

Play Sound Multiplatform

From Lazarus wiki

[Jump to navigation](#)[Jump to search](#)

Summary

- The object is to play a simple WAV sound both in Windows and Linux (Sync and ASync)
- There are lots of libraries to do this in a complicated way, but this code should suffice for simple use

Download Component

Download from the lazarus CCR [here](#) (playsoundpackage)

Installation of component

- Download all the 'playsoundpackage' package files from the Lazarus CCR repository and copy the whole folder structure to an empty folder off your lazarus components folder
- Install the package (open it, Compile then Install it) and **TPlaysound** will appear on your 'LazControls' components tab
- Drop onto a form, set properties and you are good to go
- PlayCommand property is populated automatically by testing which will work with your system.
- Opening the demo application will give you a good idea how to set the additional properties and use the component

Version

- As reported in the IDE. Initial version is 0.0.2

License

- LGPLv2

Code

- Here is the Uses clause in the Implementation section:

```
..other units..  
FileUtil{$IFDEF  
WINDOWS},mmsystem{$ELSE},asyncprocess,process{$ENDIF},StrUtils  
..other units..
```

- Declare a type:

```
TPlayStyle = (psAsync,psSync);
```

- **Declare some variables:**

```
{IFDEF WINDOWS}  
SoundPlayerAsyncProcess:Tasyncprocess;  
SoundPlayerSyncProcess:Tprocess;  
{ENDIF}  
fPathToSoundFile:String;  
fPlayStyle:TPlayStyle;
```

- **And a worker function:**

```
procedure PlaySound(Const szSoundFilename:String);
```

- **And a couple of constants:**

```
CONST C_UnableToPlay = 'Unable to play ';  
{IFDEF WINDOWS}  
  // Defined in mmsystem  
  SND_SYNC=0;  
  SND_ASYNC=1;  
  SND_NODEFAULT=2;  
{ENDIF}
```

- **Now you are good to go:**

```
fPlayStyle := psASync;  
fPathToSoundFile:='mysound.wav';  
If FileExistsUTF8(fPathToSoundFile) then PlaySound(fPathToSoundFile);
```

- **Here's the PlaySound procedure**

```
procedure PlaySound(const szSoundFilename: string);  
var  
  flags: word;  
  szNonWindowsPlayCommand: string;  
begin  
  szNonWindowsPlayCommand := '';  
{IFDEF WINDOWS}  
  if fPlayStyle = psASync then  
    flags := SND_ASYNC or SND_NODEFAULT  
  else  
    flags := SND_SYNC or SND_NODEFAULT;  
  try  
    sndPlaySound(PChar(szSoundFilename), flags);  
  except  
    On E: Exception do
```

```

        E.CreateFmt(C_UnableToPlay +
        '%s Message:%s', [szSoundFilename, E.Message]);
    end;
{$ELSE}
    // How to play in Linux? Use generic Linux commands
    // Use asyncprocess to play sound as SND_ASYNC
    // Try play
    if (FindDefaultExecutablePath('play') <> '') then
        szNonWindowsPlayCommand := 'play -q';
    // Try aplay
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('aplay') <> '') then
            szNonWindowsPlayCommand := 'aplay -q ';
    // Try paplay
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('paplay') <> '') then
            szNonWindowsPlayCommand := 'paplay';
    // Try mplayer
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('mplayer') <> '') then
            szNonWindowsPlayCommand := 'mplayer -really-quiet ';
    // Try CMus
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('CMus') <> '') then
            szNonWindowsPlayCommand := 'CMus ';
    // Try pacat
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('pacat') <> '') then
            szNonWindowsPlayCommand := 'pacat -p ';
    // Try ffplay
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('ffplay') <> '') then
            szNonWindowsPlayCommand := 'ffplay -autoexit -nodisp -loglevel quiet';
    // Try cvlc
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('cvlc') <> '') then
            szNonWindowsPlayCommand := 'cvlc -q --play-and-exit ';
    // Try canberra-gtk-play
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('canberra-gtk-play') <> '') then
            szNonWindowsPlayCommand := 'canberra-gtk-play -c never -f ';
    // Try Macintosh command?
    if (szNonWindowsPlayCommand = '') then
        if (FindDefaultExecutablePath('afplay') <> '') then

```

```

        szNonWindowsPlayCommand := 'afplay';
// Try mpg321
if (szNonWindowsPlayCommand = '') then
    if (FindDefaultExecutablePath('mpg321') <> '') then
        szNonWindowsPlayCommand := 'mpg321 -q';
// proceed if we managed to find a valid command
if (szNonWindowsPlayCommand <> '') then
begin
    if fPlayStyle = psASync then
    begin
        if SoundPlayerAsyncProcess = nil then
            SoundPlayerAsyncProcess := Tasyncprocess.Create(nil);
            SoundPlayerAsyncProcess.CurrentDirectory :=
ExtractFileDir(szSoundFilename);
            SoundPlayerAsyncProcess.Executable :=
                FindDefaultExecutablePath(Copy2Space(szNonWindowsPlayCommand));
            SoundPlayerAsyncProcess.Parameters.Clear;
            SoundPlayerAsyncProcess.Parameters.Add(szSoundFilename);
            try
                SoundPlayerAsyncProcess.Execute;
            except
                On E: Exception do
                    E.CreateFmt('Playstyle=paASync: ' + C_UnableToPlay +
                        '%s Message:%s', [szSoundFilename, E.Message]);
            end;
        end
    else
    begin
        if SoundPlayerSyncProcess = nil then
            SoundPlayerSyncProcess := Tprocess.Create(nil);
            SoundPlayerSyncProcess.CurrentDirectory :=
ExtractFileDir(szSoundFilename);
            SoundPlayerSyncProcess.Executable :=
                FindDefaultExecutablePath(Copy2Space(szNonWindowsPlayCommand));
            SoundPlayerSyncProcess.Parameters.Clear;
            SoundPlayerSyncProcess.Parameters.Add(szSoundFilename);
            try
                SoundPlayerSyncProcess.Execute;
                SoundPlayerSyncProcess.WaitOnExit;
            except
                On E: Exception do
                    E.CreateFmt('Playstyle=paSync: ' + C_UnableToPlay +
                        '%s Message:%s', [szSoundFilename, E.Message]);
            end;
        end
    end
end

```

```
        end;
    end
else
    raise Exception.CreateFmt('The play command %s does not work on your
system',
        [szNonWindowsPlayCommand]);
{$ENDIF}
end;
```

- Destroy the process objects in your Destroy procedure:

```
{$IFDEF WINDOWS}
FreeAndNil (SoundPlayerSyncProcess);
FreeAndNil (SoundPlayerAsyncProcess);
{$ENDIF}
```

Retrieved from "http://wiki.freepascal.org/index.php?title=Play_Sound_Multiplatform&oldid=144743"

Categories:

- [Audio](#)
- [Components](#)