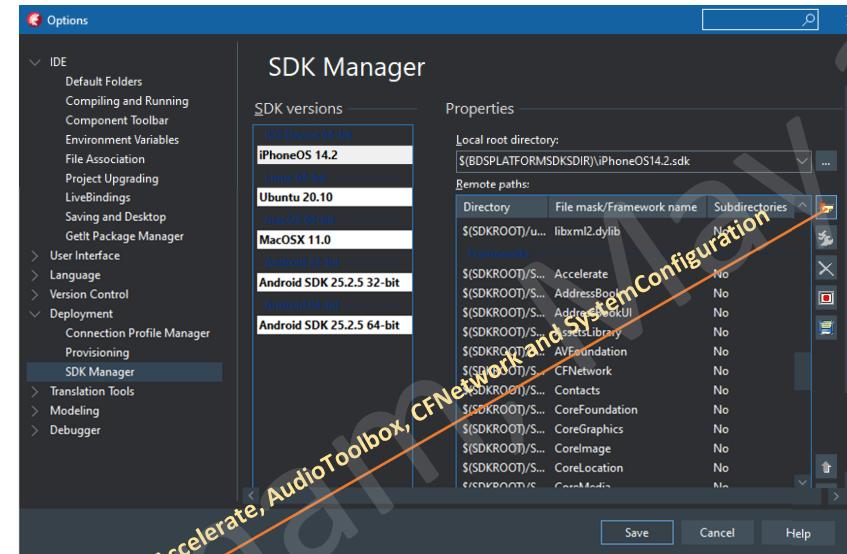
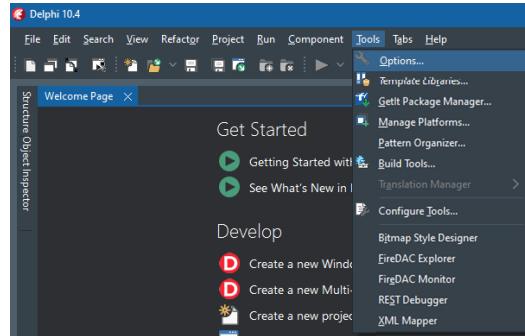


Two side-by-side screenshots of the "Project Options" dialog for the "TestFMX.exe" project, targeting "Android" and "Android64" configurations respectively. Both dialogs show the "Uses Permissions" tab selected. Under the "Target" dropdown, both are set to "All configurations - Android 32-bit platform" and "All configurations - Android 64-bit platform". In the "Entitlement List" section, "Uses Permissions" is selected. The "Record audio" permission is listed with its checkbox checked. Other permissions like "Read user dictionary (obsolete)", "Receive mms", "Receive sms", "Receive WAP push", and "Send SMS" are listed with their checkboxes unchecked.

Request user's approval for the specified permission(s) programmatically in the Delphi's FMX project:

```
if BASS_GetVersion shr 16 <> BASSVERSION then
begin
  TDIALOGService.ShowMessage('An incorrect version of BASS audio library was loaded');
  Exit;
end;
{$IFDEF ANDROID}
if PermissionsService.IsPermissionGranted('android.permission.RECORD_AUDIO') then
  bStart.Enabled:= True
else
  PermissionsService.RequestPermissions(['android.permission.RECORD_AUDIO'],
  procedure(const APermissions: TArray<string>; const AGrantResults: TArray<TPermissionStatus>)
  begin
    if Length(AGrantResults) = 1 then
      case AGrantResults[0] of
        TPermissionStatus.Granted: bStart.Enabled:= True;
        TPermissionStatus.Denied:
          TDIALOGService.ShowMessage('Cannot record audio without the relevant permission being granted');
        TPermissionStatus.PermanentlyDenied:
          TDIALOGService.ShowMessage('If you wish to use the audio recording feature of this app,
          + ' please go to app settings and enable the microphone permission');
      end
    else
      TDIALOGService.ShowMessage('Something went wrong with the permission checking');
    end,
  procedure(const APermissions: TArray<string>; const APostRationaleProc: TProc)
  begin
    // Show an explanation to the user *asynchronously* - don't block this thread waiting for the user's response!
    // After the user sees the explanation, invoke the post-rationale routine to request the permissions
    TDIALOGService.ShowMessage('We first need to be given permission to record audio with your device',
    procedure(const AResult: TModalResult)
    begin
      APostRationaleProc;
    end);
  end);
{$ELSE}
bStart.Enabled:= True;
{$ENDIF}
```

Getting additional frameworks for iOS's SDK:



Get necessary SDK files from XCode for the 4 added frameworks

Request linker to add necessary frameworks in Linking (Example: CFNetwork, AudioToolbox and Accelerate):

