

Play a song from URL

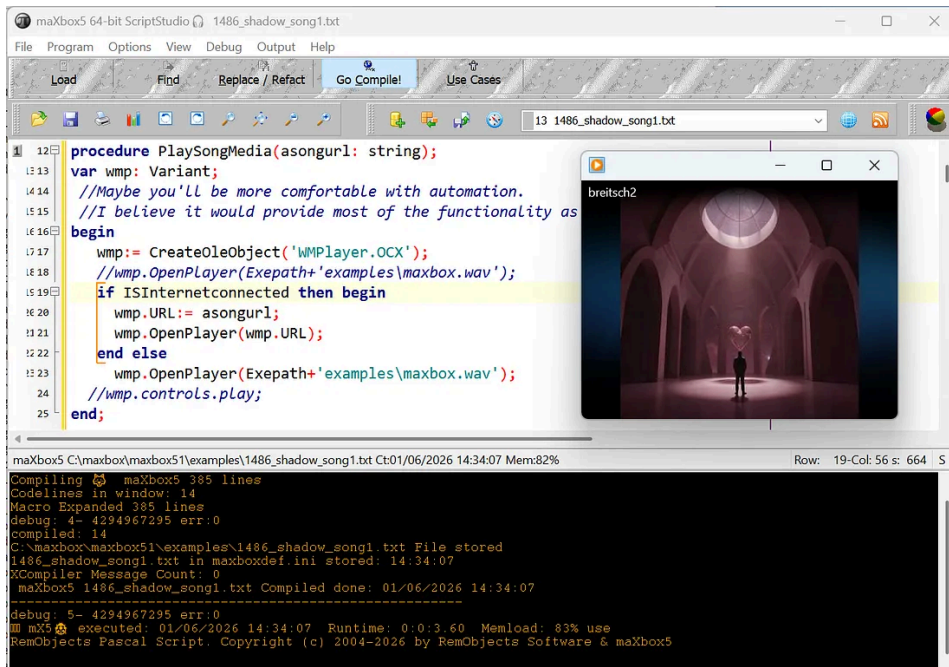
Max Kleiner 4 min read · Just now

2 Comments 0 Shares 0 Likes

maXbox or Python uses OLE/COM automation here, i.e., the dynamic control of an external ActiveX object, instead of implementing its own audio function. This allows the code to play media via URL or local file without having to write a decoder itself.

```
PlaySongMedia(SONG_URL);
```

This procedure is a small wrapper around the Windows Media Player COM object, used to play either an online audio file or a local fallback file.



https://sourceforge.net/projects/maxbox5/files/examples/1486_shadow_song1.txt/download

What each part does

- `procedure PlaySongMedia(asongurl: string);` defines a routine that receives one text parameter: the media URL or file path.

- `var wmp: Variant;` declares a late-bound object variable, which is convenient for automation objects like COM/ActiveX.
- `wmp:= CreateOleObject('WMPPlayer.OCX');` creates an instance of the Windows Media Player control.
- `if ISInternetconnected then ... else ...` chooses between the given song URL and a bundled local WAV file.
- `wmp.URL:= asongurl;` sets the media source to the input string.
- `wmp.OpenPlayer(wmp.URL);` opens and plays that media source through Windows Media Player.

Flow in plain English

1. Create a Windows Media Player object.
2. Check whether the computer is online.
3. If online, use the supplied song address.
4. If offline, use a local example WAV or mp3 file.
5. Tell the player to open and play the selected source.

Important details

The code relies on COM automation, so it works only on Windows where the Media Player control is available. It does not include error handling, so a bad URL, missing file, or player creation failure could raise an exception. The commented line `wmp.controls.play;` would be another way to start playback, but `OpenPlayer` already handles the launch here.

One subtle point and error handling

`Variant` is used instead of a strongly typed interface, which makes the code flexible but also less safe at compile time. In practice, that means maXbox can talk to the player without importing a specific type library.



In maXbox (which uses Delphi-like Object Pascal), you add error handling for missing files in COM code by combining:

1. `FileExists` to check if a local file exists before using it
2. `try...except` to catch COM and runtime exceptions

3. Optional custom error messages/logs in the `except` block

This prevents crashes when files are missing or the COM object fails.

[softwareschule]

. . .

Complete Example: PlaySongMedia with Error Handling

```

textprocedure PlaySongMedia(asongurl: string);
var
  wmp: Variant;
  localFile: string;
  errorMsg: string;
begin
  localFile := Exepath + 'examples\maxbox.wav';
  wmp := Unassigned; // Initialize variable
try
  // 1. Check file existence before using it
  if ISInternetconnected and (asongurl <> '') then begin
    // For online URLs, you could add timeout/retry logic here
    wmp := CreateOleObject('WMPlayer.OCX');
    wmp.URL := asongurl;
  end else begin
    // 2. Fallback: check local file exists
    if not FileExists(localFile) then
      raise Exception.Create('Local file not found: ' + localFile);
    wmp := CreateOleObject('WMPlayer.OCX');
    wmp.URL := localFile;
  end;
  // 3. Start playback
  wmp.OpenPlayer(wmp.URL);
  writeln('Playing: ' + wmp.URL);
except
  on E: Exception do begin
    errorMsg := 'Error playing media: ' + E.Message;
    writeln(errorMsg);
    // Optional: log to file, show message box, or use fallback
  end;
end;
end;

```

```

    wmp := CreateOleObject('WMPlayer.OCX');
    wmp.URL := localFile;
  end;

  // 3. Start playback
  wmp.OpenPlayer(wmp.URL);
  writeln('Playing: ' + wmp.URL);

except
  on E: Exception do begin
    errorMsg := 'Error playing media: ' + E.Message;
    writeln(errorMsg);
    // Optional: log to file, show message box, or use fallback
  end;
end;
end;

```

. . .

Key Points

TechniquePurpose `FileExists(path)` Check if local file exists before passing to COM [[groups.google](#)] `try...except` Catch COM errors, missing files, or runtime exceptions [[softwareschule](#)] `on E: Exception do` Access the error message via `E.Message` [[softwareschule](#)] `raise Exception.Create(...)` Manually trigger error when file is missing [[softwareschule](#)] `Unassigned` Initialize variant to avoid undefined state

Technique	Purpose
<code>FileExists(path)</code>	Check if local file exists before passing to COM [groups.google]
<code>try...except</code>	Catch COM errors, missing files, or runtime exceptions [softwareschule]
<code>on E: Exception do</code>	Access the error message via <code>E.Message</code> [softwareschule]
<code>raise Exception.Create(...)</code>	Manually trigger error when file is missing [softwareschule]
<code>Unassigned</code>	Initialize variant to avoid undefined state

Why This Works

- **Pre-check** with `FileExists` avoids passing invalid paths to COM [[groups.google](#)]
- `try...except` catches COM failures like `WMPPlayer.OCX` not being registered or player creation failing [[stackoverflow](#)]
- **Graceful degradation:** if the online URL fails, you fall back to a local file; if that's missing too, you get a clear error message instead of a crash [[es.scribd](#)]

Alternative: Silent Fallback Without Raising. If you want to silently skip playback instead of raising an exception:

```

textif not FileExists(localFile) then
begin
  writeln('Warning: Fallback file not found, skipping playback');
  Exit; // Exit procedure early
end;

```

This is useful when missing files are expected and not truly errors.

. . .

```
const SONG_URL = 'http://softwareschule.ch/download/shadow.mp3';

procedure PlaySongMedia(asongurl: string);
var wmp: Variant;
  //Maybe you'll be more comfortable with automation.
  //I believe it would provide most of the functionality as interfaces provide.
begin
  wmp:= CreateOleObject('WMPLayer.OCX');
  //wmp.OpenPlayer(Exepath+'examples\maxbox.wav');
  if ISInternetconnected then begin
    wmp.URL:= asongurl;
    wmp.OpenPlayer(wmp.URL);
  end else
    wmp.OpenPlayer(Exepath+'examples\maxbox.wav');
  //wmp.controls.play;
end;

PlaySongMedia(SONG_URL);
```

```
main call: PlaySongMedia(SONG_URL);
```

You can access many Windows Media Player properties and methods via `CreateOleObject('WMPLayer.OCX')` in maXbox using late-bound `Variant` access. The main objects are:

- `wmp` – the player itself
- `wmp.controls` – playback control (play, pause, stop, position, etc.)
- `wmp.settings` – volume, mute, rate, autoStart, etc.
- `wmp.currentMedia` – info about the current song/video
- `wmp.currentPlaylist` – playlist info
- `wmp.URL`, `wmp.playState`, `wmp.uiMode` – basic player properties

All of these work with a `Variant` in maXbox/Delphi/Python.

Music Player

Multimedia

Automation



Written by Max Kleiner

46 followers · 3 following

Max Kleiner's professional environment is in the areas of OOP, UML and coding - among other things as a trainer, developer and consultant.

Edit profile