The Industrious Squirrel

MENU

Microsoft Windows Text-to-Speech for Unity

EDIT 7/15/2019 I'm updating this post with my latest code and to remove the dependency on UniExtensions

I've made a wrapper around the Windows-only Microsoft Speech API for use in Unity. The Microsoft Speech API is a Windows COM capability that first appeared in windows Vista. It is an easy way to get text to speech in a windows application.

This post will go through the steps of making the C++ DLL and the C# behavior for Unity. If you don't care about the "how" and just want to use the plugin skip to the end for the download and usage instructions.

The goal of my wrapper is to startup the text-to-speech engine when the game loads, and provide a function to speak a string of text. I'll need this to run in a thread so that it doesn't disrupt the main Unity thread.

Native C++

The first step is to write the C++ code that will initialize COM, manage a speech queue and call the Speech API.

Declare functions

I'm going to export three functions in the C++ DLL. These functions will be available to the C# code.

```
#ifdef DLL_EXPORTS
#define DLL_API __declspec(dllexport)
#else
#define DLL_API __declspec(dllimport)
#endif

#include &amp;amp;amp;amp;amp;amp;amp;amp;amp;lt;mutex&amp;amp;amp;am
#include &amp;amp;amp;amp;amp;amp;amp;amp;amp;lt;list&amp;amp;amp;amp;a
#include &amp;amp;amp;amp;amp;amp;amp;amp;amp;lt;thread&amp;amp;amp;am

namespace WindowsVoice {
  extern "C" {
    DLL_API void __cdecl initSpeech();
    DLL_API void __cdecl addToSpeechQueue(const char* text);
    DLL_API void __cdecl clearSpeechQueue();
    DLL_API void __cdecl destroySpeech();
    DLL_API void __cdecl statusMessage(char* msg, int msgLen);
  }

  std::mutex theMutex;
  std::list&amp;amp;amp;amp;amp;amp;amp;amp;amp;lt;wchar_t*&amp;amp;amp;a
  std::thread* theSpeechThread = nullptr;
  bool shouldTerminate = false;

  std::wstring theStatusMessage;
}
```

Init

To startup speech, I need to create a thread for speech to run in, initialize COM and create the Microsoft Voice resource. When it is time to shut down, I need to send a signal to the thread to free the Voice resource, then shutdown COM.

```
#include "pch.h"
#include "WindowsVoice.h"
#include &amp;amp;amp;amp;amp;amp;amp;amp;amp;lt;sapi.h&amp;amp;amp;ar

namespace WindowsVoice {

  void speechThreadFunc()
  {
    ISpVoice * pVoice = NULL;

    if (FAILED(::CoInitializeEx(NULL, COINITBASE_MULTITHREADED)))
    {
      theStatusMessage = L"Failed to initialize COM for Voice.";
      return;
    }

    HRESULT hr = CoCreateInstance(CLSID_SpVoice, NULL, CLSCTX_ALL, IID_IS
    if (!SUCCEEDED(hr))
    {
      LPSTR pText = 0;

      ::FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER | FORMAT_MESSA
        NULL, hr, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), pText, 0, N
      LocalFree(pText);
      theStatusMessage = L"Failed to create Voice instance.";
      return;
    }
    theStatusMessage = L"Speech ready.";

    SPVOICESTATUS voiceStatus;
    wchar_t* priorText = nullptr;
    while (!shouldTerminate)
    {
      pVoice-&amp;amp;amp;amp;amp;amp;amp;amp;amp;gt;GetStatus(&amp;
      if (voiceStatus.dwRunningState == SPRS_IS_SPEAKING)
      {
        if (priorText == nullptr)
          theStatusMessage = L"Error: SPRS_IS_SPEAKING but text is NULL";
        else
        {
          theStatusMessage = L"Speaking: ";
          theStatusMessage.append(priorText);
          if (!theSpeechQueue.empty())
          {
            theMutex.lock();
          theMutex.lock();
          if (lstrcmpW(theSpeechQueue.front(), priorText) == 0)
          {
            delete[] theSpeechQueue.front();
            theSpeechQueue.pop_front();
          }
          theMutex.unlock();
        }
      }
    }
    else
    {
      theStatusMessage = L"Waiting.";
      if (priorText != NULL)
      {
        delete[] priorText;
        priorText = NULL;
      }
      if (!theSpeechQueue.empty())
      {
        theMutex.lock();
        priorText = theSpeechQueue.front();
        theSpeechQueue.pop_front();
        theMutex.unlock();
        pVoice-&amp;amp;amp;amp;amp;amp;amp;amp;amp;gt;Speak(priorTe
      }
    }
    Sleep(50);
  }
  pVoice-&amp;amp;amp;amp;amp;amp;amp;amp;amp;gt;Pause();
  pVoice-&amp;amp;amp;amp;amp;amp;amp;amp;amp;gt;Release();

  theStatusMessage = L"Speech thread terminated.";
}

void addToSpeechQueue(const char* text)
{
  if (text)
  {
    int len = strlen(text) + 1;
    wchar_t *wText = new wchar_t[len];

    memset(wText, 0, len);
    ::MultiByteToWideChar(CP_UTF8, NULL, text, -1, wText, len);

    theMutex.lock();
    theSpeechQueue.push_back(wText);
    theMutex.unlock();
  }
}
void clearSpeechQueue()
{
  theMutex.lock();
  theSpeechQueue.clear();
  theMutex.unlock();
}
void initSpeech()
{
  shouldTerminate = false;
  if (theSpeechThread != nullptr)
  {
    theStatusMessage = L"Windows Voice thread already started.";
    return;
  }
  theStatusMessage = L"Starting Windows Voice.";
  theSpeechThread = new std::thread(WindowsVoice::speechThreadFunc);
}
void destroySpeech()
{
  if (theSpeechThread == nullptr)
  {
    theStatusMessage = L"Speach thread already destroyed or not started.";
    return;
  }
  theStatusMessage = L"Destroying speech.";
  shouldTerminate = true;
  theSpeechThread-&amp;amp;amp;amp;amp;amp;amp;amp;amp;gt;join();
  theSpeechQueue.clear();
```

```
    theSpeechQueue.clear();
    delete theSpeechThread;
    theSpeechThread = nullptr;
    CoUninitialize();
    theStatusMessage = L"Speech destroyed.";
  }
  void statusMessage(char* msg, int msgLen)
  {
    size_t count;
    wcstombs_s(&amp;amp;amp;amp;amp;amp;amp;amp;amp;amp;count, msg
  }
}


BOOL APIENTRY DllMain(HMODULE, DWORD ul_reason_for_call, LPVOID)
{
  switch (ul_reason_for_call)
  {
  case DLL_PROCESS_ATTACH:
  case DLL_THREAD_ATTACH:
  case DLL_THREAD_DETACH:
  case DLL_PROCESS_DETACH:
    break;
  }

  return TRUE;
}
```

Since this code will be executing in the main thread and is modifying the shared speech queue, I need to use a mutex to lock the shared queue. The memory that I allocate in this function will need to be freed by the speech thread after the text has been spoken.

Speech

The while loop in the speechThreadFunc monitors the speech queue and calls Microsoft Speech to do the real work. I also don’t want to repeat the same text, because I often tie speech to button presses and don’t want to queue up a bunch of the same text if the user presses the button multiple times.

For details on building a C++ DLL you can download the Visual Studio files at the bottom of this post.

C#

Now that the C++ DLL is ready, I need to wrap that native code with managed C# code. I’m going to make my C# wrapper a MonoBehaviour so that it is easy to add to a scene.

```
using UnityEngine;
using UnityEngine;
using System.Text;
using System.Runtime.InteropServices;

public static Coroutine ExecuteLater(this MonoBehaviour behaviour, flo
{
 return behaviour.StartCoroutine(_realExecute(delay, fn));
}

public class WindowsVoice : MonoBehaviour {
 [DllImport("WindowsVoice")]
 public static extern void initSpeech();
 [DllImport("WindowsVoice")]
 public static extern void destroySpeech();
 [DllImport("WindowsVoice")]
 public static extern void addToSpeechQueue(string s);
 [DllImport("WindowsVoice")]
 public static extern void clearSpeechQueue();
 [DllImport("WindowsVoice")]
 public static extern void statusMessage(StringBuilder str, int length);
 public static WindowsVoice theVoice = null;
 void OnEnable () {
  if (theVoice == null)
  {
   theVoice = this;
   initSpeech();
  }
 }
 public void test()
 {
  speak("Testing");
 }
 public static void speak(string msg, float delay = 0f) {
  if (Timeline.theTimeline.QReprocessingEvents)
   return;

if ( delay == 0f )
   addToSpeechQueue(msg);
  else
   theVoice.ExecuteLater(delay, () =&amp;amp;amp;amp;amp;amp;a
 }
 void OnDestroy()
 {
  if (theVoice == this)
  {
   Debug.Log("Destroying speech");
   destroySpeech();
   Debug.Log("Speech destroyed");
   theVoice = null;
  }
 }
 public static string GetStatusMessage()
 {
  StringBuilder sb = new StringBuilder(40);
  statusMessage(sb, 40);
  return sb.ToString();
 }
}
```

IMPORTANT: *You must build your game for 64 bit Windows. This plugin will not work for any other configuration.*

I’ve named my C++ DLL “WindowsVoice.dll” and I’ve put that DLL and this .cs file in Assets/Plugins.

Using the Plugin

Here is the visual studio project and code needed to build the WindowsVoice.dll. You only need to download this if you want to change how the speech queue works or need a 32 bit DLL. WindowsVoiceProject

Here is the finished plugin with a 64 bit DLL and the WindowsVoice.cs behavior. WindowsVoiceBehavior

Instructions

To use the plugin, unzip the behavior file into <your project>/Assets/Plugins/WindowsVoice.

Create an game object in your scene to hold the behavior. This game object will be marked DontDestroyOnLoad, so you only need to put this object in your first scene. It is OK to include an extra copy of this in a second scene if you want. Speech will only be initialized once.

Use the static WindowsVoice.theVoice.speak(string) function to use the text to speech capability anywhere in your scripts. No need to worry about init or cleanup since that is tied to the life cycle of the game object.

Chad Weisshaar / 2015/07/02 / Programming / C#, unity, WindowsVoice

50 thoughts on “Microsoft Windows Text-to-Speech for Unity”

vladibo

2016/02/27 at 4:01 pm

Thank you for sharing this Chad!

Under what license are you sharing it?

Log in to Reply

Chad Weisshaar 👤

2016/03/09 at 2:18 pm

Free for all use and open source.

Log in to Reply

Symo

2016/03/06 at 4:48 am

This is great ☒ I’ve just got it working in VS2015 where I did the following:

Create a C++ DLL project
Include WindowsVoice.h
Copy over the code from dllmain.cpp
I had to change:

```
::FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
NULL, hr, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), (LPSTR)&pText, 0, NULL);
```

to

```
::FormatMessage(FORMAT_MESSAGE_ALLOCATE_BUFFER |
FORMAT_MESSAGE_FROM_SYSTEM | FORMAT_MESSAGE_IGNORE_INSERTS,
NULL, hr, MAKELANGID(LANG_NEUTRAL, SUBLANG_DEFAULT), (LPWSTR)&pText, 0, NULL);
```

(i.e. the last param is cast to LPWSTR rather than LPSTR)

Log in to Reply

jgray

2016/06/07 at 11:42 pm

This is great – thank you so much. I was trying to use a Unity plugin with this that bit crushes audio – the Bitcrusher plugin. I think, however, the plugin won’t affect the audio signal because the audio is probably not being routed through Unity’s audio, but through the PC’s system audio – is that correct? Would it be feasible to route this voice code’s audio output through Unity so that I can apply an effect to it?

Log in to Reply

Chad Weisshaar 👤

2016/06/08 at 11:12 am

I’m sorry, but I wouldn’t know how to do that.

If you’re willing to dig into the C++ side, you might look into the latest version of the Microsoft speech API where you can set the output of the ISpVoice to a stream:
https://msdn.microsoft.com/en-us/library/Ee125017(v=vs.85).aspx
From there, you might be able to have it speak the text onto that stream, get the stream data, pass that back to C#, then run it through the unity audio system.
It seems like quite a bit of work, and I haven’t done the research to know if you can get the data out of the C++ stream or if you can use programatically generated audio in Unity.
Log in to Reply

LarryDietz

2017/01/08 at 12:37 am

Anyone know how to determine when it is finished speaking?

I am running into a problem where the game is trying to say something while it is already speaking, and the 2nd thing never gets said.

I don’t know enough about C++ to figure out a way to get it to raise an event when it finishes, or to get it to hold up execution while it is speaking.

Any help would be greatly appreciated.
-Larry
Log in to Reply

Chad Weisshaar 👤

2017/01/08 at 8:27 pm

That isn’t the behavior I’d expect. It should keep a buffer of phrases and say them in order no matter how long it takes. The only thing that it wont do is say the same thing twice. That is something that I put in for my own game, and if that is causing you problems, download the C++ source from this post and remove the second part of the if check on line 46.

If the two phrases aren’t the same, then something else is going on. Email me details at weissoft@chadweisshaar.com and I’ll see if I can help.

Regarding your actual question, my code has a thread that blocks during the call to Speak. There is also a way to call it asynchronously and a way to get the current status. You could have c++ code like this:

```
    wasSpeaking = false;
    while (!shouldTerminate)
    {
      pVoice->GetStatus(&voiceStatus, NULL);
      if (voiceStatus.dwRunningState != SPRS_IS_SPEAKING)
      {
        if (!theSpeechQueue.empty())
        {
          theMutex.lock();
          text = theSpeechQueue.front();
          theSpeechQueue.pop_front();
          theMutex.unlock();
          pVoice->Speak(text, SPF_IS_XML | SPF_ASYNC, NULL);
          wasSpeaking = true;
        }
        else if ( wasSpeaking )
        {
          wasSpeaking = false;
          finishedCallback();
        }
      }
      Sleep(50);
    }
```

The C# code will have to pass the callback function. Use https://www.codeproject.com/Tips/318140/How-to-make-a-callback-to-Csharp-from-C-Cplusplus as a guide to setting up callbacks.
Log in to Reply

aduca9999

2017/01/09 at 4:27 pm

Hello Chad. Have you had an issue with voice to text working in 64bit with the recent Windows 10 Anniversary update?
Log in to Reply

Chad Weisshaar

2017/01/09 at 4:58 pm

No, I haven't. I'm currently using Unity 5.5.0f3 on windows 10 with the latest updates and it is still working for me.
Log in to Reply

heena.agg

2018/01/17 at 2:38 am

Hi Chad, This works great. Can u please tell me how to do vice versa … from voice to text. I tried

using System.Speech. via creating SpeechReocgnizer , Grammar. It crashes every time in Unity. Can you please guide me how to do that. Do you have DLL for the same? which can be directly used for converting Voice to text.
Log in to Reply

Chad Weisshaar

2018/01/17 at 9:56 am

I haven't done anything with voice to text myself. It looks like this is part of .Net 4, but probably not in the .net 2.0 that is included when Unity builds a game. You might make sure your project settings are including the full .net 2.0 instead of the subset. If that still doesn't work, you may be able to manually add the DLL (apparently named System.Speech.DLL) to your project. I haven't checked myself, but I saw a comment online that says you need to download the "WinFX Runtime Components" to get this DLL. You might also check out: https://stackoverflow.com/questions/39611728/how-to-add-speech-recognition-to-unity-project/39613264#39613264
Log in to Reply

TomKun

2017/03/01 at 8:52 am

Hello Chad,

I am using your plugin with Windows 7 64 bit, and Unity 5.5.0f3. When I am in game mode in unity, it works perfect. When I use the .exe file, generated by unity, for example on a different device, I cannot hear any voice.

What can be the reason for this?

Thanks in advance!
Log in to Reply

Chad Weisshaar

2017/03/01 at 10:55 am

The plugin only works in 64 bit mode. On the build dialog, change the architecture to x86_64 and it should work fine.
Log in to Reply

TomKun

2017/03/02 at 12:04 am

Now it's working. Thanks for your answer and fast replay!
Log in to Reply

dennisliu

2017/03/02 at 11:14 pm

Hi Chad,

I test build to APK android but I can't hear any sound in the game.
How about pack it to .APK android ? I mean run in mobile android. I have test it but there is no setting architecture in Android Build Unity.

Thanks
Log in to Reply

Chad Weisshaar

2017/03/27 at 9:51 am

This wont work for Android as it relies on Microsoft Window's system level speech capability. There is no way to port this to Android.
Log in to Reply

santhura

2017/03/27 at 5:14 am

Is it possible to set a different language for speaking?
Log in to Reply

Chad Weisshaar

2017/03/27 at 10:08 am

This isn't something that I have played with, but I know that you can change the voice that is used for the text to speech by calling ISpVoice::SetVoice. You can find the voices that are available on your system with SpEnumTokens. There is some example code here:
https://msdn.microsoft.com/en-us/library/ms719807(v=vs.85).aspx

Another thing to look at is using the XML tags that you can pass to the Speak function. If you pass the SPF_IS_XML flag to Speak() you can then include control tags in your text. Like this: hr = pVoice->Speak(L"This sounds normal but the pitch drops half way through", SPF_IS_XML, NULL);
One of the tags you can use is for language. See this page: https://msdn.microsoft.com/en-us/library/ms723638(v=vs.85).aspx

You'll probably need to be a little careful about how you do this, since all windows installations wont have all voices installed. The way I initialize, it just uses the default voice set in the control panel.
Log in to Reply

Pingback: Board game programming tutorial – Assistant to the Lounge-About

Alexander9727

2017/07/19 at 4:01 am

Hi,

Thanks for this I am having a lot of fun making the voice say random things. Just wanted to know how I could clear the que of strings. Right now I am making it say all the words in an array but if I stop it in between and start it from the beginning it starts from where I quit until it finishes and then starts the array again.

Thanks in advance.
Log in to Reply

Chad Weisshaar

2017/07/19 at 9:51 am

That doesn't currently exist, but would be useful.

If you have grabbed the source, you could add a function to the C++ code to clear the speech queue:

```
DLL_API void __cdecl clearSpeechQueue(); // In header
void clearSpeechQueue() // in C++
{
theMutex.lock();
theSpeechQueue.clear();
theMutex.unlock();
}
```

Then on the C sharp side, add this:

```
[DllImport("WindowsVoice")]
public static extern void clearSpeechQueue();
```

I've updated the pre-built DLL to include this function, so if you re-download the DLL from the link in the post, you should get this new function.
Log in to Reply

Alexander9727

2017/07/19 at 8:35 pm

Thanks for updating the dll. I am big noob at programming and I kind of just know a bit of c#. I've downloaded the dll. Thanks a lot.
Log in to Reply

coezo

2017/07/31 at 5:28 pm

Hey, Chad. Thanks for this code. It helps a lot!

I'm using it for Brazilian Portuguese TTS, and I'm having a problem. The special characters we have (accented vowels, for example: á, ó, ã …) are not being read properly. The strings are properly printed to the console in Unity, and I tried to mess with the file encoding, but it's not that. I wonder if it's something on the C++ side, but I'm not very good with that. Maybe you know what is causing it? Thanks in advance!
Log in to Reply

Chad Weisshaar

2017/08/01 at 8:08 pm

I strongly suspect that the problem is how I'm passing the string from C# to C++. I use a const char* when a wchar is probably required to correctly handle special characters.
However, I remember having trouble using a wchar in the interface. That is why there is some code on the C++ side to convert the incoming char* to a wchar*.
I'd recommend doing some reading on passing wchars in a DLL call and then changing the addToSpeechQueue function.
If you get it to work, I'd love to incorporate your fix into the library.
Log in to Reply

coezo

2017/08/01 at 9:07 pm

Thank you for your reply! ☒ I'll look into that and get back to you if I find the solution!
Log in to Reply

coezo

2017/08/04 at 8:05 pm

Hello, Chad.

I solved my problem replacing the line 124 of the dllmain.cpp file with:

```
::MultibyteToWideChar(CP_UTF8, 0, text, -1, wText, len);
```

I wasted some time trying to mess with passing the string as a wchar_t, but eventually I decided to investigate that conversion. Not sure if that will mess with "English text", but I believe UTF8 encoding is pretty much standard now.

Thank you again!
Log in to Reply

Chad Weisshaar

2017/08/07 at 6:37 pm

Thanks for your suggestion. That change works just fine for plain characters and symbols. I'll put the update into the post.
Log in to Reply

Hadi

2018/08/06 at 11:28 pm

I am getting this error
"Assets/Plugins/WindowsVoice/WindowsVoice.cs(5,7): error CS0246: The type or namespace name `UniExtensions' could not be found. Are you missing an assembly reference?"
It could not find UniExensions.
I am using unity 2017.2.0f3 on mac.
And build target is set to Windows
Architecture x86_64 in unity settings
Log in to Reply

Chad Weisshaar

2018/08/07 at 1:05 pm

Sorry about that. You can probably just delete that line. If you get other errors, you can try using my latest version:

```
using UnityEngine;
using System.Collections;
using System.Runtime.InteropServices;
using System.Text;
using UnityEngine.UI;

#if FAKE_WINDOWS_VOICE
```

```
public class WindowsVoice : MonoBehaviour
{
public Text DebugOutput = null;

public static WindowsVoice theVoice = null;
void OnEnable ()
{
if (theVoice == null)
theVoice = this;
}
public static void speak(string msg, float delay = 0f) {
if (delay == 0f)
{
if (theVoice.DebugOutput != null)
theVoice.DebugOutput.text = msg;
else
Debug.Log(“SPEAK: ” + msg);
}
else
theVoice.ExecuteLater(delay, () => speak(msg));
}
}
#else
public class WindowsVoice : MonoBehaviour {
[DllImport(“WindowsVoice”)]
public static extern void initSpeech();
[DllImport(“WindowsVoice”)]
public static extern void destroySpeech();
[DllImport(“WindowsVoice”)]
public static extern void addToSpeechQueue(string s);
[DllImport(“WindowsVoice”)]
public static extern void clearSpeechQueue();
[DllImport(“WindowsVoice”)]
public static extern void statusMessage(StringBuilder str, int length);
public static WindowsVoice theVoice = null;
// Use this for initialization
void OnEnable () {
if (theVoice == null)
{
theVoice = this;
initSpeech();
}
//else
//Destroy(gameObject);
}
public void test()
{
speak(“Testing”);
}
public static void speak(string msg, float delay = 0f) {
if ( delay == 0f )
addToSpeechQueue(msg);
else
theVoice.ExecuteLater(delay, () => speak(msg));
}
void OnDestroy()
{
if (theVoice == this)
{
Debug.Log(“Destroying speech”);
destroySpeech();
Debug.Log(“Speech destroyed”);
theVoice = null;
}
}
public static string GetStatusMessage()
{
StringBuilder sb = new StringBuilder(40);
statusMessage(sb, 40);
return sb.ToString();
}
}
#endif
```

Log in to Reply

jovemcaio

2019/03/12 at 7:41 am

Hi Chad, thanks for your plugin.
Unfortunately, I am having the same error as Hadi:
“Assets/Plugins/WindowsVoice/WindowsVoice.cs(5,7): error CS0246: The type or namespace name `UniExtensions’ could not be found. Are you missing an assembly reference?”

I tried to use your latest code but it didn’t work, it gave me a bunch of errors like
“assets/Plugins/WindowsVoiceDLL/WindowsVoice.cs(32,12): error CS1056: Unexpected character `?′”

“sets/Plugins/WindowsVoiceDLL/WindowsVoice.cs(67,22): error CS1525: Unexpected symbol `speech′”

“assets/Plugins/WindowsVoiceDLL/WindowsVoice.cs(69,18): error CS1525: Unexpected symbol `destroyed'"

I need to use the plugin to read Brazilian Portuguese words so the coezo's comment gave me a lot of hope, but I need to solve these problems, I would appreciate if you could help me. I am using Unity 5.6.6f2 on a Windows Machine. Thanks in advance.
Log in to Reply

kyos1704

2019/03/11 at 8:55 pm

Hello, Chad.
i solved problem.
if windows default language is japanese, windows use SJIS. but unity send string by utf-8.

C# speak function change to:

```
public static void speak(string msg)
Encoding encoding =
System.Text.Encoding.GetEncoding(System.Threading.Thread.CurrentThread.CurrentCulture.TextInfo.ANSICodePage);
var data = encoding.GetBytes(msg);
addToSpeechQueue(data);
}
```

and change dll import :string to byte[]

```
[DllImport("WindowsVoice")]
public static extern void addToSpeechQueue(byte[] s);
```

i dont know how to work on other language.
thanks
Log in to Reply

HasanEmreTonguc

2019/07/01 at 10:04 pm

Thanks for this.
Can we change the default voice? I'm changing windows settings but its not effecting the game.
Log in to Reply

Chad Weisshaar

2019/07/08 at 11:16 am

What you are doing should work according to this page: https://docs.microsoft.com/en-

us/previous-versions/windows/desktop/ms720163%28v%3dvs.85%29 since I am not requesting a particular voice. But this isn't something that I've tried. You might also play with calling ISpVoice::SetVoice. You can find the voices that are available on your system with SpEnumTokens. There is some example code here: https://msdn.microsoft.com/en-us/library/ms719807(v=vs.85).aspx
Log in to Reply

CO_kiruo
2019/07/15 at 1:21 am

Thanks your dll, it is very nice.
thank you very much!
Log in to Reply

CO_kiruo
2019/07/15 at 7:17 am

but i have a problem
theVoice.ExecuteLater(delay, () => speak(msg));
this function "ExecuteLater" is not Identification on my computer.
i want to get a replay from you . thank you very much!!!
Log in to Reply

Chad Weisshaar
2019/07/15 at 9:44 am

I've updated the post to fix this and include other improvements from the comments. You need to add the following code:

```
public static Coroutine ExecuteLater(this MonoBehaviour behaviour, float delay, System.Action fn)
{
return behaviour.StartCoroutine(_realExecute(delay, fn));
}
```

Log in to Reply

CO_kiruo
2019/07/15 at 6:48 pm

thank you Chad very much!!!
Thank you for your open source sharing and answer!！！
Log in to Reply

Pingback: WindowsVoiceProject + Unity3D | BIEN

Pingback: Windows Voice + Unity3D | sewonist.com

sewonist

2019/11/27 at 5:08 am

Hi~ Thank you for sharing!! I post this source on my blog and test to asian language. It works like a charm. I just leave my post also https://bien.ltd/?p=419 Thanks again~!!!
Log in to Reply

mastershift

2020/07/11 at 10:45 am

I know this is kinda old but just in case, thanks for posting it really helped in my project,

but I noticed a problem with using it in my game. How would one go about stopping it in the middle of talking?

Thank you
Log in to Reply

Chad Weisshaar

2020/07/11 at 4:38 pm

Thanks. Stopping the speech is done in the voice API by calling pVoice->Pause(). I call this only when destroySpeech() is called. In the destroySpeech function, I set shouldTerminate to true which is checked in the thread's while loop. When the loop exists I call Pause() then Release().

If you want the ability to pause, you could add another member variable that the thread's while loop checks to pause and then resume later. If you want to cancel something that is being spoken and drop that line entirely you call Pause() then Speak(nullptr, SPF_ASYNC | SPF_PURGEBEFORESPEAK, nullptr)
Log in to Reply

Alex L

2020/08/01 at 5:22 am

Hello, Chad. I know this is old, but I am trying to use this for my project. However, I am getting two errors:
-Timeline doesn't exist

-WindowsVoice doesn’t contain ExecuteLater

I downloaded the finished 64 bits dll and the plugin, and I’m on Unity 2019.4.1f. How would I be able to fix this?
Thanks.
Log in to Reply

Chad Weisshaar

2020/08/06 at 2:56 pm

I’m sorry about the problems you are seeing. I uploaded a newer version and created dependencies that shouldn’t be there. I’ll try to get a new version uploaded soon, but for now, you can simply delete the line that references the ‘Timeline’ and the following line. For the second error, add this code outside the WindowsVoice class:

```
public static Coroutine ExecuteLater(this MonoBehaviour behaviour, float delay, System.Action fn)
{
return behaviour.StartCoroutine(_realExecute(delay, fn));
}
static IEnumerator _realExecute(float delay, System.Action fn)
{
yield return new WaitForSeconds(delay);
fn();
}
```

Log in to Reply

85halfnotes

2020/08/11 at 9:05 am

When you say “outside the WindowsVoice class” does that mean on the top-level or in another class?
Log in to Reply

Chad Weisshaar

2020/08/11 at 10:34 am

Top level
Log in to Reply

85halfnotes

2020/08/10 at 4:00 pm

Hi, I'm trying to call speak using
WindowsVoice.theVoice.speak("test");

but I'm getting
"Member 'WindowsVoice.speak(string, float)' cannot be accessed with an instance reference; qualify it with a type name instead [Assembly-CSharp]"

so then I tried
WindowsVoice.speak("test");
which compiles but I'm not getting any audio.

Any ideas why? I'm very new to Unity and C# so any help would be appreciated, thank you!
Log in to Reply

Chad Weisshaar

2020/08/10 at 5:16 pm

Well, the most common issue that people have is that they are not deploying 64 bit. Double check your build and make sure you picked x64.
Log in to Reply

ighoyota@tlu.ee

2020/08/14 at 7:44 am

Hello Chad,

I am facing similar problem as @Alex L,

However, after removing the lines below
//if (Timeline.theTimeline.QReprocessingEvents)
// return;

And adding the code below to a line before public class WindowsVoice : MonoBehaviour { class}, I still get errors.

```
public static Coroutine ExecuteLater(this MonoBehaviour behaviour, float delay, System.Action fn)
{
return behaviour.StartCoroutine(_realExecute(delay, fn));
}
static IEnumerator _realExecute(float delay, System.Action fn)
{
yield return new WaitForSeconds(delay);
fn();
}
```

Please advise how I can solve this problem,

Thank you.

Log in to Reply

Chad Weisshaar

2020/08/16 at 12:01 pm

Functions have to be in a class. I've updated the zip to contain working code. Feel free to re-download. You can also just put those lines you added into a new class by doing this (That class can live right where you put the functions):

```
public static class Utility
{
public static Coroutine ExecuteLater(this MonoBehaviour behaviour, float delay, System.Action fn)
{
return behaviour.StartCoroutine(_realExecute(delay, fn));
}
static IEnumerator _realExecute(float delay, System.Action fn)
{
yield return new WaitForSeconds(delay);
fn();
}
}
```

Log in to Reply

Leave a Reply

You must be logged in to post a comment.

PREVIOUS

Touch Table 7 Wonders

NEXT

Timeline