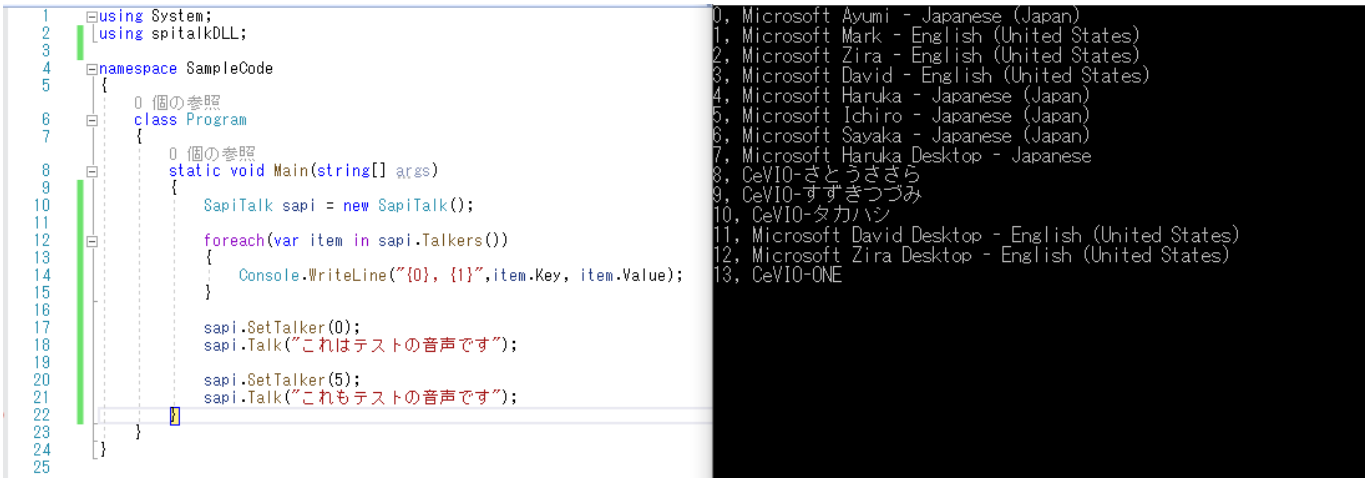
SAPI利用

https://docs.microsoft.com/en-us/previous-versions/windows/desktop/ee125663(v=vs.85)

DLL利用例



```
using System;
using spitalkDLL;

namespace SampleCode
{
    0 個の参照
    class Program
    {
        0 個の参照
        static void Main(string[] args)
        {
            SapiTalk sapi = new SapiTalk();

            foreach(var item in sapi.Talkers())
            {
                Console.WriteLine("{0}, {1}",item.Key, item.Value);
            }

            sapi.SetTalker(0);
            sapi.Talk("これはテストの音声です");

            sapi.SetTalker(5);
            sapi.Talk("これもテストの音声です");
        }
    }
}
```

```
0, Microsoft Ayumi - Japanese (Japan)
1, Microsoft Mark - English (United States)
2, Microsoft Zira - English (United States)
3, Microsoft David - English (United States)
4, Microsoft Haruka - Japanese (Japan)
5, Microsoft Ichiro - Japanese (Japan)
6, Microsoft Sayaka - Japanese (Japan)
7, Microsoft Haruka Desktop - Japanese
8, CeVIO-さとうささら
9, CeVIO-すずきつづみ
10, CeVIO-タカハシ
11, Microsoft David Desktop - English (United States)
12, Microsoft Zira Desktop - English (United States)
13, CeVIO-ONE
```

DLLコード

参照にSpeechLibを追加してください※COM の Microsoft Speech Object Library です。

sapitalk.dll

```
using SpeechLib;
using System;
using System.Collections.Generic;
using System.Threading;

namespace spitalkDLL
{
    public class SapiTalk
    {
        private SpVoice sapi = null;
        private Dictionary<int, SpObjectToken> SpeakerList = new
Dictionary<int, SpObjectToken>();

        private int Speed = 0;
        private int Volume = 100;
        private int AvatorIdx = 0;

        public SapiTalk()
        {
            int idx = 0;
```

```
            try
            {
                sapi = new SpVoice();
                SpObjectTokenCategory sapiCat = new
SpObjectTokenCategory();
                Dictionary<string, SpObjectToken> TokerPool = new
Dictionary<string, SpObjectToken>();

                // See
https://qiita.com/7shi/items/7781516d6746e29c03b4

sapiCat.SetId(@"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Speech_OneCore\Vo
ices", false);

                foreach (SpObjectToken token in
sapiCat.EnumerateTokens())
                {
                    if
(!TokerPool.ContainsKey(token.GetAttribute("name")))
                    {
                        TokerPool.Add(token.GetAttribute("name"),
token);
                    }
                }

                foreach (SpObjectToken token in sapi.GetVoices("", ""))
                {
                    if
(!TokerPool.ContainsKey(token.GetAttribute("name")))
                    {
                        TokerPool.Add(token.GetAttribute("name"),
token);
                    }
                }


                foreach (var item in TokerPool)
                {
                    SpeakerList.Add(idx, item.Value);
                    idx++;
                }
            }
            catch (Exception e)
            {
                Console.WriteLine("{0},{1},{2}", e.Message,
e.InnerException == null ? "" : e.InnerException.Message,
e.StackTrace);
            }
        }
```

```
        public Dictionary<int,string> Talkers()
        {
            Dictionary<int, string> ans = new Dictionary<int,
string>();

            for (int i = 0; i < SpeakerList.Count; i++)
            {
                ans.Add(i, SpeakerList[i].GetDescription());
            }

            return ans;
        }

        public void SetTalker(int talker)
        {
            AvatorIdx = talker;
        }

        public void SetVolume(int volume)
        {
            Volume = volume;
        }

        public void SetRate(int speed)
        {
            Speed = speed;
        }

        public void Talk(string text, bool asyncFlag = false)
        {
            try
            {
                SpObjectToken backupSapi = null;

                Thread t = new Thread(() =>
                {
                    backupSapi = sapi.Voice;
                    sapi.Voice = SpeakerList[AvatorIdx];
                    sapi.Rate = Speed;
                    sapi.Volume = Volume;
                    sapi.Speak(text);
                    sapi.Voice = backupSapi;
                });
                t.SetApartmentState(ApartmentState.STA);
                t.Start();
                if (!asyncFlag) t.Join();
            }
            catch (Exception e)
            {
                throw new Exception(string.Format("発声処理で落ちたっ
す。{0}", e.Message));
```

```
            }
        }

        public void Save(string filePath, string text)
        {
            try
            {
                SpObjectToken backupSapi = null;
                SpFileStream ss = new SpFileStream();
                ss.Open(filePath,
SpeechStreamFileMode.SSFMCreateForWrite);
                sapi.AudioOutputStream = ss;

                Thread t = new Thread(() => {
                    backupSapi = sapi.Voice;
                    sapi.Voice = SpeakerList[AvatorIdx];
                    sapi.Rate = Speed;
                    sapi.Volume = Volume;
                    sapi.Speak(text);
                    sapi.Voice = backupSapi;
                });
                t.SetApartmentState(ApartmentState.STA);
                t.Start();
                t.Join();
                ss.Close();
            }
            catch (Exception e)
            {
                throw new Exception(string.Format("保存処理で落ちたっ
す。{0}", e.Message));
            }
        }

    }
}
```

Windows, Sapi, C#

From:
https://wiki.hgotoh.jp/ - **努力したWiki**

Permanent link:
https://wiki.hgotoh.jp/documents/csharp/code-008

Last update: **2024/11/02 13:39**