

Digital Future Text-to-Speech SDK Programmer's Guide for .NET

Version 3.5.0



The software described in this manual is furnished under a valid license agreement or nondisclosure agreement with Digital Future and/or NeoSpeech Inc and/or Cepstral LLC. The software may be used only in accordance with the terms of the agreement.

Copyright Notice

Copyright © 2008 Digital Future.

Other trademarks and copyrights belong to Voiceware Co., Ltd., Pentax, Cepstral LLC and NeoSpeech, Inc.

All Rights Reserved.

Disclaimer

DIGITAL FUTURE SHALL NOT BE LIABLE FOR TECHNICAL OR EDITORIAL ERRORS OR OMISSIONS CONTAINED HEREIN; NOR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES RESULTING FROM THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL. THE INFORMATION IN THIS MANUAL IS SUBJECT TO CHANGE WITHOUT ANY PRIOR NOTICE FOR THE PURPOSES OF IMPROVING THE PERFORMANCE AND FUNCTIONALITY OF ITS PRODUCTS.

No one is allowed to copy or distribute the whole or a part of this document without prior written permission of Digital Future.

Trademark

VoiceText is the registered trademark of Voiceware Co., Ltd.

Product names mentioned herein may be trademarks and/or registered trademarks of their respective trademark holders.

Chapter 1: Overview

Digital Future Text-to-Speech SDK (DF TTS SDK) is the only **true OS native** (no COM/ActiveX, client-server, etc overheads) cross-platform provider-independent technology that provides standardized and unified API's for the implementation of the conversion of text data into speech.

The currently supported technologies are:

- Voiceware Co., Ltd. VoiceText™ for U.S. English, Chinese, Japanese, Korean
- Cepstral LLC for U.S. English, U.K. English, French, German, Spanish, Italian
- AT&T Natural Voices™ (**for internal use only** and not covered by the license of this SDK. If you are interested in using this technology, you must obtain developer licensing from AT&T directly. Please contact us at sales@digitalfuturesoft.com for details.)

This document contains descriptions and examples of DF TTS SDK API's, necessary for developing text-to-speech enabled software applications in C# and VB.NET.

The Appendix of the present manual includes explanations on the DF TTS SDK Tag Set, which controls inflection information (pitch, sound speed, volume, pause) as

well as the SSML Tag Set, one of the VoiceXML 2.0 standards defined by W3C (World Wide Web Consortium).

Supported platforms: Microsoft ® Windows ™ (all versions), Windows Mobile (2003 and up), (Mac OS X and Linux with C++ and Java).

Supported .NET compilers/ IDE's: Microsoft ® Visual Studio 2005, Microsoft ® Visual Studio 2008.

Chapter 2: Getting Started

DF TTS SDK API's are aimed at facilitating the integration of text-to-speech technology in software applications.

Microsoft ® Windows ™ SDK setup notes:

1. The following DLLs need to be distributed with the application (the best practice is to place them in the application directory):

dftts.dll;
vt_eng.dll;
vt_chi.dll;
vt_jpn.dll;
vt_kor.dll;
swift.dll;

Note: All Dll's are located in the Windows\[C# or VB.NET]\Dll\[Release.. or Debug..] folder.

2. The developer will need to include (refer to .NET samples you received with this SDK):

[C#]
`using System.Runtime.InteropServices;`

[VB.NET]
`Imports System.Runtime.InteropServices`

3. The developer will need to include the .NET SDK code regions to be able to use the APIs (through P/Invoke) (see Appendix 1 and refer to .NET samples you received with this SDK).

Chapter 3: DF TTS SDK Function References

This chapter describes DF TTS SDK Function References, which can be divided into 5 categories.

- Basic API's (loading/unloading the synthesis engine and the User Dictionary, language helper functions)

InitDFTTSEngineEx3()
UninitDFTTSEngine()
MakeLanguage()
GetMainLanguage()
GetSubLanguage()
LoadDFTTSUserDict()
UnloadDFTTSUserDict()

- Sound Card API's (Play/Stop/Pause/Resume of synthesized sound output via sound card)

DFTTSSpeak()
DFTTSStop()
DFTTSPause()
DFTTSResume()

- File API's (Synthesize and save to a voice file)

DFTTSExportToFileEx()

- Buffering API's (Synthesize to a voice buffer)

DFTTSExportToBuffer() (not currently supported by the Desktop SDK)

- Information API's (Get Engine or Voice Information)

GetDFTTSEngineInfo()
GetDFTTSVoice()

InitDFTTSEngineEx3

Loads the synthesizer's TTS database.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern void InitDFTTSEngineEx3(IntPtr hwndWinOwner,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathKate,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathPaul,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathMiyu,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathShow,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathMisaki,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathLily,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathWang,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathJunwoo,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathSujin,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathYumi,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathGyuri,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechDBFolderPathDayoung,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathKate,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathPaul,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathMiyu,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathShow,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathMisaki,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathLily,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathWang,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathJunwoo,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathSujin,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathYumi,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathGyuri,
    [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathDayoung,
    int[] psiLoadedEngines,
    int[] psiLoadedEnginesReturnValues
);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)>
Public Shared Sub InitDFTTSEngineEx3(ByVal hwndWinOwner As IntPtr, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathKate As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathPaul As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathMiyu As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathShow As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathMisaki As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathLily As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathWang As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathJunwoo As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathSujin As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathYumi As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathGyuri As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathDayoung As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathKate As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathPaul As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathMiyu As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathShow As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathMisaki As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathLily As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathWang As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathJunwoo As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathSujin As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathYumi As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathGyuri As System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathDayoung As System.String, _
    ByVal psiLoadedEngines() As Integer, _
    ByVal psiLoadedEnginesReturnValues() As Integer _
)

End Sub
```

Parameters

hwndWinOwner

Use window handle (Win32) used for speech event processing.

12 szNeoSpeechDBFolderPath[VoiceName] parameters

The paths where the NeoSpeech synthesizer databases are located for the supported NeoSpeech voices. Use blank strings for the engine default paths.

12 szNeoSpeechLicFilePath[VoiceName] parameters

NeoSpeech VoiceText™ License verification files for the indicated voices. Use blank strings for the engine default paths.

[IN] psiLoadedEngines

An array with currently used engine types.

[IN] psiLoadedEnginesReturnValues

An array with the load status return values for each engine.

Note: All other vendors' voices are automatically loaded and you do not need to specify DB paths for them.

Description

The function loads the synthesizer database and it is used when the program starts.

Return Values

None

Notes

The return status values are held by 2 arrays.

The array psiLoadedEngines has member values NEOSPEECHVOICETEXT (0) (first 12 members to match with the 12 NeoSpeech voices supported), CEPSTRAL (1) for the whole Cepstral engine (member # 13), ATTNV (2) for the whole ATTNV engine (member # 14 (*internal use only*)), MACOSXSPMAN (4) for the whole MACOSXSPMAN engine (member # 15) and MSSAPI (3) for the whole MSSAPI engine (member # 16).

The first 12 members of psiLoadedEnginesReturnValues

(psiLoadedEnginesReturnValues[0] through [11]) return the load status of each supported NEOSPEECHVOICETEXT voice in the load order (see Parameters section).

The thirteenth member of psiLoadedEnginesReturnValues

(psiLoadedEnginesReturnValues [12]) returns the load status of the whole engine CEPSTRAL.

The fourteenth member (psiLoadedEnginesReturnValues [13]) returns the load status of the whole engine ATTNV.

The fifteenth member (psiLoadedEnginesReturnValues [14]) returns the load status of the whole engine MACOSXSPMAN.

The sixteenth member (psiLoadedEnginesReturnValues [15]) returns the load status of the whole engine MSSAPI.

See the .NET samples for further clarification.

Important Note: MACOSXSPMAN (only valid for Mac OS X) and MSSAPI are not supported by the .NET SDK.

Each member of psiLoadedEnginesReturnValues can hold the following status codes:

When the database is successfully loaded, it returns INIT_DFTTS_ENGINE_SUCCESS. The following **common** values are returned when error occurs:

[INIT_DFTTS_ENGINE_ERROR_DB_PATH_DIFFERENT] Tried to load the synthesizer database with different values of szNeoSpeechDBFolderPath in case of using multiple synthesizer databases

[INIT_DFTTS_ENGINE_ERROR_CHANNEL_MEM_FAIL] Failed to secure channel memory

[INIT_DFTTS_ENGINE_ERROR_DB_MORPHEME_ANALYSIS_FAIL] Failed to load DB for the Morpheme Analysis

[INIT_DFTTS_ENGINE_ERROR_DB_BREAK_INDEX_FAIL] Failed to load DB for the Break Index

[INIT_DFTTS_ENGINE_ERROR_DB_TEXT_PREP_FAIL] Failed to load DB for the Text Pre-Processing

[INIT_DFTTS_ENGINE_ERROR_DB_ACOU_MODEL_FAIL] Failed to load DB for the Acoustic Model

[INIT_DFTTS_ENGINE_ERROR_DB_UNIT_SEL_FAIL] Failed to load DB for Unit Selection

[INIT_DFTTS_ENGINE_ERROR_DB_PROS_MODEL_FAIL] Failed to load DB for Prosody Model

[INIT_DFTTS_ENGINE_ERROR_DB_SPEECH_DB_FAIL] Failed to load DB for Speech Database

[INIT_DFTTS_ENGINE_ERROR_DB_PITCH_LOC_INFO_FAIL] Failed to load DB for Pitch Location Information

[INIT_DFTTS_ENGINE_ERROR_OTHER] Other errors

See enumeration InitDFTTSEngineReturnValue for all error codes.

See Also

UninitDFTTSEngine()

GetDFTTSEngineInfo()

Example (see C# and VB.NET samples)

[C#]

```
hWnd = this.Handle;

int[] iLoadedEngines = new int[NUM_ENGINE_INITIALIZATIONS];

int[] iLoadedEnginesReturnValues = new int[NUM_ENGINE_INITIALIZATIONS];

for (int i = 0; i <= NUM_ENGINE_INITIALIZATIONS - 1; i++)
{
    iLoadedEngines[i] = (int)DFTTSVoiceEngineType.NOENGINE;
```

```

        iLoadedEnginesReturnValues[i] =
(int)InitDFTTSEngineReturnValue.INIT_DFTTS_ENGINE_ERROR_OTHER;

    }

    //(YOU NEED TO CHANGE THE PATHS TO MATCH YOUR PATHS!)

    //This is order that you should supply paths:

    //Kate
    //Paul
    //Miyu
    //Show
    //Misaki
    //Lily
    //Wang
    //Junwoo
    //Sujin
    //Yumi
    //Gyuri
    //Dayoung
    //Kate
    //Paul
    //Miyu
    //Show
    //Misaki
    //Lily
    //Wang
    //Junwoo
    //Sujin
    //Yumi
    //Gyuri
    //Dayoung

    InitDFTTSEngineEx3(hWnd, "C:\\****", "C:\\Program Files\\VW\\VT\\Paul\\M16", "C:\\****",
"C:\\****", "C:\\****", "C:\\****", "C:\\****", "C:\\****", "C:\\****",
"C:\\****", "C:\\****", "C:\\****", "C:\\****", "C:\\Program Files\\VW\\VT\\Paul\\M16\\data-
common\\verify\\verification.txt", "C:\\****", "C:\\****", "C:\\****", "C:\\****", "C:\\****",
"C:\\****", "C:\\****", "C:\\****", "C:\\****", "C:\\****", iLoadedEngines,
iLoadedEnginesReturnValues);

    //Catch load errors

    string sEngineName;
    sEngineName = "";

    InitDFTTSEngineReturnValue result;

    string[] sNeospVoiceNames = new string[NEOSPEECH_NUM_VOICES];

    sNeospVoiceNames[0] = NEOSPEECH_KATE_ENG_NAMESTR;
    sNeospVoiceNames[1] = NEOSPEECH_PAUL_ENG_NAMESTR;
    sNeospVoiceNames[2] = NEOSPEECH_MIYU_JPN_NAMESTR;
    sNeospVoiceNames[3] = NEOSPEECH_SHOW_JPN_NAMESTR;
    sNeospVoiceNames[4] = NEOSPEECH_MISAKI_JPN_NAMESTR;
    sNeospVoiceNames[5] = NEOSPEECH_LILY_CHI_NAMESTR;
    sNeospVoiceNames[6] = NEOSPEECH_WANG_CHI_NAMESTR;
    sNeospVoiceNames[7] = NEOSPEECH_JUNWOO_KOR_NAMESTR;
    sNeospVoiceNames[8] = NEOSPEECH_SUJIN_KOR_NAMESTR;
    sNeospVoiceNames[9] = NEOSPEECH_YUMI_KOR_NAMESTR;
    sNeospVoiceNames[10] = NEOSPEECH_GYURI_KOR_NAMESTR;
    sNeospVoiceNames[11] = NEOSPEECH_DAYOUNG_KOR_NAMESTR;

    for (int intI = 0; intI <= NUM_ENGINE_INITIALIZATIONS - 1; intI++)
    {

        if (iLoadedEngines[intI] == (int)DFTTSVoiceEngineType.NEOSPEECHVOICETEXT)
        {

            sEngineName = "NEOPSEECHVOICETEXT Engine Load Result for voice " +
sNeospVoiceNames[intI] + ": ";
        }

        else if (iLoadedEngines[intI] == (int)DFTTSVoiceEngineType.CEPSTRAL)
        {

```



```

        sEngineName = "CEPSTRAL Engine Load Result: ";
    }

    else if (iLoadedEngines[intI] == (int)DFTTSVoiceEngineType.ATTNV)
    {

        sEngineName = "ATTNV (INTERNAL USE ONLY) Engine Load Result: ";
    }

    else if (iLoadedEngines[intI] == (int)DFTTSVoiceEngineType.MACOSXSPMAN)
    {

        sEngineName = "MACOSXSPMAN (MAC OS X ONLY) Engine Load Result: ";
    }

    else if (iLoadedEngines[intI] == (int)DFTTSVoiceEngineType.MSSAPI)
    {

        sEngineName = "MSSAPI (UNMAMAGED SDK USAGE ONLY) Engine Load Result: ";
    }

    result = (InitDFTTSEngineReturnValue)iLoadedEnginesReturnValues[intI];
    Debug.Write(sEngineName + result + "\n");
}

```

[VB.NET]

```

hWnd = Me.Handle

Dim iLoadedEngines(NUM_ENGINE_INITIALIZATIONS - 1) As Integer
Dim iLoadedEnginesReturnValues(NUM_ENGINE_INITIALIZATIONS - 1) As Integer
For i As Integer = 0 To NUM_ENGINE_INITIALIZATIONS - 1
    iLoadedEngines(i) = CType(DFTTSVoiceEngineType.NOENGINE, Integer)

    iLoadedEnginesReturnValues(i) = _
    CType(InitDFTTSEngineReturnValue.INIT_DFTTS_ENGINE_ERROR_OTHER, Integer)
Next

'(!YOU NEED TO CHANGE THE PATHS TO MATCH YOUR PATHS!)

'This is order that you should supply paths:

'Kate
'Paul
'Miyu
'Show
'Misaki
'Lily
'Wang
'Junwoo
'Sujin
'Yumi
'Gyuri
'Dayoung
'Kate
'Paul
'Miyu
'Show
'Misaki
'Lily
'Wang
'Junwoo
'Sujin

```

```
'Yumi  
'Gyuri  
'Dayoung
```

```
InitDFTTSEngineEx3(hWnd, _  
    "C:\\***", _  
    "C:\\Program Files\\VW\\VT\\Paul\\M16", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\Program Files\\VW\\VT\\Paul\\M16\\data-common\\verify\\verification.txt", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    "C:\\***", _  
    iLoadedEngines, _  
    iLoadedEnginesReturnValues _  
)  
  
'Catch load errors  
  
Dim sEngineName As String  
sEngineName = ""  
  
Dim result As InitDFTTSEngineReturnValue  
  
Dim sNeospVoiceNames(NEOSPEECH_NUM_VOICES - 1) As String  
  
sNeospVoiceNames(0) = NEOSPEECH_KATE_ENG_NAMESTR  
sNeospVoiceNames(1) = NEOSPEECH_PAUL_ENG_NAMESTR  
sNeospVoiceNames(2) = NEOSPEECH_MIYU_JPN_NAMESTR  
sNeospVoiceNames(3) = NEOSPEECH_SHOW_JPN_NAMESTR  
sNeospVoiceNames(4) = NEOSPEECH_MISAKI_JPN_NAMESTR  
sNeospVoiceNames(5) = NEOSPEECH_LILY_CHI_NAMESTR  
sNeospVoiceNames(6) = NEOSPEECH_WANG_CHI_NAMESTR  
sNeospVoiceNames(7) = NEOSPEECH_JUNWOO_KOR_NAMESTR  
sNeospVoiceNames(8) = NEOSPEECH_SUJIN_KOR_NAMESTR  
sNeospVoiceNames(9) = NEOSPEECH_YUMI_KOR_NAMESTR  
sNeospVoiceNames(10) = NEOSPEECH_GYURI_KOR_NAMESTR  
sNeospVoiceNames(11) = NEOSPEECH_DAYOUNG_KOR_NAMESTR  
  
For intI As Integer = 0 To NUM_ENGINE_INITIALIZATIONS - 1  
  
    If iLoadedEngines(intI) = CInt(DFTTSVoiceEngineType.NEOSPEECHVOICETEXT) Then  
  
        sEngineName = "NEOSPEECHVOICETEXT Engine Load Result for voice " & _  
            sNeospVoiceNames(intI) & ": "  
  
    ElseIf iLoadedEngines(intI) = CInt(DFTTSVoiceEngineType.CEPSTRAL) Then  
  
        sEngineName = "CEPSTRAL Engine Load Result: "  
  
    ElseIf iLoadedEngines(intI) = CInt(DFTTSVoiceEngineType.ATTNV) Then  
  
        sEngineName = "ATTNV (INTERNAL USE ONLY) Engine Load Result: "  
  
    ElseIf iLoadedEngines(intI) = CInt(DFTTSVoiceEngineType.MACOSXSPMAN) Then
```

```

        sEngineName = "MACOSXSPMAN (MAC OS X ONLY) Engine Load Result: "

        ElseIf iLoadedEngines(intI) = CInt(DFTTSVoiceEngineType.MSSAPI) Then

            sEngineName = "MSSAPI (UNMAMAGED SDK USAGE ONLY) Engine Load Result: "

        End If

        result = CType(iLoadedEnginesReturnValues(intI), InitDFTTSEngineReturnValue)

        Debug.Write(sEngineName & result & vbNewLine)

    Next

```

UninitDFTTSEngine

Unloads the synthesizer voice DB's

Synopsis

```

[C#]
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int UninitDFTTSEngine();

```

```

[VB.NET]
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function UninitDFTTSEngine() As Integer
End Function

```

Parameters

None.

Description

The function frees the assigned memory by unloading the synthesizer voice DB's.

It must be called before the program exits otherwise memory corruption will occur!

Return Values

None.

See Also

InitDFTTSEngineEx3()

Example

```

[C#]
UninitDFTTSEngine();

```

```

[VB.NET]
UninitDFTTSEngine()

```

MakeLanguage

Creates a language id from a main language id and a sub-language id.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]  
public static extern int MakeLanguage(int mainlang, int sublang);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _  
Public Shared Function MakeLanguage(ByVal mainlang As Integer, _  
                                     ByVal sublang As Integer _  
                                     ) As Integer
```

Parameters

mainlang

The id of the main or primary language. See the .NET SDK regions for all language constants.

sublang

The id of the sub-language. See the .NET SDK regions for all language constants.

Description

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Return Values

The language id needed for the SDK.

See Also

GetMainLanguage()

GetSubLanguage()

GetMainLanguage

A Helper function that returns the main language id from a language id.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]  
public static extern int GetMainLanguage(int lang);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _  
Public Shared Function GetMainLanguage(ByVal lang As Integer) As Integer
```

Parameters

lang

The id of the language.

Description

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Return Values

The id of the primary language used in the creation of the language id (lang parameter).

See Also

MakeLanguage()

GetSubLanguage()

GetSubLanguage

A Helper function that returns the sub-language id from a language id.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]  
public static extern int GetSubLanguage(int lang);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _  
Public Shared Function GetSubLanguage(ByVal lang As Integer) As Integer
```

Parameters

lang

The id of the language.

Description

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Return Values

The id of the sub-language used in the creation of the language id (lang parameter).

See Also

MakeLanguage()

GetMainLanguage()

LoadDFTTSUserDict

Loads the User Dictionary.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]  
public static extern int LoadDFTTSUserDict(int iDictIndex,  
    [MarshalAs(UnmanagedType.LPStr)] string szDictName,  
    [MarshalAs(UnmanagedType.LPStr)] string szDictFileName, int vet, int lang,  
    [MarshalAs(UnmanagedType.LPStr)] string szVoiceName,  
    [MarshalAs(UnmanagedType.LPStr)] string szDictContents);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _  
Public Shared Function LoadDFTTSUserDict(ByVal iDictIndex As Integer, _  
    <MarshalAs(UnmanagedType.LPStr)> ByVal szDictName As System.String, _  
    <MarshalAs(UnmanagedType.LPStr)> ByVal szDictFileName As System.String, _  
    ByVal vet As Integer, _  
    ByVal lang As Integer, _  
    <MarshalAs(UnmanagedType.LPStr)> ByVal szVoiceName As System.String, _  
    <MarshalAs(UnmanagedType.LPStr)> ByVal szDictContents As System.String) As Integer  
End Function
```

Parameters

iDictIndex

Dictionary index in case of using more than one user dictionary (only valid for NEOSPEECHVOICETEXT).

Default user dictionary uses the value of 0 and it can take the values between 1~1023. Use -1 for the default value.

szDictName

Dictionary name (only valid with engine ATTNV (*internal use only*)).

szDictFileName

The name (path) of user dictionary file (only valid for NEOSPEECHVOICETEXT and CEPSTRAL).

vet

Load for the specified engine type (NEOSPEECHVOICETEXT, CEPSTRAL, or ATTNV).

lang

Language id (only valid for NEOSPEECHVOICETEXT).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

szVoiceName

Voice to load the dictionary for (only valid for CEPSTRAL, where the dictionary is loaded for a voice).

szDictContents

Phoneme contents for the dictionary (only ATTNV where instead of a dictionary file we supply the actual character phoneme contents).

Description

This function loads the user dictionaries that are separate from and in addition to the default dictionary that is included in the TTS DB.

This function has to be used after the completion of the loading of the synthesizer DB's. iDictIndex is used during synthesis by NEOSPEECHVOICETEXT.

Note: For detailed instructions on how to build dictionaries for each engine provider, see instructions provided with the SDK in folder DictionarySystems.

Return Values

LOAD_USER_DICT_SUCCESS is returned when user dictionary is successfully loaded. The following common error codes are returned when errors occur:

[LOAD_USER_DICT_ERROR_DICTIDX_NOT_VALID] iDictIndex value is not within the valid range

[LOAD_USER_DICT_ERROR_DICT_ALREADY_LOADED] User dictionary file corresponding to *dictidx* is already loaded.

[LOAD_USER_DICT_ERROR_NO_DICT_FILE_OR_ENTRY] Loading failed because there was no user dictionary files or valid entry

[LOAD_USER_DICT_ERROR_OTHER] Other errors

See enumeration LoadDFTTSUserDictReturnValue for more error codes.

See Also

UnloadDFTTSUserDict()

Example (see C# and VB.NET samples)

[C#]

```
LoadDFTTSUserDictReturnValue dictreturn = LoadDFTTSUserDictReturnValue.LOAD_USER_DICT_SUCCESS;

//User Dictionary Loading:
//(read dictionary documentation per engine located in folder DictionarySystems)

//This is how to load a dictionary for the NeoSpeech provider (the dictionary is loaded _
//by language from a file (see file in exe directory) (the dict name, voice name, _
//dict content do NOT matter), the dictionary number matters)

//MakeLanguage(LANG_ENGLISH, SUBLANG_ENGLISH_US) = 1033 - You can just use 1033

dictreturn = (LoadDFTTSUserDictReturnValue)LoadDFTTSUserDict(1, null,
    "neospeech_userdict_eng.csv",
    DFTTSVoiceEngineType.NEOSPEECHVOICETEXT,
    MakeLanguage(LANG_ENGLISH, SUBLANG_ENGLISH_US), null, null);

//This is how to load a dictionary for the Cepstral provider (the dictionary is loaded
//by voice from a file (see file in exe directory);
//The dictionary number, dict name, dict content do NOT matter.)

dictreturn = (LoadDFTTSUserDictReturnValue)LoadDFTTSUserDict(-1, null,
    "cepstral_dictionary.txt",
    DFTTSVoiceEngineType.CEPSTRAL, LANG_SUBLANG_NEUTRAL, "David", null);

if (dictreturn != LoadDFTTSUserDictReturnValue.LOAD_USER_DICT_SUCCESS)
{
    MessageBox.Show("SDK Load Dictionary failed with error: " + dictreturn, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

[VB.NET]

```
Dim dictreturn As LoadDFTTSUserDictReturnValue

'User Dictionary Loading:
'(read dictionary documentation per engine located in folder DictionarySystems)

'This is how to load a dictionary for the NeoSpeech provider (the dictionary is loaded _
'by language from a file (see file in exe directory) (the dict name, voice name, _
'dict content do NOT matter), the dictionary number matters)

'MakeLanguage(LANG_ENGLISH, SUBLANG_ENGLISH_US) = 1033 - You can just use 1033

dictreturn = LoadDFTTSUserDict(1, vbNullString, _
    "neospeech_userdict_eng.csv", _
```



```

DFTTSVoiceEngineType.NEOSPEECHVOICETEXT, _
    MakeLanguage(LANG_ENGLISH, SUBLANG_ENGLISH_US), _
    vbNullString, vbNullString)

'This is how to load a dictionary for the Cepstral provider (the dictionary is loaded
'by voice from a file (see file in exe directory);
'The dictionary number, dict name, dict content do NOT matter.)

dictreturn = LoadDFTTSUserDict(-1, vbNullString, _
    "cepstral_dictionary.txt", _
    DFTTSVoiceEngineType.CEPSTRAL, LANG_SUBLANG_NEUTRAL, _
    "David", vbNullString)

If dictreturn <> LoadDFTTSUserDictReturnValue.LOAD_USER_DICT_SUCCESS Then
    MessageBox.Show("SDK Load Dictionary failed with error: " & dictreturn, "Error", _
        MessageBoxButtons.OK, MessageBoxIcon.Error)
End If

```

UnloadDFTTSUserDict

Unloads the User Dictionary.

Synopsis

[C#]

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int UnloadDFTTSUserDict(int iDictIndex,
[MarshalAs(UnmanagedType.LPStr)] string szDictName,
int vet, int lang);

```

[VB.NET]

```

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function UnloadDFTTSUserDict( _
    ByVal iDictIndex As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szDictName As System.String, _
    ByVal vet As Integer, _
    ByVal lang As Integer) As Integer
End Function

```

Parameters

iDictIndex

Dictionary index in case of using more than one user dictionary (only valid for NEOSPEECHVOICETEXT).

Default user dictionary uses the value of 0 and it can take the values between 1~1023.

szDictName

Dictionary name (only valid with engine ATTNV (**internal use only**)).

vet

Unload for the specified engine type (NEOSPEECHVOICETEXT or ATTNV).

lang

Language id (only valid for NEOSPEECHVOICETEXT).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Description

The function unloads the user dictionary.

Any user dictionary that has not been unloaded is automatically unloaded during the process of unloading the synthesizer DB.

CEPSTRAL engine does not support user dictionary unloading. The dictionary data unloads with the unloading the voice DB.

Return Values

UNLOAD_USER_DICT_SUCCESS is returned when user dictionary is successfully loaded. The following error codes are returned when errors occur:

[UNLOAD_USER_DICT_ERROR_DICTIDX_NOT_VALID] iDictIndex value is not within the valid range

[UNLOAD_USER_DICT_ERROR_DICT_UNLOADED] User dictionary file corresponding to iDictIndex is already unloaded.

[UNLOAD_USER_DICT_ERROR_OTHER] Other errors

See Also

LoadDFTTSUserDict()

Example

[C#]

```
UnloadDFTTSUserDict(1, "", DFTTSVoiceEngineType.NEOSPEECHVOICETEXT, 1033);
```

[VB.NET]

```
UnloadDFTTSUserDict(1, "", DFTTSVoiceEngineType.NEOSPEECHVOICETEXT, 1033)
```

DFTTSSpeak

It plays synthesized TTS output through a sound card.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSSpeak(IntPtr hwndWinOwner, int vet,
    [MarshalAs(UnmanagedType.LPStr)] string szVoiceName,
    int iVoiceID, int lang,
    [MarshalAs(UnmanagedType.LPStr)] string szText,
    int iPitch,
    int iSpeed, int iVolume, int iPause, int iDictID,
    int ttTextType, int ofOutPutFormat);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSSpeak(ByVal hwndWinOwner As IntPtr, _
    ByVal vet As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szVoiceName As System.String, _
    ByVal iVoiceID As Integer, ByVal lang As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szText As System.String, _
    ByVal iPitch As Integer, ByVal iSpeed As Integer, _
    ByVal iVolume As Integer, _
    ByVal iPause As Integer, _
    ByVal iDictID As Integer, _
    ByVal ttTextType As Integer, _
    ByVal ofOutPutFormat As Integer) As Integer
End Function
```

Parameters

hwndWinOwner

Window handle (WIN32) where to send speech event messages.

vet

Which engine type (NEOSPEECHVOICETEXT, CEPSTRAL, or ATTNV (*internal use only*)).

szVoiceName

Internal voice name (matters for CEPSTRAL and ATTNV). Example: "David" or "Mike16".

iVoiceID

Internal voice id (matters for NEOSPEECHVOICETEXT).

See the API declarations (values of NEOSPEECH_KATE_ENG, etc)

lang

Language id (only valid for NEOSPEECHVOICETEXT).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

szText

Text string to be synthesized (no size limit).

iPitch

Defines the pitch of synthesized voice.

The default value is set to 100(%). The possible pitch range varies depending on the engine type (see below). ATTNV does not support pitch modifications by design.

For -1, use default value.

iSpeed

Defines the speed of synthesized voice. The default value is set to 100%. The range varies depending on the engine type (see below).

For -1, use default value.

iVolume

Defines the volume of synthesized voice. The default value is set to 100%. The range varies depending on the engine type (see below).

For -1, use default value.

iPause

Defines the length of pause between sentences of synthesized voice

(NEOSPEECHVOICETEXT ONLY). The default value is set to 670(msec). The range is 0~20000(msec) and the lower value indicates shorter pause.

For -1, use default value

The following enumeration outlines the default values and supported ranges between engines (see API declaration):

[C#]

```
public enum DFTTSPredefinedSpeechParams
{
```

```
    DF_TTS_DEFAULT_PITCH = 100,
    ///*100%*/
    DF_TTS_DEFAULT_SPEED = 100,
    ///*100%*/
    DF_TTS_DEFAULT_VOLUME = 100,
    ///*100%*/
    DF_TTS_DEFAULT_PAUSE = 670,
    ///*670 msec*/
    DF_TTS_NEOSPEECH_MIN_PITCH = 50,
    DF_TTS_NEOSPEECH_MAX_PITCH = 200,
    DF_TTS_NEOSPEECH_MIN_SPEED = 50,
    DF_TTS_NEOSPEECH_MAX_SPEED = 400,
    DF_TTS_NEOSPEECH_MIN_VOLUME = 0,
    DF_TTS_NEOSPEECH_MAX_VOLUME = 500,
    DF_TTS_NEOSPEECH_MIN_PAUSE = 0,
    DF_TTS_NEOSPEECH_MAX_PAUSE = 20000,
```

```

    DF_TTS_CEPSTRAL_MIN_PITCH = 100,
    DF_TTS_CEPSTRAL_MAX_PITCH = 500,
    DF_TTS_CEPSTRAL_MIN_SPEED = 0,
    DF_TTS_CEPSTRAL_MAX_SPEED = 400,
    DF_TTS_CEPSTRAL_MIN_VOLUME = 0,
    DF_TTS_CEPSTRAL_MAX_VOLUME = 500,
    DF_TTS_ATTNV_MIN_SPEED = 13,
    DF_TTS_ATTNV_MAX_SPEED = 800,
    DF_TTS_ATTNV_MIN_VOLUME = 0,
    DF_TTS_ATTNV_MAX_VOLUME = 500,
    DF_TTS_ATTNV_MIN_PITCH = 0,
    ///*AT&T NV do not support this*/
    DF_TTS_ATTNV_MAX_PITCH = 0,
    ///*AT&T NV do not support this*/
    DF_TTS_MACOSXSPMAN_MIN_PITCH = 1,
    DF_TTS_MACOSXSPMAN_MAX_PITCH = 1000,
    DF_TTS_MACOSXSPMAN_MIN_SPEED = 1,
    DF_TTS_MACOSXSPMAN_MAX_SPEED = 1000,
    DF_TTS_MACOSXSPMAN_MIN_VOLUME = 100,
    DF_TTS_MACOSXSPMAN_MAX_VOLUME = 500,
    DF_TTS_MSSAPI_MIN_PITCH = 30,
    DF_TTS_MSSAPI_MAX_PITCH = 350,
    DF_TTS_MSSAPI_MIN_SPEED = 30,
    DF_TTS_MSSAPI_MAX_SPEED = 350,
    DF_TTS_MSSAPI_MIN_VOLUME = 0,
    DF_TTS_MSSAPI_MAX_VOLUME = 100

```

```

}

```

[VB.NET]

Public Enum DFTTSPredefinedSpeechParams

```

    DF_TTS_DEFAULT_PITCH = 100 '/*100%*/
    DF_TTS_DEFAULT_SPEED = 100 '/*100%*/
    DF_TTS_DEFAULT_VOLUME = 100 '/*100%*/
    DF_TTS_DEFAULT_PAUSE = 670 '/*670 msec*/
    DF_TTS_NEOSPEECH_MIN_PITCH = 50
    DF_TTS_NEOSPEECH_MAX_PITCH = 200
    DF_TTS_NEOSPEECH_MIN_SPEED = 50
    DF_TTS_NEOSPEECH_MAX_SPEED = 400
    DF_TTS_NEOSPEECH_MIN_VOLUME = 0
    DF_TTS_NEOSPEECH_MAX_VOLUME = 500
    DF_TTS_NEOSPEECH_MIN_PAUSE = 0
    DF_TTS_NEOSPEECH_MAX_PAUSE = 20000
    DF_TTS_CEPSTRAL_MIN_PITCH = 100
    DF_TTS_CEPSTRAL_MAX_PITCH = 500
    DF_TTS_CEPSTRAL_MIN_SPEED = 0
    DF_TTS_CEPSTRAL_MAX_SPEED = 400
    DF_TTS_CEPSTRAL_MIN_VOLUME = 0
    DF_TTS_CEPSTRAL_MAX_VOLUME = 500
    DF_TTS_ATTNV_MIN_SPEED = 13
    DF_TTS_ATTNV_MAX_SPEED = 800
    DF_TTS_ATTNV_MIN_VOLUME = 0
    DF_TTS_ATTNV_MAX_VOLUME = 500
    DF_TTS_ATTNV_MIN_PITCH = 0 '/*AT&T NV do not support this*/
    DF_TTS_ATTNV_MAX_PITCH = 0 '/*AT&T NV do not support this*/
    DF_TTS_MACOSXSPMAN_MIN_PITCH = 1

```

```

DF_TTS_MACOSXSPMAN_MAX_PITCH = 1000
DF_TTS_MACOSXSPMAN_MIN_SPEED = 1
DF_TTS_MACOSXSPMAN_MAX_SPEED = 1000
DF_TTS_MACOSXSPMAN_MIN_VOLUME = 100
DF_TTS_MACOSXSPMAN_MAX_VOLUME = 500
DF_TTS_MSSAPI_MIN_PITCH = 30
DF_TTS_MSSAPI_MAX_PITCH = 350
DF_TTS_MSSAPI_MIN_SPEED = 30
DF_TTS_MSSAPI_MAX_SPEED = 350
DF_TTS_MSSAPI_MIN_VOLUME = 0
DF_TTS_MSSAPI_MAX_VOLUME = 100

```

End Enum

iDictID

ID of the user dictionary when multiple user dictionaries are in use. The default dictionary uses value 0 and the range is between 1~1023 (ONLY applicable with NEOSPEECHVOICETEXT).

For -1, use default value.

ttTextType

Defines the text type to be synthesized.

For regular text, use DFTTS_TEXT_TYPE_PLAIN, and for VoiceXML/SSML text, use DFTTS_TEXT_TYPE_XML. The value of -1 is regarded as regular text.

ofOutPutFormat

Speech output format (MSSAPI only). Must be -1 since MSSAPI is only used with unmanaged SDK applications.

Notes

The function synthesizes text entries and produces the TTS output through a sound card. **On Windows, it sends the message WM_USER+12359 with the beginning character number of the word being spoken (zero-based and in regard to the whole text fed to the engine) in WPARAM and the end character number in LPARAM. The SDK sends WM_USER+12359 with WPARAM "0" and LPARAM "-1" when synthesizing is complete. To test this behavior use Spy++. Check the sample code to see how these values are acquired by capturing the message directly in the window procedure (in the sample**

WIN_SPEAK_MSG = WM_USER+12359):

[C#]

```

protected override void WndProc(ref Message m)
{
    switch (m.Msg)
    {
        case WIN_SPEAK_MSG:
            StringBuilder sb = new StringBuilder();
            sb.Append("OnWord: Start Character: ");
            sb.Append(m.WParam);
            sb.Append(", End Character: ");
            sb.Append((int)m.LParam);

            this.label1.Text = sb.ToString();
            break;
    }
}

```

```

        case WIN_EXPORT_MSG:
            MessageBox.Show("Export succesful!", "Success");
            break;

        default:
            base.WndProc(ref m);
            break;
    }

```

[VB.NET]

```

Protected Overloads Overrides Sub WndProc(ByRef m As Message)
    Select Case m.Msg
        Case WIN_SPEAK_MSG
            Dim sb As New StringBuilder()
            sb.Append("OnWord: Start Character: ")
            sb.Append(m.WParam)
            sb.Append(", End Character: ")
            sb.Append(CInt(m.LParam))

            Me.label1.Text = sb.ToString()
            Exit Select
        Case WIN_EXPORT_MSG

            MessageBox.Show("Export succesful!", "Success")
            Exit Select
        Case Else

            MyBase.WndProc(m)
            Exit Select
    End Select

End Sub

```

If DFTTSpeak() is called again during playback, it stops what was being played for the specified engine provider and plays the new TTS output that was requested. This behavior may not be supported with all engines.

Return Values

DFTTS_SPEAK_SUCCESS is returned when it was executed successfully. The following error codes are returned when errors occur:

[DFTTS_SPEAK_ERROR_CHANNEL_MEM_FAIL] Failed to secure channel memory
 [DFTTS_SPEAK_ERROR_TEXT_NULL] The text string is a NULL pointer
 [DFTTS_SPEAK_ERROR_TEXT_ZERO_LEN] The length of text string is 0
 [DFTTS_SPEAK_ERROR_DB_NOT_LOADED] The TTS DB of the voice requested is not loaded
 [DFTTS_SPEAK_ERROR_SET_SOUND_CARD_FAIL] Failed to set the sound card

Additional return values:

DFTTS_SPEAK_ERROR_UNIMPLEMENTED,
 DFTTS_SPEAK_ERROR_INTERNAL,
 DFTTS_SPEAK_ERROR_INVALID_PARAM,
 DFTTS_SPEAK_ERROR_INVALID_POINTER,

DFTTS_SPEAK_ERROR_OBJECT_NOT_FOUND,
DFTTS_SPEAK_ERROR_UNKNOWN_ENCODING,
DFTTS_SPEAK_ERROR_INTERRUPTED,
DFTTS_SPEAK_ERROR_INVALID_VOICE,
DFTTS_SPEAK_ERROR_WRONG_EVENT,
DFTTS_SPEAK_ERROR_ENGINE_INUSE,
DFTTS_SPEAK_ERROR_NETWORK_ERROR,
DFTTS_SPEAK_ERROR_INVALID_KEY,
DFTTS_SPEAK_ERROR_QUEUE_FULL,
DFTTS_SPEAK_ERROR_TOKEN_TIMEOUT,
DFTTS_SPEAK_ERROR_FAILED,
DFTTS_SPEAK_ERROR_INVALIDARG,
DFTTS_SPEAK_ERROR_OUTOFMEMORY,
DFTTS_SPEAK_ERROR_NOTIMPL,
DFTTS_SPEAK_ERROR_ABORT,
DFTTS_SPEAK_ERROR_UNKNOWN,
DFTTS_SPEAK_ERROR_BADHANDLE,
DFTTS_SPEAK_ERROR_EXCEPTION,
DFTTS_SPEAK_ERROR_EMPTY,
DFTTS_SPEAK_ERROR_FULL,
DFTTS_SPEAK_ERROR_INVALIDSTATE,
DFTTS_SPEAK_ERROR_BADVERSION,
DFTTS_SPEAK_ERROR_INSUFFICIENT_BUFFER,
DFTTS_SPEAK_ERROR_UNSUPPORTED,
DFTTS_SPEAK_ERROR_NOLICENSE,
DFTTS_SPEAK_ERROR_CREATECHILDPROCESS_FAILED,
DFTTS_SPEAK_ERROR_NOENVIRONMENTPATH,
DFTTS_SPEAK_ERROR_TIMEOUT,
DFTTS_SPEAK_ERROR_OUTOFRESOURCES,
DFTTS_SPEAK_ERROR_NOVOICES,
DFTTS_SPEAK_ERROR_CREATEFAIL,
DFTTS_SPEAK_ERROR_CONNECTFAIL,
DFTTS_SPEAK_ERROR_BINDFAIL,
DFTTS_SPEAK_ERROR_LISTENFAIL,
DFTTS_SPEAK_ERROR_CONNECTIONCLOSED,
DFTTS_SPEAK_ERROR_ACCEPTFAIL,
DFTTS_SPEAK_ERROR_SOCKETTIMEOUT,
DFTTS_SPEAK_ERROR_SOCKETERROR,
DFTTS_SPEAK_ERROR_NOMORESERSERVERS,
DFTTS_SPEAK_ERROR_SOCKET_EWOULDBLOCK,
DFTTS_SPEAK_ERROR_SOCKET_EINPROGRESS,
DFTTS_SPEAK_ERROR_SOCKET_EALREADY,
DFTTS_SPEAK_ERROR_SOCKET_ENOTSOCK,
DFTTS_SPEAK_ERROR_SOCKET_EDESTADDRREQ,
DFTTS_SPEAK_ERROR_SOCKET_EMSGSIZE,
DFTTS_SPEAK_ERROR_SOCKET_EPROTOTYPE,
DFTTS_SPEAK_ERROR_SOCKET_ENOPROTOOPT,
DFTTS_SPEAK_ERROR_SOCKET_EPROTONOSUPPORT,
DFTTS_SPEAK_ERROR_SOCKET_ESOCKTNOSUPPORT,
DFTTS_SPEAK_ERROR_SOCKET_EOPNOTSUPP,
DFTTS_SPEAK_ERROR_SOCKET_EPFNOSUPPORT,
DFTTS_SPEAK_ERROR_SOCKET_EAFNOSUPPORT,
DFTTS_SPEAK_ERROR_SOCKET_EADDRINUSE,

DFTTS_SPEAK_ERROR_SOCKET_EADDRNOTAVAIL,
 DFTTS_SPEAK_ERROR_SOCKET_ENETDOWN,
 DFTTS_SPEAK_ERROR_SOCKET_ENETUNREACH,
 DFTTS_SPEAK_ERROR_SOCKET_ENETRESET,
 DFTTS_SPEAK_ERROR_SOCKET_ECONNABORTED,
 DFTTS_SPEAK_ERROR_SOCKET_ECONNRESET,
 DFTTS_SPEAK_ERROR_SOCKET_ENOBUFS,
 DFTTS_SPEAK_ERROR_SOCKET_EISCONN,
 DFTTS_SPEAK_ERROR_SOCKET_ENOTCONN,
 DFTTS_SPEAK_ERROR_SOCKET_ESHUTDOWN,
 DFTTS_SPEAK_ERROR_SOCKET_ETOOMANYREFS,
 DFTTS_SPEAK_ERROR_SOCKET_ECONNREFUSED,
 DFTTS_SPEAK_ERROR_SOCKET_ELOOP,
 DFTTS_SPEAK_ERROR_SOCKET_ENAMETOOLONG,
 DFTTS_SPEAK_ERROR_SOCKET_EHOSTDOWN,
 DFTTS_SPEAK_ERROR_SOCKET_EHOSTUNREACH,
 DFTTS_SPEAK_ERROR_SOCKET_ENOTEMPTY,
 DFTTS_SPEAK_ERROR_SOCKET_EPROCLIM,
 DFTTS_SPEAK_ERROR_THREADSTARTED,
 DFTTS_SPEAK_ERROR_THREADNOTSTARTED,
 DFTTS_SPEAK_ERROR_THREADCOULDNOTCREATE,
 DFTTS_SPEAK_ERROR_BADNVFILE,
 DFTTS_SPEAK_ERROR_NOAUDIODRIVER,
 DFTTS_SPEAK_ERROR_DICTNOTFOUND,
 DFTTS_SPEAK_ERROR_ALREADYPLAYING,
 DFTTS_SPEAK_ERROR_AUDIOFORMATNOTSUPPORTED,
 DFTTS_SPEAK_ERROR_XML_INVALID,
 DFTTS_SPEAK_ERROR_WL_INVALID,
 DFTTS_SPEAK_ERROR_INVALIDPHONESET,
 DFTTS_SPEAK_ERROR_INVALIDPHONEME,
 DFTTS_SPEAK_ERROR_MSGQ_CREATEFAILED,
 DFTTS_SPEAK_ERROR_MSGQ_ALREADYEXISTS,
 DFTTS_SPEAK_ERROR_MSGQ_NOTFOUND,
 DFTTS_SPEAK_ERROR_MSGQ_INVALIDOP,
 DFTTS_SPEAK_ERROR_MSGQ_NOTOPEN,
 DFTTS_SPEAK_ERROR_MSGQ_LOCKFAILED,
 DFTTS_SPEAK_ERROR_MSGQ_ABANDONED,
 DFTTS_SPEAK_ERROR_MSGQ_OPENFAILED,

[DFTTS_SPEAK_ERROR_OTHER] Other errors

Example (see C# and VB.NET samples)

[C#]

```

int iSelIndex = ComboBox1.SelectedIndex;

cVoiceData oVData = (cVoiceData)ComboBox1.Items[iSelIndex];

DFTTSSpeakReturnValue result = (DFTTSSpeakReturnValue)DFTTSSpeak(hWnd,
    oVData.vet, oVData.sVoiceName, oVData.iVoiceId, oVData.lang,
    textBox1.Text, -1, -1, -1, -1,
    1, (int)DFTTSTextType.DFTTS_TEXT_TYPE_XML, -1);

if (result != DFTTSSpeakReturnValue.DFTTS_SPEAK_SUCCESS)
{

```

```

        MessageBox.Show("SDK Speak error: " + result.ToString(), "Error");
    }

[VB.NET]
Dim iSelIndex As Integer = ComboBox1.SelectedIndex

Dim oVData As cVoiceData = ComboBox1.Items.Item(iSelIndex)

Dim result As DFTTSSpeakReturnValue = DirectCast(DFTTSSpeak(hWnd, _
    oVData.vet, oVData.sVoiceName, oVData.iVoiceId, oVData.lang, textBox1.Text, _
    -1, -1, -1, -1, 1, CInt(DFTTSTextType.DFTTS_TEXT_TYPE_XML), -1), _
    DFTTSSpeakReturnValue)

If result <> DFTTSSpeakReturnValue.DFTTS_SPEAK_SUCCESS Then

    MessageBox.Show("SDK Speak error: " & result.ToString(), "Error")

End If

```

DFTTSStop

It stops the playback of synthesized voices from a sound card.

Synopsis

```

[C#]
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSStop(int vet, int lang);

```

```

[VB.NET]
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSStop(ByVal vet As Integer, _
    ByVal lang As Integer) As Integer
End Function

```

Parameters

vet

Which engine type (NEOSPEECHVOICETEXT, CEPSTRAL, or ATTNV (*internal use only*)).

lang

Language id (only valid for NEOSPEECHVOICETEXT, for all other engines use 0 (zero)).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Notes

Using DFTTSSpeak(), stops the playback of synthesized voice before synthesizing.

Return Values

DFTTS_STOP_SUCCESS,
DFTTS_STOP_ERROR_INVALID_POINTER,
DFTTS_STOP_ERROR_OTHER

DFTTSPause

Pauses the playback of synthesized voices from a sound card.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]  
public static extern int DFTTSPause(int vet, int lang);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _  
Public Shared Function DFTTSPause(ByVal vet As Integer, _  
ByVal lang As Integer) As Integer  
End Function
```

Parameters

vet

Which engine type (NEOSPEECHVOICETEXT, CEPSTRAL, or ATTNV (*internal use only*)).

lang

Language id (only valid for NEOSPEECHVOICETEXT, for all other engines use 0 (zero)).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Description

Pauses the playback of synthesized voice that is carried out using DFTTSSpeak().

Return Values

```
DFTTS_PAUSE_SUCCESS,  
DFTTS_PAUSE_ERROR_UNIMPLEMENTED,  
DFTTS_PAUSE_ERROR_INTERNAL,  
DFTTS_PAUSE_ERROR_INVALID_PARAM,  
DFTTS_PAUSE_ERROR_INVALID_POINTER,  
DFTTS_PAUSE_ERROR_OBJECT_NOT_FOUND,  
DFTTS_PAUSE_ERROR_UNKNOWN_ENCODING,  
DFTTS_PAUSE_ERROR_INTERRUPTED,  
DFTTS_PAUSE_ERROR_INVALID_VOICE,  
DFTTS_PAUSE_ERROR_WRONG_EVENT,  
DFTTS_PAUSE_ERROR_ENGINE_INUSE,  
DFTTS_PAUSE_ERROR_NETWORK_ERROR,  
DFTTS_PAUSE_ERROR_INVALID_KEY,  
DFTTS_PAUSE_ERROR_QUEUE_FULL,  
DFTTS_PAUSE_ERROR_TOKEN_TIMEOUT,  
DFTTS_PAUSE_ERROR_OTHER
```

DFTTSResume

Resumes the playback of the synthesized voice from a sound card.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]  
public static extern int DFTTSResume(int vet, int lang);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _  
Public Shared Function DFTTSResume(ByVal vet As Integer, _  
ByVal lang As Integer) As Integer  
End Function
```

Parameters

vet

Which engine type (NEOSPEECHVOICETEXT, CEPSTRAL, or ATTNV (*internal use only*)).

lang

Language id (only valid for NEOSPEECHVOICETEXT, for all other engines use 0 (zero)).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

Description

Resumes the playback of synthesized voice that was paused using DFTTSPause().

Return Values

DFTTS_RESUME_SUCCESS if succeeded (check [enum](#) DFTTSResumeReturnValue for the list of all error codes)

DFTTSExportToFileEx

It saves the synthesized output as a file.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSExportToFileEx(int vet,
    [MarshalAs(UnmanagedType.LPStr)] string szVoiceName,
    int iVoiceID, int lang,
    [MarshalAs(UnmanagedType.LPStr)] string szText,
    int iPitch, int iSpeed,
    int iVolume, int iPause, int iDictID, int ttTextType,
    [MarshalAs(UnmanagedType.LPStr)] string szFilePath,
    int ffFileFormat,
    [MarshalAs(UnmanagedType.LPStr)] string szAudioEncoding,
    int iAudioSamplingRate,
    int iAudioChannels);
```

[VB.NET]

```
<DllImport("dftts.dll", CharSet:=CharSet.Auto)>
Public Shared Function DFTTSExportToFileEx(ByVal _vet As Integer, _
<MarshalAs(UnmanagedType.LPStr)> ByVal szVoiceName As System.String, _
ByVal iVoiceID As Integer, _
ByVal lang As Integer, _
<MarshalAs(UnmanagedType.LPStr)> ByVal szText As System.String, _
ByVal iPitch As Integer, _
ByVal iSpeed As Integer, _
ByVal iVolume As Integer, _
ByVal iPause As Integer, _
ByVal iDictID As Integer, _
ByVal ttTextType As Integer, _
<MarshalAs(UnmanagedType.LPStr)> ByVal szFilePath As System.String, _
ByVal ffFileFormat As Integer, _
<MarshalAs(UnmanagedType.LPStr)> ByVal szAudioEncoding As _
System.String, _
ByVal iAudioSamplingRate As Integer, _
ByVal iAudioChannels As Integer) As Integer
End Function
```

Parameters

hwndWinOwner

Window handle (WIN32) where to send speech event messages

vet

Which engine type (NEOSPEECHVOICETEXT, CEPSTRAL, or ATTNV (*internal use only*)).

szVoiceName

Internal voice name (matters for CEPSTRAL and ATTNV). Example: "David" or "Mike16".

iVoiceID

Internal voice id (matters for NEOSPEECHVOICETEXT).

See the API declarations (values of `NEOSPEECH_KATE_ENG`, etc)

lang

Language id (only valid for NEOSPEECHVOICETEXT).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use `MakeLanguage` with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with `MakeLanguage` id, use `GetMainLanguage`.

To obtain a sub-language id from a created with `MakeLanguage` id, use `GetSubLanguage`.

szText

Text string to be synthesized (no size limit).

iPitch

Defines the pitch of synthesized voice.

The default value is set to 100(%). The possible pitch range varies depending on the engine type (see below). ATTNV does not support pitch modifications by design.

For -1, use default value.

iSpeed

Defines the speed of synthesized voice. The default value is set to 100%. The range varies depending on the engine type (see below).

For -1, use default value.

iVolume

Defines the volume of synthesized voice. The default value is set to 100%. The range varies depending on the engine type (see below).

For -1, use default value.

iPause

Defines the length of pause between sentences of synthesized voice (NEOSPEECHVOICETEXT ONLY). The default value is set to 670(msec). The range is 0~20000(msec) and the lower value indicates shorter pause.

For -1, use default value

The following enumeration outlines the default values and supported ranges between engines (see API declaration):

```
[C#]
public enum DFTTSPredefinedSpeechParams
{
    DF_TTS_DEFAULT_PITCH = 100,
    ///*100%*/
    DF_TTS_DEFAULT_SPEED = 100,
    ///*100%*/
    DF_TTS_DEFAULT_VOLUME = 100,
    ///*100%*/
    DF_TTS_DEFAULT_PAUSE = 670,
    ///*670 msec*/
    DF_TTS_NEOSPEECH_MIN_PITCH = 50,
    DF_TTS_NEOSPEECH_MAX_PITCH = 200,
    DF_TTS_NEOSPEECH_MIN_SPEED = 50,
    DF_TTS_NEOSPEECH_MAX_SPEED = 400,
    DF_TTS_NEOSPEECH_MIN_VOLUME = 0,
    DF_TTS_NEOSPEECH_MAX_VOLUME = 500,
    DF_TTS_NEOSPEECH_MIN_PAUSE = 0,
    DF_TTS_NEOSPEECH_MAX_PAUSE = 20000,
    DF_TTS_CEPSTRAL_MIN_PITCH = 100,
    DF_TTS_CEPSTRAL_MAX_PITCH = 500,
    DF_TTS_CEPSTRAL_MIN_SPEED = 0,
    DF_TTS_CEPSTRAL_MAX_SPEED = 400,
    DF_TTS_CEPSTRAL_MIN_VOLUME = 0,
    DF_TTS_CEPSTRAL_MAX_VOLUME = 500,
    DF_TTS_ATTNV_MIN_SPEED = 13,
    DF_TTS_ATTNV_MAX_SPEED = 800,
    DF_TTS_ATTNV_MIN_VOLUME = 0,
    DF_TTS_ATTNV_MAX_VOLUME = 500,
    DF_TTS_ATTNV_MIN_PITCH = 0,
    ///*AT&T NV do not support this*/
    DF_TTS_ATTNV_MAX_PITCH = 0,
    ///*AT&T NV do not support this*/
    DF_TTS_MACOSXSPMAN_MIN_PITCH = 1,
    DF_TTS_MACOSXSPMAN_MAX_PITCH = 1000,
    DF_TTS_MACOSXSPMAN_MIN_SPEED = 1,
    DF_TTS_MACOSXSPMAN_MAX_SPEED = 1000,
    DF_TTS_MACOSXSPMAN_MIN_VOLUME = 100,
    DF_TTS_MACOSXSPMAN_MAX_VOLUME = 500,
    DF_TTS_MSSAPI_MIN_PITCH = 30,
    DF_TTS_MSSAPI_MAX_PITCH = 350,
    DF_TTS_MSSAPI_MIN_SPEED = 30,
    DF_TTS_MSSAPI_MAX_SPEED = 350,
    DF_TTS_MSSAPI_MIN_VOLUME = 0,
    DF_TTS_MSSAPI_MAX_VOLUME = 100
}
```

```
}
```

```
[VB.NET]
```

```
Public Enum DFTTSPredefinedSpeechParams
```

```
    DF_TTS_DEFAULT_PITCH = 100 '/*100%*/  
    DF_TTS_DEFAULT_SPEED = 100 '/*100%*/  
    DF_TTS_DEFAULT_VOLUME = 100 '/*100%*/  
    DF_TTS_DEFAULT_PAUSE = 670 '/*670 msec*/  
    DF_TTS_NEOSPEECH_MIN_PITCH = 50  
    DF_TTS_NEOSPEECH_MAX_PITCH = 200  
    DF_TTS_NEOSPEECH_MIN_SPEED = 50  
    DF_TTS_NEOSPEECH_MAX_SPEED = 400  
    DF_TTS_NEOSPEECH_MIN_VOLUME = 0  
    DF_TTS_NEOSPEECH_MAX_VOLUME = 500  
    DF_TTS_NEOSPEECH_MIN_PAUSE = 0  
    DF_TTS_NEOSPEECH_MAX_PAUSE = 20000  
    DF_TTS_CEPSTRAL_MIN_PITCH = 100  
    DF_TTS_CEPSTRAL_MAX_PITCH = 500  
    DF_TTS_CEPSTRAL_MIN_SPEED = 0  
    DF_TTS_CEPSTRAL_MAX_SPEED = 400  
    DF_TTS_CEPSTRAL_MIN_VOLUME = 0  
    DF_TTS_CEPSTRAL_MAX_VOLUME = 500  
    DF_TTS_ATTNV_MIN_SPEED = 13  
    DF_TTS_ATTNV_MAX_SPEED = 800  
    DF_TTS_ATTNV_MIN_VOLUME = 0  
    DF_TTS_ATTNV_MAX_VOLUME = 500  
    DF_TTS_ATTNV_MIN_PITCH = 0 '/*AT&T NV do not support this*/  
    DF_TTS_ATTNV_MAX_PITCH = 0 '/*AT&T NV do not support this*/  
    DF_TTS_MACOSXSPMAN_MIN_PITCH = 1  
    DF_TTS_MACOSXSPMAN_MAX_PITCH = 1000  
    DF_TTS_MACOSXSPMAN_MIN_SPEED = 1  
    DF_TTS_MACOSXSPMAN_MAX_SPEED = 1000  
    DF_TTS_MACOSXSPMAN_MIN_VOLUME = 100  
    DF_TTS_MACOSXSPMAN_MAX_VOLUME = 500  
    DF_TTS_MSSAPI_MIN_PITCH = 30  
    DF_TTS_MSSAPI_MAX_PITCH = 350  
    DF_TTS_MSSAPI_MIN_SPEED = 30  
    DF_TTS_MSSAPI_MAX_SPEED = 350  
    DF_TTS_MSSAPI_MIN_VOLUME = 0  
    DF_TTS_MSSAPI_MAX_VOLUME = 100
```

```
End Enum
```

iDictID

ID of the user dictionary when multiple user dictionaries are in use. The default dictionary uses value 0 and the range is between 1~1023 (ONLY applicable with NEOSPEECHVOICETEXT).

For -1, use default value.

ttTextType

Defines the text type to be synthesized.

For regular text, use DFTTS_TEXT_TYPE_PLAIN, and for VoiceXML/SSML text, use DFTTS_TEXT_TYPE_XML. The value of -1 is regarded as regular text.

szFilePath

File path to save the synthesized voice output under.

ffFileFormat

Defines the types of synthesized output formats.

CEPSTRAL: Ignored.

ATTNV: Ignored. It always exports as PCM WAV.

NEOSPEECHVOICETEXT: The followings are the types of synthesized output file format that the DF TTS SDK supports for NeoSpeech VoiceText™:

VT_FILE_API_FMT_S16PCM 16bits Linear PCM

VT_FILE_API_FMT_ALAW 8bits A-law PCM

VT_FILE_API_FMT_MULAW 8bits Mu-law PCM

VT_FILE_API_FMT_DADPCM 4bits Dialogic ADPCM

VT_FILE_API_FMT_S16PCM_WAVE 16bits Linear PCM WAVE

VT_FILE_API_FMT_U08PCM_WAVE 8bits Unsigned Linear PCM WAVE

VT_FILE_API_FMT_IMA_WAVE 4bits IMA ADPCM WAVE

VT_FILE_API_FMT_ALAW_WAVE 8bits A-law PCM WAVE

VT_FILE_API_FMT_MULAW_WAVE 8bits Mu-law PCM WAVE

VT_FILE_API_FMT_MULAW_AU 8bits Mu-law PCM SUN AU

szAudioEncoding, iAudioSamplingRate, iAudioChannels

NEOSPEECHVOICETEXT: Ignored.

ATTNV: Ignored.

CEPSTRAL: Possible values:

szAudioEncoding:

"pcm16", "pcm8" PCM 16 bit/8 bit WAV

"ulaw" - μ -Law (8-bit), "alaw" - A-Law (8-bit)

"riff": Microsoft RIFF (WAV) file

"snd": Sun/NeXT .au (SND) format.

"raw": unheadered audio data, native byte order

"le": unheadered audio data, little-endian (LSB first)

"be": unheadered audio data, big-endian (MSB first)

iAudioSamplingRate:

8000 (8 KHz), 16000 (16 KHz), 11025 (11.025 kHz), etc.

iAudioChannels:

1 (mono), 2 (stereo)

Note: ATTNV ignores all format parameters and always exports in PCM WAV format.

Notes

On Windows, *DFTTSExportToFile...()* SDK function sends the message **WM_USER+12360** with WPARAM "0" and LPARAM "-1" when audio file synthesizing is complete. To test this behavior use Spy++. Check the sample code to see how these values are acquired by capturing the message directly in the window procedure (in the sample WIN_EXPORT_MSG = **WM_USER+12360**):

[C#]

```
protected override void WndProc(ref Message m)
{
    switch (m.Msg)
    {
        case WIN_SPEAK_MSG:
            StringBuilder sb = new StringBuilder();
            sb.Append("OnWord: Start Character: ");
            sb.Append(m.WParam);
            sb.Append(", End Character: ");
            sb.Append((int)m.LParam);

            this.label1.Text = sb.ToString();
            break;

        case WIN_EXPORT_MSG:
            MessageBox.Show("Export succesful!", "Success");
            break;

        default:
            base.WndProc(ref m);
            break;
    }
}
```

[VB.NET]

```
Protected Overloads Overrides Sub WndProc(ByRef m As Message)
    Select Case m.Msg
        Case WIN_SPEAK_MSG
            Dim sb As New StringBuilder()
            sb.Append("OnWord: Start Character: ")
            sb.Append(m.WParam)
            sb.Append(", End Character: ")
            sb.Append(CInt(m.LParam))

            Me.label1.Text = sb.ToString()
            Exit Select
        Case WIN_EXPORT_MSG

            MessageBox.Show("Export succesful!", "Success")
            Exit Select
        Case Else

            MyBase.WndProc(m)
            Exit Select
    End Select

End Sub
```

Return Values

DFTTS_EXPORT_SUCCESS is returned when successfully synthesized and the following error codes are returned when errors occur:

[DFTTS_EXPORT_ERROR_FORMAT_NOT_SUPPORTED] Used format that is not supported.

[DFTTS_EXPORT_ERROR_CHANNEL_MEM_FAIL] Failed to secure channel memory

[DFTTS_EXPORT_ERROR_TEXT_NULL] Text string is a NULL pointer

[DFTTS_EXPORT_ERROR_TEXT_ZERO_LEN] The length of text string is 0.

[DFTTS_EXPORT_ERROR_DB_NOT_LOADED] The TTS DB of the voice requested is not loaded

[DFTTS_EXPORT_ERROR_GEN_FILE_FAIL] Failed to generate the synthesized voice file

Additional return values:

DFTTS_EXPORT_ERROR_BUFFER_NULL,
DFTTS_EXPORT_ERROR_THREAD_IN_USE,
DFTTS_EXPORT_ERROR_UNIMPLEMENTED,
DFTTS_EXPORT_ERROR_INTERNAL,
DFTTS_EXPORT_ERROR_INVALID_PARAM,
DFTTS_EXPORT_ERROR_INVALID_POINTER,
DFTTS_EXPORT_ERROR_OBJECT_NOT_FOUND,
DFTTS_EXPORT_ERROR_UNKNOWN_ENCODING,
DFTTS_EXPORT_ERROR_INTERRUPTED,
DFTTS_EXPORT_ERROR_INVALID_VOICE,
DFTTS_EXPORT_ERROR_WRONG_EVENT,
DFTTS_EXPORT_ERROR_ENGINE_INUSE,
DFTTS_EXPORT_ERROR_NETWORK_ERROR,
DFTTS_EXPORT_ERROR_INVALID_KEY,
DFTTS_EXPORT_ERROR_QUEUE_FULL,
DFTTS_EXPORT_ERROR_TOKEN_TIMEOUT,
DFTTS_EXPORT_ERROR_FAILED,
DFTTS_EXPORT_ERROR_INVALIDARG,
DFTTS_EXPORT_ERROR_OUTOFMEMORY,
DFTTS_EXPORT_ERROR_NOTIMPL,
DFTTS_EXPORT_ERROR_ABORT,
DFTTS_EXPORT_ERROR_UNKNOWN,
DFTTS_EXPORT_ERROR_BADHANDLE,
DFTTS_EXPORT_ERROR_EXCEPTION,
DFTTS_EXPORT_ERROR_EMPTY,
DFTTS_EXPORT_ERROR_FULL,
DFTTS_EXPORT_ERROR_INVALIDSTATE,
DFTTS_EXPORT_ERROR_BADVERSION,
DFTTS_EXPORT_ERROR_INSUFFICIENT_BUFFER,
DFTTS_EXPORT_ERROR_UNSUPPORTED,
DFTTS_EXPORT_ERROR_NOLICENSE,
DFTTS_EXPORT_ERROR_CREATECHILDPROCESS_FAILED,
DFTTS_EXPORT_ERROR_NOENVIRONMENTPATH,
DFTTS_EXPORT_ERROR_TIMEOUT,
DFTTS_EXPORT_ERROR_OUTOFRESOURCES,
DFTTS_EXPORT_ERROR_NOVOICES,
DFTTS_EXPORT_ERROR_CREATEFAIL,

DFTTS_EXPORT_ERROR_CONNECTFAIL,
DFTTS_EXPORT_ERROR_BINDFAIL,
DFTTS_EXPORT_ERROR_LISTENFAIL,
DFTTS_EXPORT_ERROR_CONNECTIONCLOSED,
DFTTS_EXPORT_ERROR_ACCEPTFAIL,
DFTTS_EXPORT_ERROR_SOCKETTIMEOUT,
DFTTS_EXPORT_ERROR_SOCKETERROR,
DFTTS_EXPORT_ERROR_NOMORESERSERVERS,
DFTTS_EXPORT_ERROR_SOCKET_EWOULDBLOCK,
DFTTS_EXPORT_ERROR_SOCKET_EINPROGRESS,
DFTTS_EXPORT_ERROR_SOCKET_EALREADY,
DFTTS_EXPORT_ERROR_SOCKET_ENOTSOCK,
DFTTS_EXPORT_ERROR_SOCKET_EDESTADDRREQ,
DFTTS_EXPORT_ERROR_SOCKET_EMSGSIZE,
DFTTS_EXPORT_ERROR_SOCKET_EPROTOTYPE,
DFTTS_EXPORT_ERROR_SOCKET_ENOPROTOOPT,
DFTTS_EXPORT_ERROR_SOCKET_EPROTONOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_ESOCKTNOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_EOPNOTSUPP,
DFTTS_EXPORT_ERROR_SOCKET_EPFNOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_EAFNOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_EADDRINUSE,
DFTTS_EXPORT_ERROR_SOCKET_EADDRNOTAVAIL,
DFTTS_EXPORT_ERROR_SOCKET_ENETDOWN,
DFTTS_EXPORT_ERROR_SOCKET_ENETUNREACH,
DFTTS_EXPORT_ERROR_SOCKET_ENETRESET,
DFTTS_EXPORT_ERROR_SOCKET_ECONNABORTED,
DFTTS_EXPORT_ERROR_SOCKET_ECONNRESET,
DFTTS_EXPORT_ERROR_SOCKET_ENOBUFS,
DFTTS_EXPORT_ERROR_SOCKET_EISCONN,
DFTTS_EXPORT_ERROR_SOCKET_ENOTCONN,
DFTTS_EXPORT_ERROR_SOCKET_ESHUTDOWN,
DFTTS_EXPORT_ERROR_SOCKET_ETOOMANYREFS,
DFTTS_EXPORT_ERROR_SOCKET_ECONNREFUSED,
DFTTS_EXPORT_ERROR_SOCKET_ELOOP,
DFTTS_EXPORT_ERROR_SOCKET_ENAMETOOLONG,
DFTTS_EXPORT_ERROR_SOCKET_EHOSTDOWN,
DFTTS_EXPORT_ERROR_SOCKET_EHOSTUNREACH,
DFTTS_EXPORT_ERROR_SOCKET_ENOTEMPTY,
DFTTS_EXPORT_ERROR_SOCKET_EPROCLIM,
DFTTS_EXPORT_ERROR_THREADSTARTED,
DFTTS_EXPORT_ERROR_THREADNOTSTARTED,
DFTTS_EXPORT_ERROR_THREADCOULDNOTCREATE,
DFTTS_EXPORT_ERROR_BADNVFILE,
DFTTS_EXPORT_ERROR_NOAUDIODRIVER,
DFTTS_EXPORT_ERROR_DICTNOTFOUND,
DFTTS_EXPORT_ERROR_ALREADYPLAYING,
DFTTS_EXPORT_ERROR_AUDIOFORMATNOTSUPPORTED,
DFTTS_EXPORT_ERROR_XML_INVALID,
DFTTS_EXPORT_ERROR_WL_INVALID,
DFTTS_EXPORT_ERROR_INVALIDPHONESET,
DFTTS_EXPORT_ERROR_INVALIDPHONEME,
DFTTS_EXPORT_ERROR_MSGQ_CREATEFAILED,

DFTTS_EXPORT_ERROR_MSGQ_ALREADYEXISTS,
DFTTS_EXPORT_ERROR_MSGQ_NOTFOUND,
DFTTS_EXPORT_ERROR_MSGQ_INVALIDOP,
DFTTS_EXPORT_ERROR_MSGQ_NOTOPEN,
DFTTS_EXPORT_ERROR_MSGQ_LOCKFAILED,
DFTTS_EXPORT_ERROR_MSGQ_ABANDONED,
DFTTS_EXPORT_ERROR_MSGQ_OPENFAILED,
[DFTTS_EXPORT_ERROR_OTHER] Other errors

Example (see C# and VB.NET samples)

[C#]

```
int iSelIndex = ComboBox1.SelectedIndex;

cVoiceData oVData = (cVoiceData)ComboBox1.Items[iSelIndex];

DFTTSExportReturnValue result = (DFTTSExportReturnValue)DFTTSExportToFileEx(
    oVData.vet, oVData.sVoiceName, oVData.iVoiceId, oVData.lang,
    textBox1.Text, -1, -1, -1, -1, -1,
    (int)DFTTSTextType.DFTTS_TEXT_TYPE_PLAIN, sExportFileName,
    (int)NeoSpeechVoiceTextAudioFormat.VT_FILE_API_FMT_ALAW_WAVE,
    "pcm16", 16000, 2);

if (result != DFTTSExportReturnValue.DFTTS_EXPORT_SUCCESS)
{
    MessageBox.Show("SDK Export error: " + result.ToString(), "Error");
}
```

[VB.NET]

```
Dim iSelIndex As Integer = ComboBox1.SelectedIndex

Dim oVData As cVoiceData = ComboBox1.Items.Item(iSelIndex)

Dim result As DFTTSExportReturnValue = DirectCast( _
    DFTTSExportToFileEx(oVData.vet, _
        oVData.sVoiceName, _
        oVData.iVoiceId, _
        oVData.lang, _
        textBox1.Text, -1, _
        -1, -1, -1, -1, _
        CInt(DFTTSTextType.DFTTS_TEXT_TYPE_PLAIN), _
        sExportFileName, _
        CInt(NeoSpeechVoiceTextAudioFormat.VT_FILE_API_FMT_ALAW_WAVE), _
        "pcm16", 16000, 2), DFTTSExportReturnValue)

If result <> DFTTSExportReturnValue.DFTTS_EXPORT_SUCCESS Then

    MessageBox.Show("SDK Export error: " + result.ToString(), "Error")

End If
```

GetDFTTSEngineInfo

It gets DF TTS SDK engine-related information.

Synopsis

[C#]

```
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int GetDFTTSEngineInfo(int vetVType, int lang, int
    ttsParam, int[] iValue, [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechLicFilePath);
```

```
[VB.NET]
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function GetDFTTSEngineInfo( _
ByVal vetVType As Integer, ByVal lang As Integer, _
ByVal ttsParam As Integer, _
ByVal iValue As Integer(), _
<MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePath As _
System.String) As Integer
End Function
```

Parameters

vetVType

Specifies the DF TTS SDK Engine used. (NEOSPEECHVOICETEXT, CEPSTRAL, ATTNV (internal use only)).

lang

Language id (only valid for NEOSPEECHVOICETEXT, otherwise use 0 (zero)).

The new language system of the SDK is designed after the Windows language id implementation.

See the .NET SDK regions for all language constants. A language id is derived from a primary language id and a sub-language id.

To create a language id you must use MakeLanguage with the desired main and sub-language ids.

U.S. English language id is 1033.

To obtain a primary language id from a created with MakeLanguage id, use GetMainLanguage.

To obtain a sub-language id from a created with MakeLanguage id, use GetSubLanguage.

ttsParam

Specifies the DF TTS SDK Engine (specified by *vetVType*) information to be extracted

DFTTS_VERIFY_CODE Verification Result

DFTTS_MAX_CHANNEL Number of maximum channels specified on the license verification file

DFTTS_LOAD_SUCCESS_CODE Return code for successful synthesizer DB loading

DFTTS_MAX_SPEAKER Maximum number of voices supported. The SDK currently supports 2 U.S. English voices but they cannot co-exist on 1 device.

DFTTS_DEF_SPEAKER Default speaker voice ID (1=Paul, 0=Kate)

DFTTS_CODEPAGE code page (Win32) that the engine supports

DFTTS_DB_ACCESS_MODE Database access mode: File(0), RAM(1)

DFTTS_FIXED_POINT_SUPPORT Engine's integer simulation Yes(1)/No(0)

DFTTS_SAMPLING_FREQUENCY Sampling frequency of voice in Hz (8000, 11025, 16000)

DFTTS_MAX_PITCH_RATE Maximum pitch value

DFTTS_DEF_PITCH_RATE Default pitch value

DFTTS_MIN_PITCH_RATE Minimum pitch value

DFTTS_MAX_SPEED_RATE Maximum speed value

DFTTS_DEF_SPEED_RATE Default speed value
DFTTS_MIN_SPEED_RATE Minimum speed value
DFTTS_MAX_VOLUME Maximum volume
DFTTS_DEF_VOLUME Default volume
DFTTS_MIN_VOLUME Minimum volume
DFTTS_MAX_SENT_PAUSE Maximum sentence pause
DFTTS_DEF_SENT_PAUSE Default sentence pause
DFTTS_MIN_SENT_PAUSE Minimum sentence pause

iValue

Integer array that receives the result of *request* (see example)

szNeoSpeechLicFilePath

License Verification File path for the NeoSpeech VoiceText™ Engine.

Notes

Not all ttsParam are supported by all engines!

It can be used before loading the TTS DB.

It can be used to check errors when developing TTS-enabled applications by getting various information on what the engine supports.

Return Values

It returns DFTTS_PARAM_INFO_SUCCESS when the information is successfully drawn. It returns DFTTS_PARAM_INFO_ERROR_ENGINE_NOT_SUPPORTED if the specified by vetVType engine is not supported for the request.

Example

[C#]

```
DFTTSParamInfoReturnValue result;
int[] minPitch = new int[2];
int[] defPitch = new int[2];
int[] maxPitch = new int[2];

int[] minSpeed = new int[2];
int[] defSpeed = new int[2];
int[] maxSpeed = new int[2];

int[] minVolume = new int[2];
int[] defVolume = new int[2];
int[] maxVolume = new int[2];

VoiceEngineType vetVoiceEngineType = VoiceEngineType.NEOSPEECHVOICETEXT;

int lang = 1033;

result = (DFTTSParamInfoReturnValue)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSParamType.DFTTS_MIN_PITCH_RATE, minPitch, bNeoSpeechLicFilePath);

result = (DFTTSParamInfoReturnValue)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSParamType.DFTTS_DEF_PITCH_RATE, defPitch, bNeoSpeechLicFilePath);

result = (DFTTSParamInfoReturnValue)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSParamType.DFTTS_MAX_PITCH_RATE, maxPitch, bNeoSpeechLicFilePath);

result = (DFTTSParamInfoReturnValue)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSParamType.DFTTS_MIN_SPEED_RATE, minSpeed, bNeoSpeechLicFilePath);

result = (DFTTSParamInfoReturnValue)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSParamType.DFTTS_DEF_SPEED_RATE, defSpeed, bNeoSpeechLicFilePath);

result = (DFTTSParamInfoReturnValue)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSParamType.DFTTS_MAX_SPEED_RATE, maxSpeed, bNeoSpeechLicFilePath);
```

```

        result = (DFTTSPARAMINFORETURNVALUE)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSPARAMTYPE.DFTTS_MIN_VOLUME, minVolume, bNeoSpeechLicFilePath);

        result = (DFTTSPARAMINFORETURNVALUE)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSPARAMTYPE.DFTTS_DEF_VOLUME, defVolume, bNeoSpeechLicFilePath);

        result = (DFTTSPARAMINFORETURNVALUE)GetDFTTSEngineInfo((int)vetVoiceEngineType, lang,
(int)DFTTSPARAMTYPE.DFTTS_MAX_VOLUME, maxVolume, bNeoSpeechLicFilePath);

MessageBox.Show("MIN Engine Pitch: " + minPitch[0] + "\nDEFAULT Engine Pitch: " + defPitch[0] + "\nMAX
Engine Pitch: " + maxPitch[0] + "\nMIN Engine Speed: " + minSpeed[0] + "\nDEFAULT Engine Speed: " +
defSpeed[0] + "\nMAX Engine Speed: " + maxSpeed[0] + "\nMIN Engine Volume: " + minVolume[0] +
"\nDEFAULT Engine Volume: " + defVolume[0] + "\nMAX Engine Volume: " + maxVolume[0]);

```

[VB.NET]

```

Dim result As DFTTSPARAMINFORETURNVALUE
Dim minPitch(1) As Integer
Dim defPitch(1) As Integer
Dim maxPitch(1) As Integer

Dim minSpeed(1) As Integer
Dim defSpeed(1) As Integer
Dim maxSpeed(1) As Integer

Dim minVolume(1) As Integer
Dim defVolume(1) As Integer
Dim maxVolume(1) As Integer

Dim vetVoiceEngineType As VoiceEngineType = VoiceEngineType.NEOSPEECHVOICETEXT

Dim lang As Integer = 1033

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_MIN_PITCH_RATE,
minPitch, bNeoSpeechLicFilePath)

        Debug.WriteLine("Min pitch info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_DEF_PITCH_RATE,
defPitch, bNeoSpeechLicFilePath)

        Debug.WriteLine("Default pitch info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_MAX_PITCH_RATE,
maxPitch, bNeoSpeechLicFilePath)

        Debug.WriteLine("Max pitch info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_MIN_SPEED_RATE,
minSpeed, bNeoSpeechLicFilePath)

        Debug.WriteLine("Min speed info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_DEF_SPEED_RATE,
defSpeed, bNeoSpeechLicFilePath)

        Debug.WriteLine("Default speed info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_MAX_SPEED_RATE,
maxSpeed, bNeoSpeechLicFilePath)

        Debug.WriteLine("Max speed info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_MIN_VOLUME,
minVolume, bNeoSpeechLicFilePath)

        Debug.WriteLine("Min volume info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_DEF_VOLUME,
defVolume, bNeoSpeechLicFilePath)

        Debug.WriteLine("Default volume info return value: " & result & vbNewLine)

        result = GetDFTTSEngineInfo(vetVoiceEngineType, lang, DFTTSPARAMTYPE.DFTTS_MAX_VOLUME,
maxVolume, bNeoSpeechLicFilePath)

        Debug.WriteLine("Max volume info return value: " & result & vbNewLine)

```



```

        MsgBox("MIN Engine Pitch: " & minPitch(0) & vbNewLine & "DEFAULT Engine Pitch: " & defPitch(0)
        & vbNewLine & "MAX Engine Pitch: " & maxPitch(0) & vbNewLine & _
        "MIN Engine Speed: " & minSpeed(0) & vbNewLine & "DEFAULT Engine Speed: " & defSpeed(0) &
        vbNewLine & "MAX Engine Speed: " & maxSpeed(0) & vbNewLine & _
        "MIN Engine Volume: " & minVolume(0) & vbNewLine & "DEFAULT Engine Volume: " & defVolume(0) &
        vbNewLine & "MAX Engine Volume: " & maxVolume(0) )

```

GetDFTTSVoice

Returns information about an SDK-supported voice installed on the current system (voice name, engine, voice id, language id).

Synopsis

```

[C#]
[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int GetDFTTSVoice(int iVoiceInfoIdIn, ref int
pvetOut, byte[] szVoiceNameOut, ref int piVoiceNameLenOut, ref int
piVoiceIDOut, ref int plangOut);

[VB.NET]
<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function GetDFTTSVoice(ByVal iVoiceInfoIdIn As Integer, _
ByRef pvetOut As Integer, _
ByVal szVoiceNameOut() As Byte, _
ByRef piVoiceNameLenOut As Integer, _
ByRef piVoiceIDOut As Integer, _
ByRef plangOut As Integer _
) As Integer
End Function

```

Parameters

[IN] iVoiceInfoIdIn

Zero-based number of the voice info retrieval.

[OUT] pvetOut

Voice engine type of the retrieved voice (NEOSPEECHVOICETEXT, CEPSTRAL, ATTNV, MACOSXSPMAN or MSSAPI).

[IN] [OUT] szVoiceNameOut

ASCII c-string name of the voice. Must have sufficient size to hold the retrieved voice name.

[OUT] piVoiceNameLenOut

The length of szVoiceNameOut. Does not count the null-terminating character ('\0').

[OUT] piVoiceIDOut

Voice id of the retrieved voice. Only matters for NEOSPEECHVOICETEXT. Other voices will have an id of -1.

[OUT] plangOut

Language id of the retrieved voice. Only makes a difference right now for NEOSPEECHVOICETEXT.

Return Values

GET_DFTTS_VI_SUCCESS when the information is successfully retrieved.

GET_DFTTS_VI_NO_MORE_ITEMS when there is no voice information for the provided retrieval id (see sample).

GET_DFTTS_VI_ERROR_BUFFER_TOO_SMALL when the buffer szVoiceNameOut is too small to hold the retrieved voice name.

GET_DFTTS_VI_ERROR_OTHER other errors.

Example (see .NET samples)

```
[C#]
//This is how to list all installed supported voices

string szVoiceName = "*****";

byte[] bVoiceName2 = new byte[szVoiceName.Length];
ascii.GetBytes(szVoiceName.ToCharArray(), 0, szVoiceName.Length, bVoiceName2, 0);

int iVoiceNameLen;
iVoiceNameLen = 0;

int vet;
vet = 0;

int lang;
lang = 0;

int iVoiceID;
iVoiceID = 0;

int z;
z = 0;

int iUsedNumOfVoices;
iUsedNumOfVoices = 0;

while (0 == 0)
{
    DFTTSVoiceInfoReturnValue viresult;

    viresult = (DFTTSVoiceInfoReturnValue)GetDFTTSVoice(z, ref vet, bVoiceName2,
        ref iVoiceNameLen, ref iVoiceID, ref lang);

    //DO NOT USE MSSAPI NATIVELY FROM .NET! IT WILL CRASH THE APP!!!
    if (viresult == DFTTSVoiceInfoReturnValue.GET_DFTTS_VI_SUCCESS &&
        vet != (int)DFTTSVoiceEngineType.MSSAPI)
    {
        iUsedNumOfVoices++;

        string sVNameOut = ascii.GetString(bVoiceName2, 0, iVoiceNameLen);

        ComboBox1.DisplayMember = "sValue";

        cVoiceData oVData = new cVoiceData();
        oVData.vet = vet;
        oVData.lang = lang;
        oVData.sVoiceName = sVNameOut;
        oVData.iVoiceId = iVoiceID;

        ComboBox1.Items.Add(oVData);
    }
    else
    {
        break;
    }

    z++;
}

ComboBox1.SelectedIndex = 0;
```

```

if (iUsedNumOfVoices == 0)
{
    MessageBox.Show("No voices can be found on your system! Please download and install at least 1 " +
        "voice!");
}

[VB.NET]
'This is how to list all installed supported voices

Dim szVoiceName As String = "*****"

Dim bVoiceName2() As Byte = New Byte(szVoiceName.Length) {}
ascii.GetBytes(szVoiceName.ToCharArray(), 0, szVoiceName.Length, bVoiceName2, 0)

Dim iVoiceNameLen As Integer
iVoiceNameLen = 0

Dim vet As Integer
vet = 0

Dim lang As Integer
lang = 0

Dim iVoiceID As Integer
iVoiceID = 0

Dim z As Integer
z = 0

Dim iUsedNumOfVoices As Integer
iUsedNumOfVoices = 0

Do While 0 = 0

    Dim viresult As DFTTSVoiceInfoReturnValue

    viresult = GetDFTTSVoice(
        z, vet, bVoiceName2, iVoiceNameLen, _
        iVoiceID, lang)

    'DO NOT USE MSSAPI NATIVELY FROM .NET! IT WILL CRASH THE APP!!!
    If viresult = DFTTSVoiceInfoReturnValue.GET_DFTTS_VI_SUCCESS _
        And vet <> DFTTSVoiceEngineType.MSSAPI Then

        iUsedNumOfVoices = iUsedNumOfVoices + 1

        Dim sVNameOut As String = ascii.GetString(bVoiceName2, 0, iVoiceNameLen)

        Dim oVData As cVoiceData = New cVoiceData
        oVData.vet = vet
        oVData.lang = lang
        oVData.sVoiceName = sVNameOut
        oVData.iVoiceId = iVoiceID

        ComboBox1.Items.Add(oVData)

    Else

        Exit Do

    End If

    z = z + 1

Loop

ComboBox1.SelectedIndex = 0

If iUsedNumOfVoices = 0 Then
    MessageBox.Show("No voices can be found on your system! Please download and install at least 1 " & _
        "voice!")
End If

```

Appendix 1

.NET SDK Code Regions

[C#]

```
#region "API Enums and Constants"

public const int NEOSPEECH_NUM_VOICES = 12;

public const int NUM_ENGINE_INITIALIZATIONS = 16;

public enum DFTTSPredefinedSpeechParams
{
    DF_TTS_DEFAULT_PITCH = 100,
    ///*100%*/
    DF_TTS_DEFAULT_SPEED = 100,
    ///*100%*/
    DF_TTS_DEFAULT_VOLUME = 100,
    ///*100%*/
    DF_TTS_DEFAULT_PAUSE = 670,
    ///*670 msec*/
    DF_TTS_NEOSPEECH_MIN_PITCH = 50,
    DF_TTS_NEOSPEECH_MAX_PITCH = 200,
    DF_TTS_NEOSPEECH_MIN_SPEED = 50,
    DF_TTS_NEOSPEECH_MAX_SPEED = 400,
    DF_TTS_NEOSPEECH_MIN_VOLUME = 0,
    DF_TTS_NEOSPEECH_MAX_VOLUME = 500,
    DF_TTS_NEOSPEECH_MIN_PAUSE = 0,
    DF_TTS_NEOSPEECH_MAX_PAUSE = 20000,
    DF_TTS_CEPSTRAL_MIN_PITCH = 100,
    DF_TTS_CEPSTRAL_MAX_PITCH = 500,
    DF_TTS_CEPSTRAL_MIN_SPEED = 0,
    DF_TTS_CEPSTRAL_MAX_SPEED = 400,
    DF_TTS_CEPSTRAL_MIN_VOLUME = 0,
    DF_TTS_CEPSTRAL_MAX_VOLUME = 500,
    DF_TTS_ATTNV_MIN_SPEED = 13,
    DF_TTS_ATTNV_MAX_SPEED = 800,
    DF_TTS_ATTNV_MIN_VOLUME = 0,
    DF_TTS_ATTNV_MAX_VOLUME = 500,
    DF_TTS_ATTNV_MIN_PITCH = 0,
    ///*AT&T NV do not support this*/
    DF_TTS_ATTNV_MAX_PITCH = 0,
    ///*AT&T NV do not support this*/
    DF_TTS_MACOSXSPMAN_MIN_PITCH = 1,
    DF_TTS_MACOSXSPMAN_MAX_PITCH = 1000,
    DF_TTS_MACOSXSPMAN_MIN_SPEED = 1,
    DF_TTS_MACOSXSPMAN_MAX_SPEED = 1000,
    DF_TTS_MACOSXSPMAN_MIN_VOLUME = 100,
    DF_TTS_MACOSXSPMAN_MAX_VOLUME = 500,
    DF_TTS_MSSAPI_MIN_PITCH = 30,
    DF_TTS_MSSAPI_MAX_PITCH = 350,
    DF_TTS_MSSAPI_MIN_SPEED = 30,
    DF_TTS_MSSAPI_MAX_SPEED = 350,
    DF_TTS_MSSAPI_MIN_VOLUME = 0,
    DF_TTS_MSSAPI_MAX_VOLUME = 100
}

public enum DFTTSVoiceEngineType
{
    NOENGINE = -4,
    ALLENGINES = -3,
    NEOSPEECHVOICETEXT = 0,
    CEPSTRAL,
```

```

        ATTNV,
        MSSAPI,
        MACOSXSPMAN
    }

public enum InitDFTTSEngineReturnValue
{
    INIT_DFTTS_ENGINE_SUCCESS,
    INIT_DFTTS_ENGINE_ERROR_DB_PATH_DIFFERENT,
    INIT_DFTTS_ENGINE_ERROR_CHANNEL_MEM_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_MORPHEME_ANALYSIS_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_BREAK_INDEX_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_TEXT_PREP_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_ACOU_MODEL_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_UNIT_SEL_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_PROS_MODEL_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_SPEECH_DB_FAIL,
    INIT_DFTTS_ENGINE_ERROR_DB_PITCH_LOC_INFO_FAIL,
    INIT_DFTTS_ENGINE_ERROR_FAILED,
    INIT_DFTTS_ENGINE_ERROR_INVALIDARG,
    INIT_DFTTS_ENGINE_ERROR_OUTOFMEMORY,
    INIT_DFTTS_ENGINE_ERROR_NOTIMPL,
    INIT_DFTTS_ENGINE_ERROR_ABORT,
    INIT_DFTTS_ENGINE_ERROR_UNKNOWN,
    INIT_DFTTS_ENGINE_ERROR_BADHANDLE,
    INIT_DFTTS_ENGINE_ERROR_EXCEPTION,
    INIT_DFTTS_ENGINE_ERROR_EMPTY,
    INIT_DFTTS_ENGINE_ERROR_FULL,
    INIT_DFTTS_ENGINE_ERROR_INVALIDSTATE,
    INIT_DFTTS_ENGINE_ERROR_BADVERSION,
    INIT_DFTTS_ENGINE_ERROR_INSUFFICIENT_BUFFER,
    INIT_DFTTS_ENGINE_ERROR_UNSUPPORTED,
    INIT_DFTTS_ENGINE_ERROR_NOLICENSE,
    INIT_DFTTS_ENGINE_ERROR_CREATECHILDPROCESS_FAILED,
    INIT_DFTTS_ENGINE_ERROR_NOENVIRONMENTPATH,
    INIT_DFTTS_ENGINE_ERROR_TIMEOUT,
    INIT_DFTTS_ENGINE_ERROR_OUTOFRESOURCES,
    INIT_DFTTS_ENGINE_ERROR_NOVOICES,
    INIT_DFTTS_ENGINE_ERROR_CREATEFAIL,
    INIT_DFTTS_ENGINE_ERROR_CONNECTFAIL,
    INIT_DFTTS_ENGINE_ERROR_BINDFAIL,
    INIT_DFTTS_ENGINE_ERROR_LISTENFAIL,
    INIT_DFTTS_ENGINE_ERROR_CONNECTIONCLOSED,
    INIT_DFTTS_ENGINE_ERROR_ACCEPTFAIL,
    INIT_DFTTS_ENGINE_ERROR_SOCKETTIMEOUT,
    INIT_DFTTS_ENGINE_ERROR_SOCKETERROR,
    INIT_DFTTS_ENGINE_ERROR_NOMORESERVERS,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EWOULDBLOCK,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EINPROGRESS,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EALREADY,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOTSOCK,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EDESTADDRREQ,
    INIT_DFTTS_ENGINE_ERROR_SOCKET EMSGSIZE,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EPROTOTYPE,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOPROTOOPT,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EPROTONOSUPPORT,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ESOCKTNOSUPPORT,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EOPNOTSUPP,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EPFNOSUPPORT,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EAFNOSUPPORT,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EADDRINUSE,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EADDRNOTAVAIL,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ENETDOWN,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ENETUNREACH,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ENETRESET,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ECONNABORTED,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ECONNRESET,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOBUFS,
    INIT_DFTTS_ENGINE_ERROR_SOCKET_EISCONN,

```

```

        INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOTCONN,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_ESHUTDOWN,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_ETOOMANYREFS,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_ECONNREFUSED,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_ELOOP,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_ENAMETOOLONG,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_EHOSTDOWN,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_EHOSTUNREACH,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOTEMPTY,
        INIT_DFTTS_ENGINE_ERROR_SOCKET_EPROCLIM,
        INIT_DFTTS_ENGINE_ERROR_THREADSTARTED,
        INIT_DFTTS_ENGINE_ERROR_THREADNOTSTARTED,
        INIT_DFTTS_ENGINE_ERROR_THREADCOULDNOTCREATE,
        INIT_DFTTS_ENGINE_ERROR_BADNVFILE,
        INIT_DFTTS_ENGINE_ERROR_NOAUDIODRIVER,
        INIT_DFTTS_ENGINE_ERROR_DICTNOTFOUND,
        INIT_DFTTS_ENGINE_ERROR_OTHER
    }

    public enum UninitDFTTSEngineReturnValue
    {
        UNINIT_DFTTS_ENGINE_SUCCESS,
        UNINIT_DFTTS_ENGINE_FAIL
    }

    public enum LoadDFTTSUserDictReturnValue
    {
        LOAD_USER_DICT_SUCCESS,
        LOAD_USER_DICT_ERROR_DICTIDX_NOT_VALID,
        LOAD_USER_DICT_ERROR_DICT_ALREADY_LOADED,
        LOAD_USER_DICT_ERROR_DICT_WORD_ALREADY_LOADED,
        LOAD_USER_DICT_ERROR_NO_DICT_FILE_OR_ENTRY,
        LOAD_USER_DICT_ERROR_INVALID_DICT_FILE,
        LOAD_USER_DICT_ERROR_INVALID_VOICE,
        LOAD_USER_DICT_ERROR_INVALID_ARG,
        LOAD_USER_DICT_ERROR_INVALID_PHONEME_SET,
        LOAD_USER_DICT_ERROR_INVALID_PHONEME,
        LOAD_USER_DICT_ERROR_INVALID_PARTOFSP,
        LOAD_USER_DICT_ERROR_DENIED,
        LOAD_USER_DICT_ERROR_OUTOFMEMORY,
        LOAD_USER_DICT_ERROR_INTERNAL,
        LOAD_USER_DICT_ERROR_OTHER
    }

    public enum UnloadDFTTSUserDictReturnValue
    {
        UNLOAD_USER_DICT_SUCCESS,
        UNLOAD_USER_DICT_ERROR_DICTIDX_NOT_VALID,
        UNLOAD_USER_DICT_ERROR_DICT_UNLOADED,
        UNLOAD_USER_DICT_ERROR_NO_DICT_FILE_OR_ENTRY,
        UNLOAD_USER_DICT_ERROR_INVALID_PARTOFSP,
        UNLOAD_USER_DICT_ERROR_INVALID_PHONEME_SET,
        UNLOAD_USER_DICT_ERROR_INTERNAL,
        UNLOAD_USER_DICT_ERROR_OTHER
    }

    public enum DFTTSTextType
    {
        DFTTS_TEXT_TYPE_PLAIN,
        DFTTS_TEXT_TYPE_XML
    }

```

```

public enum DFTTSpeakReturnValue
{
    DFTTS_SPEAK_SUCCESS,
    DFTTS_SPEAK_ERROR_CHANNEL_MEM_FAIL,
    DFTTS_SPEAK_ERROR_TEXT_NULL,
    DFTTS_SPEAK_ERROR_TEXT_ZERO_LEN,
    DFTTS_SPEAK_ERROR_DB_NOT_LOADED,
    DFTTS_SPEAK_ERROR_SET_SOUND_CARD_FAIL,
    DFTTS_SPEAK_ERROR_UNIMPLEMENTED,
    DFTTS_SPEAK_ERROR_INTERNAL,
    DFTTS_SPEAK_ERROR_INVALID_PARAM,
    DFTTS_SPEAK_ERROR_INVALID_POINTER,
    DFTTS_SPEAK_ERROR_UNKNOWN_ENCODING,
    DFTTS_SPEAK_ERROR_OBJECT_NOT_FOUND,
    DFTTS_SPEAK_ERROR_INTERRUPTED,
    DFTTS_SPEAK_ERROR_INVALID_VOICE,
    DFTTS_SPEAK_ERROR_WRONG_EVENT,
    DFTTS_SPEAK_ERROR_ENGINE_INUSE,
    DFTTS_SPEAK_ERROR_NETWORK_ERROR,
    DFTTS_SPEAK_ERROR_INVALID_KEY,
    DFTTS_SPEAK_ERROR_QUEUE_FULL,
    DFTTS_SPEAK_ERROR_TOKEN_TIMEOUT,
    DFTTS_SPEAK_ERROR_FAILED,
    DFTTS_SPEAK_ERROR_INVALIDARG,
    DFTTS_SPEAK_ERROR_OUTOFMEMORY,
    DFTTS_SPEAK_ERROR_NOTIMPL,
    DFTTS_SPEAK_ERROR_ABORT,
    DFTTS_SPEAK_ERROR_UNKNOWN,
    DFTTS_SPEAK_ERROR_BADHANDLE,
    DFTTS_SPEAK_ERROR_EXCEPTION,
    DFTTS_SPEAK_ERROR_EMPTY,
    DFTTS_SPEAK_ERROR_FULL,
    DFTTS_SPEAK_ERROR_INVALIDSTATE,
    DFTTS_SPEAK_ERROR_BADVERSION,
    DFTTS_SPEAK_ERROR_INSUFFICIENT_BUFFER,
    DFTTS_SPEAK_ERROR_UNSUPPORTED,
    DFTTS_SPEAK_ERROR_NOLICENSE,
    DFTTS_SPEAK_ERROR_CREATECHILDPROCESS_FAILED,
    DFTTS_SPEAK_ERROR_NOENVIRONMENTPATH,
    DFTTS_SPEAK_ERROR_TIMEOUT,
    DFTTS_SPEAK_ERROR_OUTOFRESOURCES,
    DFTTS_SPEAK_ERROR_NOVOICES,
    DFTTS_SPEAK_ERROR_CREATEFAIL,
    DFTTS_SPEAK_ERROR_CONNECTFAIL,
    DFTTS_SPEAK_ERROR_BINDFAIL,
    DFTTS_SPEAK_ERROR_LISTENFAIL,
    DFTTS_SPEAK_ERROR_CONNECTIONCLOSED,
    DFTTS_SPEAK_ERROR_ACCEPTFAIL,
    DFTTS_SPEAK_ERROR_SOCKETTIMEOUT,
    DFTTS_SPEAK_ERROR_SOCKETERROR,
    DFTTS_SPEAK_ERROR_NOMORESERSERVERS,
    DFTTS_SPEAK_ERROR_SOCKET_EWOULDBLOCK,
    DFTTS_SPEAK_ERROR_SOCKET_EINPROGRESS,
    DFTTS_SPEAK_ERROR_SOCKET_EALREADY,
    DFTTS_SPEAK_ERROR_SOCKET_ENOTSOCK,
    DFTTS_SPEAK_ERROR_SOCKET_EDESTADDRREQ,
    DFTTS_SPEAK_ERROR_SOCKET EMSGSIZE,
    DFTTS_SPEAK_ERROR_SOCKET_EPROTOTYPE,
    DFTTS_SPEAK_ERROR_SOCKET_ENOPROTOPT,
    DFTTS_SPEAK_ERROR_SOCKET_EPROTONOSUPPORT,
    DFTTS_SPEAK_ERROR_SOCKET_ESOCKTNOSUPPORT,
    DFTTS_SPEAK_ERROR_SOCKET_EOPNOTSUPP,
    DFTTS_SPEAK_ERROR_SOCKET_EPFNOSUPPORT,
    DFTTS_SPEAK_ERROR_SOCKET_EAFNOSUPPORT,
    DFTTS_SPEAK_ERROR_SOCKET_EADDRINUSE,
    DFTTS_SPEAK_ERROR_SOCKET_EADDRNOTAVAIL,
    DFTTS_SPEAK_ERROR_SOCKET_ENETDOWN,
    DFTTS_SPEAK_ERROR_SOCKET_ENETUNREACH,
    DFTTS_SPEAK_ERROR_SOCKET_ENETRESET,

```

```

DFTTS_SPEAK_ERROR_SOCKET_ECONNABORTED,
DFTTS_SPEAK_ERROR_SOCKET_ECONNRESET,
DFTTS_SPEAK_ERROR_SOCKET_ENOBUFS,
DFTTS_SPEAK_ERROR_SOCKET_EISCONN,
DFTTS_SPEAK_ERROR_SOCKET_ENOTCONN,
DFTTS_SPEAK_ERROR_SOCKET_ESHUTDOWN,
DFTTS_SPEAK_ERROR_SOCKET_ETOOMANYREFS,
DFTTS_SPEAK_ERROR_SOCKET_ECONNREFUSED,
DFTTS_SPEAK_ERROR_SOCKET_ELOOP,
DFTTS_SPEAK_ERROR_SOCKET_ENAMETOOLONG,
DFTTS_SPEAK_ERROR_SOCKET_EHOSTDOWN,
DFTTS_SPEAK_ERROR_SOCKET_EHOSTUNREACH,
DFTTS_SPEAK_ERROR_SOCKET_ENOTEMPTY,
DFTTS_SPEAK_ERROR_SOCKET_EPROCLIM,
DFTTS_SPEAK_ERROR_THREADSTARTED,
DFTTS_SPEAK_ERROR_THREADNOTSTARTED,
DFTTS_SPEAK_ERROR_THREADCOULDNOTCREATE,
DFTTS_SPEAK_ERROR_BADNVFILE,
DFTTS_SPEAK_ERROR_NOAUDIODRIVER,
DFTTS_SPEAK_ERROR_DICTNOTFOUND,
DFTTS_SPEAK_ERROR_ALREADYPLAYING,
DFTTS_SPEAK_ERROR_AUDIOFORMATNOTSUPPORTED,
DFTTS_SPEAK_ERROR_XML_INVALID,
DFTTS_SPEAK_ERROR_WL_INVALID,
DFTTS_SPEAK_ERROR_INVALIDPHONESET,
DFTTS_SPEAK_ERROR_INVALIDPHONEME,
DFTTS_SPEAK_ERROR_MSGQ_CREATEFAILED,
DFTTS_SPEAK_ERROR_MSGQ_ALREADYEXISTS,
DFTTS_SPEAK_ERROR_MSGQ_NOTFOUND,
DFTTS_SPEAK_ERROR_MSGQ_INVALIDOP,
DFTTS_SPEAK_ERROR_MSGQ_NOTOPEN,
DFTTS_SPEAK_ERROR_MSGQ_LOCKFAILED,
DFTTS_SPEAK_ERROR_MSGQ_ABANDONED,
DFTTS_SPEAK_ERROR_MSGQ_OPENFAILED,
DFTTS_SPEAK_ERROR_OTHER
}

```

```

public enum DFTTSPauseReturnValue
{

```

```

    DFTTS_PAUSE_SUCCESS,
    DFTTS_PAUSE_ERROR_UNIMPLEMENTED,
    DFTTS_PAUSE_ERROR_INTERNAL,
    DFTTS_PAUSE_ERROR_INVALID_PARAM,
    DFTTS_PAUSE_ERROR_INVALID_POINTER,
    DFTTS_PAUSE_ERROR_OBJECT_NOT_FOUND,
    DFTTS_PAUSE_ERROR_UNKNOWN_ENCODING,
    DFTTS_PAUSE_ERROR_INTERRUPTED,
    DFTTS_PAUSE_ERROR_INVALID_VOICE,
    DFTTS_PAUSE_ERROR_WRONG_EVENT,
    DFTTS_PAUSE_ERROR_ENGINE_INUSE,
    DFTTS_PAUSE_ERROR_NETWORK_ERROR,
    DFTTS_PAUSE_ERROR_INVALID_KEY,
    DFTTS_PAUSE_ERROR_QUEUE_FULL,
    DFTTS_PAUSE_ERROR_TOKEN_TIMEOUT,
    DFTTS_PAUSE_ERROR_OTHER

```

```

}

```

```

public enum DFTTSResumeReturnValue
{

```

```

    DFTTS_RESUME_SUCCESS,
    DFTTS_RESUME_ERROR_UNIMPLEMENTED,
    DFTTS_RESUME_ERROR_INTERNAL,
    DFTTS_RESUME_ERROR_INVALID_PARAM,
    DFTTS_RESUME_ERROR_INVALID_POINTER,
    DFTTS_RESUME_ERROR_OBJECT_NOT_FOUND,

```


DFTTS_RESUME_ERROR_UNKNOWN_ENCODING,
DFTTS_RESUME_ERROR_INTERRUPTED,
DFTTS_RESUME_ERROR_INVALID_VOICE,
DFTTS_RESUME_ERROR_WRONG_EVENT,
DFTTS_RESUME_ERROR_ENGINE_INUSE,
DFTTS_RESUME_ERROR_NETWORK_ERROR,
DFTTS_RESUME_ERROR_INVALID_KEY,
DFTTS_RESUME_ERROR_QUEUE_FULL,
DFTTS_RESUME_ERROR_TOKEN_TIMEOUT,
DFTTS_RESUME_ERROR_FAILED,
DFTTS_RESUME_ERROR_INVALIDARG,
DFTTS_RESUME_ERROR_OUTOFMEMORY,
DFTTS_RESUME_ERROR_NOTIMPL,
DFTTS_RESUME_ERROR_ABORT,
DFTTS_RESUME_ERROR_UNKNOWN,
DFTTS_RESUME_ERROR_BADHANDLE,
DFTTS_RESUME_ERROR_EXCEPTION,
DFTTS_RESUME_ERROR_EMPTY,
DFTTS_RESUME_ERROR_FULL,
DFTTS_RESUME_ERROR_INVALIDSTATE,
DFTTS_RESUME_ERROR_BADVERSION,
DFTTS_RESUME_ERROR_INSUFFICIENT_BUFFER,
DFTTS_RESUME_ERROR_UNSUPPORTED,
DFTTS_RESUME_ERROR_NOLICENSE,
DFTTS_RESUME_ERROR_CREATECHILDPROCESS_FAILED,
DFTTS_RESUME_ERROR_NOENVIRONMENTPATH,
DFTTS_RESUME_ERROR_TIMEOUT,
DFTTS_RESUME_ERROR_OUTOFRESOURCES,
DFTTS_RESUME_ERROR_NOVOICES,
DFTTS_RESUME_ERROR_CREATEFAIL,
DFTTS_RESUME_ERROR_CONNECTFAIL,
DFTTS_RESUME_ERROR_BINDFAIL,
DFTTS_RESUME_ERROR_LISTENFAIL,
DFTTS_RESUME_ERROR_CONNECTIONCLOSED,
DFTTS_RESUME_ERROR_ACCEPTFAIL,
DFTTS_RESUME_ERROR_SOCKETTIMEOUT,
DFTTS_RESUME_ERROR_SOCKETERROR,
DFTTS_RESUME_ERROR_NOMORESERVERS,
DFTTS_RESUME_ERROR_SOCKET_EWOULDBLOCK,
DFTTS_RESUME_ERROR_SOCKET_EINPROGRESS,
DFTTS_RESUME_ERROR_SOCKET_EALREADY,
DFTTS_RESUME_ERROR_SOCKET_ENOTSOCK,
DFTTS_RESUME_ERROR_SOCKET_EDESTADDRREQ,
DFTTS_RESUME_ERROR_SOCKET EMSGSIZE,
DFTTS_RESUME_ERROR_SOCKET_EPROTOTYPE,
DFTTS_RESUME_ERROR_SOCKET_ENOPROTOOPT,
DFTTS_RESUME_ERROR_SOCKET_EPROTONOSUPPORT,
DFTTS_RESUME_ERROR_SOCKET_ESOCKTNOSUPPORT,
DFTTS_RESUME_ERROR_SOCKET_EOPNOTSUPP,
DFTTS_RESUME_ERROR_SOCKET_EPFNOSUPPORT,
DFTTS_RESUME_ERROR_SOCKET_EAFNOSUPPORT,
DFTTS_RESUME_ERROR_SOCKET_EADDRINUSE,
DFTTS_RESUME_ERROR_SOCKET_EADDRNOTAVAIL,
DFTTS_RESUME_ERROR_SOCKET_ENETDOWN,
DFTTS_RESUME_ERROR_SOCKET_ENETUNREACH,
DFTTS_RESUME_ERROR_SOCKET_ENETRESET,
DFTTS_RESUME_ERROR_SOCKET_ECONNABORTED,
DFTTS_RESUME_ERROR_SOCKET_ECONNRESET,
DFTTS_RESUME_ERROR_SOCKET_ENOBUFFS,
DFTTS_RESUME_ERROR_SOCKET_EISCONN,
DFTTS_RESUME_ERROR_SOCKET_ENOTCONN,
DFTTS_RESUME_ERROR_SOCKET_ESHUTDOWN,
DFTTS_RESUME_ERROR_SOCKET_ETOOMANYREFS,
DFTTS_RESUME_ERROR_SOCKET_ECONNREFUSED,
DFTTS_RESUME_ERROR_SOCKET_ELOOP,
DFTTS_RESUME_ERROR_SOCKET_ENAMETOOLONG,
DFTTS_RESUME_ERROR_SOCKET_EHOSTDOWN,
DFTTS_RESUME_ERROR_SOCKET_EHOSTUNREACH,
DFTTS_RESUME_ERROR_SOCKET_ENOTEMPTY,
DFTTS_RESUME_ERROR_SOCKET_EPROCLIM,
DFTTS_RESUME_ERROR_THREADSTARTED,

```

DFTTS_RESUME_ERROR_THREADNOTSTARTED,
DFTTS_RESUME_ERROR_THREADCOULDNOTCREATE,
DFTTS_RESUME_ERROR_BADNVFILE,
DFTTS_RESUME_ERROR_NOAUDIODRIVER,
DFTTS_RESUME_ERROR_DICTNOTFOUND,
DFTTS_RESUME_ERROR_ALREADYPLAYING,
DFTTS_RESUME_ERROR_AUDIOFORMATNOTSUPPORTED,
DFTTS_RESUME_ERROR_XML_INVALID,
DFTTS_RESUME_ERROR_WL_INVALID,
DFTTS_RESUME_ERROR_INVALIDPHONESET,
DFTTS_RESUME_ERROR_INVALIDPHONEME,
DFTTS_RESUME_ERROR_MSGQ_CREATEFAILED,
DFTTS_RESUME_ERROR_MSGQ_ALREADYEXISTS,
DFTTS_RESUME_ERROR_MSGQ_NOTFOUND,
DFTTS_RESUME_ERROR_MSGQ_INVALIDOP,
DFTTS_RESUME_ERROR_MSGQ_NOTOPEN,
DFTTS_RESUME_ERROR_MSGQ_LOCKFAILED,
DFTTS_RESUME_ERROR_MSGQ_ABANDONED,
DFTTS_RESUME_ERROR_MSGQ_OPENFAILED,
DFTTS_RESUME_ERROR_NOTHING_TO_RESUME,
DFTTS_RESUME_ERROR_OTHER

}

public enum DFTTSStopReturnValue
{
    DFTTS_STOP_SUCCESS,
    DFTTS_STOP_ERROR_INVALID_POINTER,
    DFTTS_STOP_ERROR_INVALID_ARG,
    DFTTS_STOP_ERROR_OUT_OF_MEMORY,
    DFTTS_STOP_ERROR_INVALID_FLAGS,
    DFTTS_STOP_ERROR_ENGINE_INUSE,
    DFTTS_STOP_ERROR_OTHER

}

public enum DFTTSExportReturnValue
{
    DFTTS_EXPORT_SUCCESS,
    DFTTS_EXPORT_ERROR_FORMAT_NOT_SUPPORTED,
    DFTTS_EXPORT_ERROR_CHANNEL_MEM_FAIL,
    DFTTS_EXPORT_ERROR_TEXT_NULL,
    DFTTS_EXPORT_ERROR_TEXT_ZERO_LEN,
    DFTTS_EXPORT_ERROR_DB_NOT_LOADED,
    DFTTS_EXPORT_ERROR_GEN_FILE_FAIL,
    DFTTS_EXPORT_ERROR_BUFFER_NULL,
    DFTTS_EXPORT_ERROR_THREAD_IN_USE,
    DFTTS_EXPORT_ERROR_UNIMPLEMENTED,
    DFTTS_EXPORT_ERROR_INTERNAL,
    DFTTS_EXPORT_ERROR_INVALID_PARAM,
    DFTTS_EXPORT_ERROR_INVALID_POINTER,
    DFTTS_EXPORT_ERROR_OBJECT_NOT_FOUND,
    DFTTS_EXPORT_ERROR_UNKNOWN_ENCODING,
    DFTTS_EXPORT_ERROR_INTERRUPTED,
    DFTTS_EXPORT_ERROR_INVALID_VOICE,
    DFTTS_EXPORT_ERROR_WRONG_EVENT,
    DFTTS_EXPORT_ERROR_ENGINE_INUSE,
    DFTTS_EXPORT_ERROR_NETWORK_ERROR,
    DFTTS_EXPORT_ERROR_INVALID_KEY,
    DFTTS_EXPORT_ERROR_QUEUE_FULL,
    DFTTS_EXPORT_ERROR_TOKEN_TIMEOUT,
    DFTTS_EXPORT_ERROR_FAILED,
    DFTTS_EXPORT_ERROR_INVALIDARG,
    DFTTS_EXPORT_ERROR_OUTOFMEMORY,
    DFTTS_EXPORT_ERROR_NOTIMPL,
    DFTTS_EXPORT_ERROR_ABORT,
    DFTTS_EXPORT_ERROR_UNKNOWN,
    DFTTS_EXPORT_ERROR_BADHANDLE,

```

DFTTS_EXPORT_ERROR_EXCEPTION,
DFTTS_EXPORT_ERROR_EMPTY,
DFTTS_EXPORT_ERROR_FULL,
DFTTS_EXPORT_ERROR_INVALIDSTATE,
DFTTS_EXPORT_ERROR_BADVERSION,
DFTTS_EXPORT_ERROR_INSUFFICIENT_BUFFER,
DFTTS_EXPORT_ERROR_UNSUPPORTED,
DFTTS_EXPORT_ERROR_NOLICENSE,
DFTTS_EXPORT_ERROR_CREATECHILDPROCESS_FAILED,
DFTTS_EXPORT_ERROR_NOENVIRONMENTPATH,
DFTTS_EXPORT_ERROR_TIMEOUT,
DFTTS_EXPORT_ERROR_OUTOFRESOURCES,
DFTTS_EXPORT_ERROR_NOVOICES,
DFTTS_EXPORT_ERROR_CREATEFAIL,
DFTTS_EXPORT_ERROR_CONNECTFAIL,
DFTTS_EXPORT_ERROR_BINDFAIL,
DFTTS_EXPORT_ERROR_LISTENFAIL,
DFTTS_EXPORT_ERROR_CONNECTIONCLOSED,
DFTTS_EXPORT_ERROR_ACCEPTFAIL,
DFTTS_EXPORT_ERROR_SOCKETTIMEOUT,
DFTTS_EXPORT_ERROR_SOCKETERROR,
DFTTS_EXPORT_ERROR_NOMORESERSVERS,
DFTTS_EXPORT_ERROR_SOCKET_EWOULDBLOCK,
DFTTS_EXPORT_ERROR_SOCKET_EINPROGRESS,
DFTTS_EXPORT_ERROR_SOCKET_EALREADY,
DFTTS_EXPORT_ERROR_SOCKET_ENOTSOCK,
DFTTS_EXPORT_ERROR_SOCKET_EDESTADDRREQ,
DFTTS_EXPORT_ERROR_SOCKET EMSGSIZE,
DFTTS_EXPORT_ERROR_SOCKET_EPROTOTYPE,
DFTTS_EXPORT_ERROR_SOCKET_ENOPROTOOPT,
DFTTS_EXPORT_ERROR_SOCKET_EPROTONOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_ESOCKTNOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_EOPNOTSUPP,
DFTTS_EXPORT_ERROR_SOCKET_EPFNOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_EAFNOSUPPORT,
DFTTS_EXPORT_ERROR_SOCKET_EADDRINUSE,
DFTTS_EXPORT_ERROR_SOCKET_EADDRNOTAVAIL,
DFTTS_EXPORT_ERROR_SOCKET_ENETDOWN,
DFTTS_EXPORT_ERROR_SOCKET_ENETUNREACH,
DFTTS_EXPORT_ERROR_SOCKET_ENETRESET,
DFTTS_EXPORT_ERROR_SOCKET_ECONNABORTED,
DFTTS_EXPORT_ERROR_SOCKET_ECONNRESET,
DFTTS_EXPORT_ERROR_SOCKET_ENOBUFS,
DFTTS_EXPORT_ERROR_SOCKET_EISCONN,
DFTTS_EXPORT_ERROR_SOCKET_ENOTCONN,
DFTTS_EXPORT_ERROR_SOCKET_ESHUTDOWN,
DFTTS_EXPORT_ERROR_SOCKET_ETOOMANYREFS,
DFTTS_EXPORT_ERROR_SOCKET_ECONNREFUSED,
DFTTS_EXPORT_ERROR_SOCKET_ELOOP,
DFTTS_EXPORT_ERROR_SOCKET_ENAMETOOLONG,
DFTTS_EXPORT_ERROR_SOCKET_EHOSTDOWN,
DFTTS_EXPORT_ERROR_SOCKET_EHOSTUNREACH,
DFTTS_EXPORT_ERROR_SOCKET_ENOTEMPTY,
DFTTS_EXPORT_ERROR_SOCKET_EPROCLIM,
DFTTS_EXPORT_ERROR_THREADSTARTED,
DFTTS_EXPORT_ERROR_THREADNOTSTARTED,
DFTTS_EXPORT_ERROR_THREADCOULDNOTCREATE,
DFTTS_EXPORT_ERROR_BADNVFILE,
DFTTS_EXPORT_ERROR_NOAUDIODRIVER,
DFTTS_EXPORT_ERROR_DICTNOTFOUND,
DFTTS_EXPORT_ERROR_ALREADYPLAYING,
DFTTS_EXPORT_ERROR_AUDIOFORMATNOTSUPPORTED,
DFTTS_EXPORT_ERROR_XML_INVALID,
DFTTS_EXPORT_ERROR_WL_INVALID,
DFTTS_EXPORT_ERROR_INVALIDPHONESET,
DFTTS_EXPORT_ERROR_INVALIDPHONEME,
DFTTS_EXPORT_ERROR_MSGQ_CREATEFAILED,
DFTTS_EXPORT_ERROR_MSGQ_ALREADYEXISTS,
DFTTS_EXPORT_ERROR_MSGQ_NOTFOUND,
DFTTS_EXPORT_ERROR_MSGQ_INVALIDOP,
DFTTS_EXPORT_ERROR_MSGQ_NOTOPEN,

```

DFTTS_EXPORT_ERROR_MSGQ_LOCKFAILED,
DFTTS_EXPORT_ERROR_MSGQ_ABANDONED,
DFTTS_EXPORT_ERROR_MSGQ_OPENFAILED,
DFTTS_EXPORT_ERROR_OTHER
}

public enum DFTTSVoiceInfoReturnValue
{
    GET_DFTTS_VI_SUCCESS,
    GET_DFTTS_VI_NO_MORE_ITEMS,
    GET_DFTTS_VI_ERROR_BUFFER_TOO_SMALL,
    GET_DFTTS_VI_ERROR_OTHER
}

//
// Language IDs.
//
// The following two combinations of primary language ID and
// sublanguage ID have special semantics:
//
// Primary Language ID    Sublanguage ID    Result
// -----
// LANG_NEUTRAL           SUBLANG_NEUTRAL    Language neutral
// LANG_NEUTRAL           SUBLANG_DEFAULT    User default language
// LANG_NEUTRAL           SUBLANG_SYS_DEFAULT System default language
// LANG_INVARIANT         SUBLANG_NEUTRAL    Invariant locale
//

//
// Primary language IDs.
//

public const int LANG_NEUTRAL = 0;
public const int LANG_INVARIANT = 127;

public const int LANG_AFIKAANS = 54;
public const int LANG_ALBANIAN = 28;
public const int LANG_ALSATIAN = 132;
public const int LANG_AMHARIC = 94;
public const int LANG_ARABIC = 1;
public const int LANG_ARMENIAN = 43;
public const int LANG_ASSAMESE = 77;
public const int LANG_AZERI = 44;
public const int LANG_BASHKIR = 109;
public const int LANG_BASQUE = 45;
public const int LANG_BELARUSIAN = 35;
public const int LANG_BENGALI = 69;
public const int LANG_BRETON = 126;
// Use with SUBLANG_BOSNIAN_* Sublanguage IDs
public const int LANG_BOSNIAN = 26;
// Use with the ConvertDefaultLocale function
public const int LANG_BOSNIAN_NEUTRAL = 30746;
public const int LANG_BULGARIAN = 2;
public const int LANG_CATALAN = 3;
// Use with SUBLANG_CHINESE_* Sublanguage IDs
public const int LANG_CHINESE = 4;
// Use with the ConvertDefaultLocale function
public const int LANG_CHINESE_SIMPLIFIED = 4;
// Use with the ConvertDefaultLocale function
public const int LANG_CHINESE_TRADITIONAL = 31748;
public const int LANG_CORSICAN = 131;
public const int LANG_CROATIAN = 26;
public const int LANG_CZECH = 5;
public const int LANG_DANISH = 6;
public const int LANG_DARI = 140;
public const int LANG_DIVEHI = 101;
public const int LANG_DUTCH = 19;
public const int LANG_ENGLISH = 9;

```

```

public const int LANG_ESTONIAN = 37;
public const int LANG_FAEROESE = 56;
// Deprecated: use LANG_PERSIAN instead
public const int LANG_FARSI = 41;
public const int LANG_FILIPINO = 100;
public const int LANG_FINNISH = 11;
public const int LANG_FRENCH = 12;
public const int LANG_FRISIAN = 98;
public const int LANG_GALICIAN = 86;
public const int LANG_GEORGIAN = 55;
public const int LANG_GERMAN = 7;
public const int LANG_GREEK = 8;
public const int LANG_GREENLANDIC = 111;
public const int LANG_GUJARATI = 71;
public const int LANG_HAUSA = 104;
public const int LANG_HEBREW = 13;
public const int LANG_HINDI = 57;
public const int LANG_HUNGARIAN = 14;
public const int LANG_ICELANDIC = 15;
public const int LANG_IGBO = 112;
public const int LANG_INDONESIAN = 33;
public const int LANG_INUKTITUT = 93;
// Use with the SUBLANG_IRISH_IRELAND Sublanguage ID
public const int LANG_IRISH = 60;
public const int LANG_ITALIAN = 16;
public const int LANG_JAPANESE = 17;
public const int LANG_KANNADA = 75;
public const int LANG_KASHMIRI = 96;
public const int LANG_KAZAK = 63;
public const int LANG_KHMER = 83;
public const int LANG_KICHE = 134;
public const int LANG_KINYARWANDA = 135;
public const int LANG_KONKANI = 87;
public const int LANG_KOREAN = 18;
public const int LANG_KYRGYZ = 64;
public const int LANG_LAO = 84;
public const int LANG_LATVIAN = 38;
public const int LANG_LITHUANIAN = 39;
public const int LANG_LOWER_SORBIAN = 46;
public const int LANG_LUXEMBOURGISH = 110;
// the Former Yugoslav Republic of Macedonia
public const int LANG_MACEDONIAN = 47;
public const int LANG_MALAY = 62;
public const int LANG_MALAYALAM = 76;
public const int LANG_MALTESE = 58;
public const int LANG_MANIPURI = 88;
public const int LANG_MAORI = 129;
public const int LANG_MAPUDUNGUN = 122;
public const int LANG_MARATHI = 78;
public const int LANG_MOHAWK = 124;
public const int LANG_MONGOLIAN = 80;
public const int LANG_NEPALI = 97;
public const int LANG_NORWEGIAN = 20;
public const int LANG_OCCITAN = 130;
public const int LANG_ORIYA = 72;
public const int LANG_PASHTO = 99;
public const int LANG_PERSIAN = 41;
public const int LANG_POLISH = 21;
public const int LANG_PORTUGUESE = 22;
public const int LANG_PUNJABI = 70;
public const int LANG_QUECHUA = 107;
public const int LANG_ROMANIAN = 24;
public const int LANG_ROMANSH = 23;
public const int LANG_RUSSIAN = 25;
public const int LANG_SAMI = 59;
public const int LANG_SANSKRIT = 79;
// Use with the SUBLANG_SERBIAN_* Sublanguage IDs
public const int LANG_SERBIAN = 26;
// Use with the ConvertDefaultLocale function
public const int LANG_SERBIAN_NEUTRAL = 31770;
public const int LANG_SINDHI = 89;

```

```

public const int LANG_SINHALESE = 91;
public const int LANG_SLOVAK = 27;
public const int LANG_SLOVENIAN = 36;
public const int LANG_SOTHO = 108;
public const int LANG_SPANISH = 10;
public const int LANG_SWAHILI = 65;
public const int LANG_SWEDISH = 29;
public const int LANG_SYRIAC = 90;
public const int LANG_TAJIK = 40;
public const int LANG_TAMAZIGHT = 95;
public const int LANG_TAMIL = 73;
public const int LANG_TATAR = 68;
public const int LANG_TELUGU = 74;
public const int LANG_THAI = 30;
public const int LANG_TIBETAN = 81;
public const int LANG_TIGRIGNA = 115;
public const int LANG_TSWANA = 50;
public const int LANG_TURKISH = 31;
public const int LANG_TURKMEN = 66;
public const int LANG_UGHUR = 128;
public const int LANG_UKRAINIAN = 34;
public const int LANG_UPPER_SORBIAN = 46;
public const int LANG_URDU = 32;
public const int LANG_UZBEK = 67;
public const int LANG_VIETNAMESE = 42;
public const int LANG_WELSH = 82;
public const int LANG_WOLOF = 136;
public const int LANG_XHOSA = 52;
public const int LANG_YAKUT = 133;
public const int LANG_YI = 120;
public const int LANG_YORUBA = 106;
public const int LANG_ZULU = 53;

//
// Sublanguage IDs.
//
// The name immediately following SUBLANG_ dictates which primary
// language ID that sublanguage ID can be combined with to form a
// valid language ID.
//

// language neutral
public const int SUBLANG_NEUTRAL = 0;
// user default
public const int SUBLANG_DEFAULT = 1;
// system default
public const int SUBLANG_SYS_DEFAULT = 2;
// default custom language/locale
public const int SUBLANG_CUSTOM_DEFAULT = 3;
// custom language/locale
public const int SUBLANG_CUSTOM_UNSPECIFIED = 4;
// Default custom MUI language/locale
public const int SUBLANG_UI_CUSTOM_DEFAULT = 5;

// Afrikaans (South Africa) = &H0436 af-ZA
public const int SUBLANG_AFRICAANS_SOUTH_AFRICA = 1;
// Albanian (Albania) = &H041c sq-AL
public const int SUBLANG_ALBANIAN_ALBANIA = 1;
// Alsatian (France) = &H0484
public const int SUBLANG_ALSATIAN_FRANCE = 1;
// Amharic (Ethiopia) = &H045e
public const int SUBLANG_AMHARIC_ETHIOPIA = 1;
// Arabic (Saudi Arabia)
public const int SUBLANG_ARABIC_SAUDI_ARABIA = 1;
// Arabic (Iraq)
public const int SUBLANG_ARABIC IRAQ = 2;
// Arabic (Egypt)
public const int SUBLANG_ARABIC_EGYPT = 3;
// Arabic (Libya)
public const int SUBLANG_ARABIC_LIBYA = 4;

```

```

// Arabic (Algeria)
public const int SUBLANG_ARABIC_ALGERIA = 5;
// Arabic (Morocco)
public const int SUBLANG_ARABIC_MOROCCO = 6;
// Arabic (Tunisia)
public const int SUBLANG_ARABIC_TUNISIA = 7;
// Arabic (Oman)
public const int SUBLANG_ARABIC_OMAN = 8;
// Arabic (Yemen)
public const int SUBLANG_ARABIC_YEMEN = 9;
// Arabic (Syria)
public const int SUBLANG_ARABIC_SYRIA = 10;
// Arabic (Jordan)
public const int SUBLANG_ARABIC_JORDAN = 11;
// Arabic (Lebanon)
public const int SUBLANG_ARABIC_LEBANON = 12;
// Arabic (Kuwait)
public const int SUBLANG_ARABIC_KUWAIT = 13;
// Arabic (U.A.E)
public const int SUBLANG_ARABIC_UAE = 14;
// Arabic (Bahrain)
public const int SUBLANG_ARABIC_BAHRAIN = 15;
// Arabic (Qatar)
public const int SUBLANG_ARABIC_QATAR = 16;
// Armenian (Armenia) = &H042b hy-AM
public const int SUBLANG_ARMENIAN_ARMENIA = 1;
// Assamese (India) = &H044d
public const int SUBLANG_ASSAMESE_INDIA = 1;
// Azeri (Latin)
public const int SUBLANG_AZERI_LATIN = 1;
// Azeri (Cyrillic)
public const int SUBLANG_AZERI_CYRILLIC = 2;
// Bashkir (Russia) = &H046d ba-RU
public const int SUBLANG_BASHKIR_RUSSIA = 1;
// Basque (Basque) = &H042d eu-ES
public const int SUBLANG_BASQUE_BASQUE = 1;
// Belarusian (Belarus) = &H0423 be-BY
public const int SUBLANG_BELARUSIAN_BELARUS = 1;
// Bengali (India)
public const int SUBLANG_BENGALI_INDIA = 1;
// Bengali (Bangladesh)
public const int SUBLANG_BENGALI_BANGLADESH = 2;
// Bosnian (Bosnia and Herzegovina - Latin) = &H141a bs-BA-Latn
public const int SUBLANG_BOSNIAN_BOSNIA_HERZEGOVINA_LATIN = 5;
// Bosnian (Bosnia and Herzegovina - Cyrillic) = &H201a bs-BA-Cyrl
public const int SUBLANG_BOSNIAN_BOSNIA_HERZEGOVINA_CYRILLIC = 8;
// Breton (France) = &H047e
public const int SUBLANG_BRETON_FRANCE = 1;
// Bulgarian (Bulgaria) = &H0402
public const int SUBLANG_BULGARIAN_BULGARIA = 1;
// Catalan (Catalan) = &H0403
public const int SUBLANG_CATALAN_CATALAN = 1;
// Chinese (Taiwan) = &H0404 zh-TW
public const int SUBLANG_CHINESE_TRADITIONAL = 1;
// Chinese (PR China) = &H0804 zh-CN
public const int SUBLANG_CHINESE_SIMPLIFIED = 2;
// Chinese (Hong Kong S.A.R., P.R.C.) = &H0c04 zh-HK
public const int SUBLANG_CHINESE_HONGKONG = 3;
// Chinese (Singapore) = &H1004 zh-SG
public const int SUBLANG_CHINESE_SINGAPORE = 4;
// Chinese (Macau S.A.R.) = &H1404 zh-MO
public const int SUBLANG_CHINESE_MACAU = 5;
// Corsican (France) = &H0483
public const int SUBLANG_CORSICAN_FRANCE = 1;
// Czech (Czech Republic) = &H0405
public const int SUBLANG_CZECH_CZECH_REPUBLIC = 1;
// Croatian (Croatia)
public const int SUBLANG_CROATIAN_CROATIA = 1;
// Croatian (Bosnia and Herzegovina - Latin) = &H101a hr-BA
public const int SUBLANG_CROATIAN_BOSNIA_HERZEGOVINA_LATIN = 4;
// Danish (Denmark) = &H0406

```

```

public const int SUBLANG_DANISH_DENMARK = 1;
// Dari (Afghanistan)
public const int SUBLANG_DARI_AFGHANISTAN = 1;
// Divehi (Maldives) = &H0465 div-MV
public const int SUBLANG_DIVEHI_MALDIVES = 1;
// Dutch
public const int SUBLANG_DUTCH = 1;
// Dutch (Belgian)
public const int SUBLANG_DUTCH_BELGIAN = 2;
// English (USA)
public const int SUBLANG_ENGLISH_US = 1;
// English (UK)
public const int SUBLANG_ENGLISH_UK = 2;
// English (Australian)
public const int SUBLANG_ENGLISH_AUS = 3;
// English (Canadian)
public const int SUBLANG_ENGLISH_CAN = 4;
// English (New Zealand)
public const int SUBLANG_ENGLISH_NZ = 5;
// English (Irish)
public const int SUBLANG_ENGLISH_EIRE = 6;
// English (South Africa)
public const int SUBLANG_ENGLISH_SOUTH_AFRICA = 7;
// English (Jamaica)
public const int SUBLANG_ENGLISH_JAMAICA = 8;
// English (Caribbean)
public const int SUBLANG_ENGLISH_CARIBBEAN = 9;
// English (Belize)
public const int SUBLANG_ENGLISH_BELIZE = 10;
// English (Trinidad)
public const int SUBLANG_ENGLISH_TRINIDAD = 11;
// English (Zimbabwe)
public const int SUBLANG_ENGLISH_ZIMBABWE = 12;
// English (Philippines)
public const int SUBLANG_ENGLISH_PHILIPPINES = 13;
// English (India)
public const int SUBLANG_ENGLISH_INDIA = 16;
// English (Malaysia)
public const int SUBLANG_ENGLISH_MALAYSIA = 17;
// English (Singapore)
public const int SUBLANG_ENGLISH_SINGAPORE = 18;
// Estonian (Estonia) = &H0425 et-EE
public const int SUBLANG_ESTONIAN_ESTONIA = 1;
// Faroese (Faroe Islands) = &H0438 fo-FO
public const int SUBLANG_FAEROESE_FAROE_ISLANDS = 1;
// Filipino (Philippines) = &H0464 fil-PH
public const int SUBLANG_FILIPINO_PHILIPPINES = 1;
// Finnish (Finland) = &H040b
public const int SUBLANG_FINNISH_FINLAND = 1;
// French
public const int SUBLANG_FRENCH = 1;
// French (Belgian)
public const int SUBLANG_FRENCH_BELGIAN = 2;
// French (Canadian)
public const int SUBLANG_FRENCH_CANADIAN = 3;
// French (Swiss)
public const int SUBLANG_FRENCH_SWISS = 4;
// French (Luxembourg)
public const int SUBLANG_FRENCH_LUXEMBOURG = 5;
// French (Monaco)
public const int SUBLANG_FRENCH_MONACO = 6;
// Frisian (Netherlands) = &H0462 fy-NL
public const int SUBLANG_FRISIAN_NETHERLANDS = 1;
// Galician (Galician) = &H0456 gl-ES
public const int SUBLANG_GALICIAN_GALICIAN = 1;
// Georgian (Georgia) = &H0437 ka-GE
public const int SUBLANG_GEORGIAN_GEORGIA = 1;
// German
public const int SUBLANG_GERMAN = 1;
// German (Swiss)
public const int SUBLANG_GERMAN_SWISS = 2;

```



```

// German (Austrian)
public const int SUBLANG_GERMAN_AUSTRIAN = 3;
// German (Luxembourg)
public const int SUBLANG_GERMAN_LUXEMBOURG = 4;
// German (Liechtenstein)
public const int SUBLANG_GERMAN_LIECHTENSTEIN = 5;
// Greek (Greece)
public const int SUBLANG_GREEK_GREECE = 1;
// Greenlandic (Greenland) = &H046f kl-GL
public const int SUBLANG_GREENLANDIC_GREENLAND = 1;
// Gujarati (India (Gujarati Script)) = &H0447 gu-IN
public const int SUBLANG_GUJARATI_INDIA = 1;
// Hausa (Latin, Nigeria) = &H0468 ha-NG-Latn
public const int SUBLANG_HAUSA_NIGERIA_LATIN = 1;
// Hebrew (Israel) = &H040d
public const int SUBLANG_HEBREW_ISRAEL = 1;
// Hindi (India) = &H0439 hi-IN
public const int SUBLANG_HINDI_INDIA = 1;
// Hungarian (Hungary) = &H040e
public const int SUBLANG_HUNGARIAN_HUNGARY = 1;
// Icelandic (Iceland) = &H040f
public const int SUBLANG_ICELANDIC_ICELAND = 1;
// Igbo (Nigeria) = &H0470 ig-NG
public const int SUBLANG_IGBO_NIGERIA = 1;
// Indonesian (Indonesia) = &H0421 id-ID
public const int SUBLANG_INDONESIAN_INDONESIA = 1;
// Inuktitut (Syllabics) (Canada) = &H045d iu-CA-Cans
public const int SUBLANG_INUKTITUT_CANADA = 1;
// Inuktitut (Canada - Latin)
public const int SUBLANG_INUKTITUT_CANADA_LATIN = 2;
// Irish (Ireland)
public const int SUBLANG_IRISH_IRELAND = 2;
// Italian
public const int SUBLANG_ITALIAN = 1;
// Italian (Swiss)
public const int SUBLANG_ITALIAN_SWISS = 2;
// Japanese (Japan) = &H0411
public const int SUBLANG_JAPANESE_JAPAN = 1;
// Kannada (India (Kannada Script)) = &H044b kn-IN
public const int SUBLANG_KANNADA_INDIA = 1;
// Kashmiri (South Asia)
public const int SUBLANG_KASHMIRI_SASIA = 2;
// For app compatibility only
public const int SUBLANG_KASHMIRI_INDIA = 2;
// Kazakh (Kazakhstan) = &H043f kk-KZ
public const int SUBLANG_KAZAK_KAZAKHSTAN = 1;
// Khmer (Cambodia) = &H0453 kh-KH
public const int SUBLANG_KHMER_CAMBODIA = 1;
// K'iche (Guatemala)
public const int SUBLANG_KICHE_GUATEMALA = 1;
// Kinyarwanda (Rwanda) = &H0487 rw-RW
public const int SUBLANG_KINYARWANDA_RWANDA = 1;
// Konkani (India) = &H0457 kok-IN
public const int SUBLANG_KONKANI_INDIA = 1;
// Korean (Extended Wansung)
public const int SUBLANG_KOREAN = 1;
// Kyrgyz (Kyrgyzstan) = &H0440 ky-KG
public const int SUBLANG_KYRGYZ_KYRGYZSTAN = 1;
// Lao (Lao PDR) = &H0454 lo-LA
public const int SUBLANG_LAO_LAO = 1;
// Latvian (Latvia) = &H0426 lv-LV
public const int SUBLANG_LATVIAN_LATVIA = 1;
// Lithuanian
public const int SUBLANG_LITHUANIAN = 1;
// Lower Sorbian (Germany) = &H082e wee-DE
public const int SUBLANG_LOWER_SORBIAN_GERMANY = 2;
// Luxembourgish (Luxembourg) = &H046e lb-LU
public const int SUBLANG_LUXEMBOURGISH_LUXEMBOURG = 1;
// Macedonian (Macedonia (FYROM)) = &H042f mk-MK
public const int SUBLANG_MACEDONIAN_MACEDONIA = 1;
// Malay (Malaysia)

```

```

public const int SUBLANG_MALAY_MALAYSIA = 1;
// Malay (Brunei Darussalam)
public const int SUBLANG_MALAY_BRUNEI_DARUSSALAM = 2;
// Malayalam (India (Malayalam Script)) = &H044c ml-IN
public const int SUBLANG_MALAYALAM_INDIA = 1;
// Maltese (Malta) = &H043a mt-MT
public const int SUBLANG_MALTESE_MALTA = 1;
// Maori (New Zealand) = &H0481 mi-NZ
public const int SUBLANG_MAORI_NEW_ZEALAND = 1;
// Mapudungun (Chile) = &H047a arn-CL
public const int SUBLANG_MAPUDUNGUN_CHILE = 1;
// Marathi (India) = &H044e mr-IN
public const int SUBLANG_MARATHI_INDIA = 1;
// Mohawk (Mohawk) = &H047c moh-CA
public const int SUBLANG_MOHAWK_MOHAWK = 1;
// Mongolian (Cyrillic, Mongolia)
public const int SUBLANG_MONGOLIAN_CYRILLIC_MONGOLIA = 1;
// Mongolian (PRC)
public const int SUBLANG_MONGOLIAN_PRC = 2;
// Nepali (India)
public const int SUBLANG_NEPALI_INDIA = 2;
// Nepali (Nepal) = &H0461 ne-NP
public const int SUBLANG_NEPALI_NEPAL = 1;
// Norwegian (Bokmal)
public const int SUBLANG_NORWEGIAN_BOKMAL = 1;
// Norwegian (Nynorsk)
public const int SUBLANG_NORWEGIAN_NYNORSK = 2;
// Occitan (France) = &H0482 oc-FR
public const int SUBLANG_OCCITAN_FRANCE = 1;
// Oriya (India (Oriya Script)) = &H0448 or-IN
public const int SUBLANG_ORIYA_INDIA = 1;
// Pashto (Afghanistan)
public const int SUBLANG_PASHTO_AFGHANISTAN = 1;
// Persian (Iran) = &H0429 fa-IR
public const int SUBLANG_PERSIAN_IRAN = 1;
// Polish (Poland) = &H0415
public const int SUBLANG_POLISH_POLAND = 1;
// Portuguese
public const int SUBLANG_PORTUGUESE = 2;
// Portuguese (Brazilian)
public const int SUBLANG_PORTUGUESE_BRAZILIAN = 1;
// Punjabi (India (Gurmukhi Script)) = &H0446 pa-IN
public const int SUBLANG_PUNJABI_INDIA = 1;
// Quechua (Bolivia)
public const int SUBLANG_QUECHUA_BOLIVIA = 1;
// Quechua (Ecuador)
public const int SUBLANG_QUECHUA_ECUADOR = 2;
// Quechua (Peru)
public const int SUBLANG_QUECHUA_PERU = 3;
// Romanian (Romania) = &H0418
public const int SUBLANG_ROMANIAN_ROMANIA = 1;
// Romansh (Switzerland) = &H0417 rm-CH
public const int SUBLANG_ROMANSH_SWITZERLAND = 1;
// Russian (Russia) = &H0419
public const int SUBLANG_RUSSIAN_RUSSIA = 1;
// Northern Sami (Norway)
public const int SUBLANG_SAMI_NORTHERN_NORWAY = 1;
// Northern Sami (Sweden)
public const int SUBLANG_SAMI_NORTHERN_SWEDEN = 2;
// Northern Sami (Finland)
public const int SUBLANG_SAMI_NORTHERN_FINLAND = 3;
// Lule Sami (Norway)
public const int SUBLANG_SAMI_LULE_NORWAY = 4;
// Lule Sami (Sweden)
public const int SUBLANG_SAMI_LULE_SWEDEN = 5;
// Southern Sami (Norway)
public const int SUBLANG_SAMI_SOUTHERN_NORWAY = 6;
// Southern Sami (Sweden)
public const int SUBLANG_SAMI_SOUTHERN_SWEDEN = 7;
// Skolt Sami (Finland)
public const int SUBLANG_SAMI_SKOLT_FINLAND = 8;

```

```

// Inari Sami (Finland)
public const int SUBLANG_SAMI_INARI_FINLAND = 9;
// Sanskrit (India) = &H044f sa-IN
public const int SUBLANG_SANSKRIT_INDIA = 1;
// Serbian (Bosnia and Herzegovina - Latin)
public const int SUBLANG_SERBIAN_BOSNIA_HERZEGOVINA_LATIN = 6;
// Serbian (Bosnia and Herzegovina - Cyrillic)
public const int SUBLANG_SERBIAN_BOSNIA_HERZEGOVINA_CYRILLIC = 7;
// Croatian (Croatia) = &H041a hr-HR
public const int SUBLANG_SERBIAN_CROATIA = 1;
// Serbian (Latin)
public const int SUBLANG_SERBIAN_LATIN = 2;
// Serbian (Cyrillic)
public const int SUBLANG_SERBIAN_CYRILLIC = 3;
// Sindhi (India) reserved = &H0459
public const int SUBLANG_SINDHI_INDIA = 1;
// Sindhi (Pakistan) reserved = &H0859
public const int SUBLANG_SINDHI_PAKISTAN = 2;
// For app compatibility only
public const int SUBLANG_SINDHI_AFGHANISTAN = 2;
// Sinhalese (Sri Lanka)
public const int SUBLANG_SINHALESE_SRI_LANKA = 1;
// Northern Sotho (South Africa)
public const int SUBLANG_SOTHO_NORTHERN_SOUTH_AFRICA = 1;
// Slovak (Slovakia) = &H041b sk-SK
public const int SUBLANG_SLOVAK_SLOVAKIA = 1;
// Slovenian (Slovenia) = &H0424 sl-SI
public const int SUBLANG_SLOVENIAN_SLOVENIA = 1;
// Spanish (Castilian)
public const int SUBLANG_SPANISH = 1;
// Spanish (Mexican)
public const int SUBLANG_SPANISH_MEXICAN = 2;
// Spanish (Modern)
public const int SUBLANG_SPANISH_MODERN = 3;
// Spanish (Guatemala)
public const int SUBLANG_SPANISH_GUATEMALA = 4;
// Spanish (Costa Rica)
public const int SUBLANG_SPANISH_COSTA_RICA = 5;
// Spanish (Panama)
public const int SUBLANG_SPANISH_PANAMA = 6;
// Spanish (Dominican Republic)
public const int SUBLANG_SPANISH_DOMINICAN_REPUBLIC = 7;
// Spanish (Venezuela)
public const int SUBLANG_SPANISH_VENEZUELA = 8;
// Spanish (Colombia)
public const int SUBLANG_SPANISH_COLOMBIA = 9;
// Spanish (Peru)
public const int SUBLANG_SPANISH_PERU = 10;
// Spanish (Argentina)
public const int SUBLANG_SPANISH_ARGENTINA = 11;
// Spanish (Ecuador)
public const int SUBLANG_SPANISH_ECUADOR = 12;
// Spanish (Chile)
public const int SUBLANG_SPANISH_CHILE = 13;
// Spanish (Uruguay)
public const int SUBLANG_SPANISH_URUGUAY = 14;
// Spanish (Paraguay)
public const int SUBLANG_SPANISH_PARAGUAY = 15;
// Spanish (Bolivia)
public const int SUBLANG_SPANISH_BOLIVIA = 16;
// Spanish (El Salvador)
public const int SUBLANG_SPANISH_EL_SALVADOR = 17;
// Spanish (Honduras)
public const int SUBLANG_SPANISH_HONDURAS = 18;
// Spanish (Nicaragua)
public const int SUBLANG_SPANISH_NICARAGUA = 19;
// Spanish (Puerto Rico)
public const int SUBLANG_SPANISH_PUERTO_RICO = 20;
// Spanish (United States)
public const int SUBLANG_SPANISH_US = 21;
// Swahili (Kenya) = &H0441 sw-KE

```

```

public const int SUBLANG_SWAHILI_KENYA = 1;
// Swedish
public const int SUBLANG_SWEDISH = 1;
// Swedish (Finland)
public const int SUBLANG_SWEDISH_FINLAND = 2;
// Syriac (Syria) = &H045a syr-SY
public const int SUBLANG_SYRIAC_SYRIA = 1;
// Tajik (Tajikistan) = &H0428 tg-TJ-Cyrl
public const int SUBLANG_TAJIK_TAJIKISTAN = 1;
// Tamazight (Latin, Algeria) = &H085f tmz-DZ-Latn
public const int SUBLANG_TAMAZIGHT_ALGERIA_LATIN = 2;
// Tamil (India)
public const int SUBLANG_TAMIL_INDIA = 1;
// Tatar (Russia) = &H0444 tt-RU
public const int SUBLANG_TATAR_RUSSIA = 1;
// Telugu (India (Telugu Script)) = &H044a te-IN
public const int SUBLANG_TELUGU_INDIA = 1;
// Thai (Thailand) = &H041e th-TH
public const int SUBLANG_THAI_THAILAND = 1;
// Tibetan (PRC)
public const int SUBLANG_TIBETAN_PRC = 1;
// Tigrigna (Eritrea)
public const int SUBLANG_TIGRIGNA_ERITREA = 2;
// Setswana / Tswana (South Africa) = &H0432 tn-ZA
public const int SUBLANG_TSWANA_SOUTH_AFRICA = 1;
// Turkish (Turkey) = &H041f tr-TR
public const int SUBLANG_TURKISH_TURKEY = 1;
// Turkmen (Turkmenistan) = &H0442 tk-TM
public const int SUBLANG_TURKMEN_TURKMENISTAN = 1;
// Uighur (PRC) = &H0480 ug-CN
public const int SUBLANG_UGHUR_PRC = 1;
// Ukrainian (Ukraine) = &H0422 uk-UA
public const int SUBLANG_UKRAINIAN_UKRAINE = 1;
// Upper Sorbian (Germany) = &H042e wen-DE
public const int SUBLANG_UPPER_SORBIAN_GERMANY = 1;
// Urdu (Pakistan)
public const int SUBLANG_URDU_PAKISTAN = 1;
// Urdu (India)
public const int SUBLANG_URDU_INDIA = 2;
// Uzbek (Latin)
public const int SUBLANG_UZBEK_LATIN = 1;
// Uzbek (Cyrillic)
public const int SUBLANG_UZBEK_CYRILLIC = 2;
// Vietnamese (Vietnam) = &H042a vi-VN
public const int SUBLANG_VIETNAMESE_VIETNAM = 1;
// Welsh (United Kingdom) = &H0452 cy-GB
public const int SUBLANG_WELSH_UNITED_KINGDOM = 1;
// Wolof (Senegal)
public const int SUBLANG_WOLOF_SENEGAL = 1;
// isiXhosa / Xhosa (South Africa) = &H0434 xh-ZA
public const int SUBLANG_XHOSA_SOUTH_AFRICA = 1;
// Yakut (Russia) = &H0485 sah-RU
public const int SUBLANG_YAKUT_RUSSIA = 1;
// Yi (PRC) = &H0478
public const int SUBLANG_YI_PRC = 1;
// Yoruba (Nigeria) 046a yo-NG
public const int SUBLANG_YORUBA_NIGERIA = 1;
// isiZulu / Zulu (South Africa) = &H0435 zu-ZA
public const int SUBLANG_ZULU_SOUTH_AFRICA = 1;

//
// A language ID is a 16 bit value which is the combination of a
// primary language ID and a secondary language ID. The bits are
// allocated as follows:
//
//
//      +-----+-----+
//      | Sublanguage ID | Primary Language ID |
//      +-----+-----+
//      15             10 9             0 bit
//

```

```

//Neutral language

public const int LANG_SUBLANG_NEUTRAL = 0;

public enum NeoSpeechVoiceTextAudioFormat
{
    VT_FILE_API_FMT_S16PCM = 0,
    VT_FILE_API_FMT_ALAW = 1,
    VT_FILE_API_FMT_MULAW = 2,
    VT_FILE_API_FMT_DADPCM = 3,
    VT_FILE_API_FMT_S16PCM_WAVE = 4,
    VT_FILE_API_FMT_U08PCM_WAVE = 5,
    VT_FILE_API_FMT_IMA_WAVE = 6,
    VT_FILE_API_FMT_ALAW_WAVE = 7,
    VT_FILE_API_FMT_MULAW_WAVE = 8,
    VT_FILE_API_FMT_MULAW_AU = 9
}

//Export formats for Cepstral voices

//You need to set the following instead:

//szAudioEncoding [possible values:
// * "riff": Microsoft RIFF (WAV) file
// * "pcm16", "pcm8" PCM 16 bit/8 bit WAV
// * "ulaw" - u-Law (8-bit by definition), "alaw" - A-Law (8-bit by definition)
// * "snd": Sun/NeXT .au (SND) format.
// * "raw": unheadered audio data, native byte order
// * "le": unheadered audio data, little-endian (LSB first)
// * "be": unheadered audio data, big-endian (MSB first)
// * NOTE: that not all encoding types may be supported by all formats. For
instance, SND doesn't support A-Law. ], _
//iAudioSamplingRate [possible values: 8000 (8 KHz), 16000 (16 KHz), 11025
(11.025 kHz), etc ], _
//iAudioChannels [possible values: 1 (mono), 2 (stereo)]

//Export formats for AT&T NV (INTERNAL USE ONLY) _
//Currently only PCM WAV supported (all audio format arguments will be ignored)

public enum NeoSpeechVoiceID
{
    NEOSPEECH_KATE_ENG = 0,

    NEOSPEECH_PAUL_ENG = 1,

    NEOSPEECH_MIYU_JPN = 0,

    NEOSPEECH_SHOW_JPN = 1,

    NEOSPEECH_MISAKI_JPN = 2,

    NEOSPEECH_LILY_CHI = 0,

    NEOSPEECH_WANG_CHI = 1,

    NEOSPEECH_JUNWOO_KOR = 3,

    NEOSPEECH_SUJIN_KOR = 8,

```

```

        NEOSPEECH_YUMI_KOR = 10,

        NEOSPEECH_GYURI_KOR = 11,

        NEOSPEECH_DAYOUNG_KOR = 12

    }

    public const string NEOSPEECH_KATE_ENG_NAMESTR = "Kate";
    public const string NEOSPEECH_PAUL_ENG_NAMESTR = "Paul";

    public const string NEOSPEECH_MIYU_JPN_NAMESTR = "Miyu";
    public const string NEOSPEECH_SHOW_JPN_NAMESTR = "Show";
    public const string NEOSPEECH_MISAKI_JPN_NAMESTR = "Misaki";

    public const string NEOSPEECH_LILY_CHI_NAMESTR = "Lily";
    public const string NEOSPEECH_WANG_CHI_NAMESTR = "Wang";

    public const string NEOSPEECH_JUNWOO_KOR_NAMESTR = "Junwoo";
    public const string NEOSPEECH_SUJIN_KOR_NAMESTR = "Sujin";
    public const string NEOSPEECH_YUMI_KOR_NAMESTR = "Yumi";
    public const string NEOSPEECH_GYURI_KOR_NAMESTR = "Gyuri";
    public const string NEOSPEECH_DAYOUNG_KOR_NAMESTR = "Dayoung";

    //Messages used for speech events
    public const int WM_USER = 1024;

    public const int WIN_SPEAK_MSG = WM_USER + 12359;

    public const int WIN_EXPORT_MSG = WM_USER + 12360;

    #endregion

    #region "P/Invoke declarations"

    [DllImport("dftts.dll", CharSet = CharSet.Auto)]
    public static extern int GetDFTTSVoice(int iVoiceInfoIdIn, ref int pvoutOut,
    byte[] szVoiceNameOut, ref int piVoiceNameLenOut, ref int piVoiceIDOut, ref int plangOut);

    [DllImport("dftts.dll", CharSet = CharSet.Auto)]
    public static extern void InitDFTTSEngineEx3(IntPtr hwndWinOwner,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathKate,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathPaul,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathMiyu,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathShow,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathMisaki,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathLily,
    [MarshalAs(UnmanagedType.LPStr)] string
    szNeoSpeechDBFolderPathWang,

```

```

        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechDBFolderPathJunwoo,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechDBFolderPathSujin,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechDBFolderPathYumi,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechDBFolderPathGyuri,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechDBFolderPathDayoung,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathKate,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathPaul,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathMiyu,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathShow,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechLicFilePathMisaki,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathLily,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathWang,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechLicFilePathJunwoo,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechLicFilePathSujin,
        [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePathYumi,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechLicFilePathGyuri,
        [MarshalAs(UnmanagedType.LPStr)] string
szNeoSpeechLicFilePathDayoung,
        int[] psiLoadedEngines,
        int[] psiLoadedEnginesReturnValues
    );

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int UninitDFTTSEngine();

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int MakeLanguage(int mainlang, int sublang);

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int GetMainLanguage(int lang);

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int GetSubLanguage(int lang);

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int LoadDFTTSUserDict(int iDictIndex,
        [MarshalAs(UnmanagedType.LPStr)] string szDictName,
        [MarshalAs(UnmanagedType.LPStr)] string szDictFileName, int vet, int lang,
        [MarshalAs(UnmanagedType.LPStr)] string szVoiceName,
        [MarshalAs(UnmanagedType.LPStr)] string szDictContents);

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int UnloadDFTTSUserDict(int iDictIndex,
        [MarshalAs(UnmanagedType.LPStr)] string szDictName,
        int vet, int lang);

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSSpeak(IntPtr hwndWinOwner, int vet,
        [MarshalAs(UnmanagedType.LPStr)] string szVoiceName,
        int iVoiceID, int lang,
        [MarshalAs(UnmanagedType.LPStr)] string szText,
        int iPitch,
        int iSpeed, int iVolume, int iPause, int iDictID, int ttTextType, int
ofOutPutFormat);

```

```

[DllImport("dftts.dll", CharSet = CharSet.Auto)]

```

```

public static extern int DFTTSPause(int vet, int lang);

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSResume(int vet, int lang);

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSStop(int vet, int lang);

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int DFTTSExportToFileEx(int vet,
    [MarshalAs(UnmanagedType.LPStr)] string szVoiceName,
    int iVoiceID, int lang,
    [MarshalAs(UnmanagedType.LPStr)] string szText,
    int iPitch, int iSpeed,
    int iVolume, int iPause, int iDictID, int ttTextType,
    [MarshalAs(UnmanagedType.LPStr)] string szFilePath,
    int ffFileFormat,
    [MarshalAs(UnmanagedType.LPStr)] string szAudioEncoding,
    int iAudioSamplingRate,
    int iAudioChannels);

[DllImport("dftts.dll", CharSet = CharSet.Auto)]
public static extern int GetDFTTSEngineInfo(int vetVType, int lang, int
ttspParam, int[] iValue, [MarshalAs(UnmanagedType.LPStr)] string szNeoSpeechLicFilePath);

```

```

#endregion

```

```

[VB.NET]

```

```

#Region "API Enums and Constants"

```

```

Public Const NEOSPEECH_NUM_VOICES As Integer = 12

```

```

Public Const NUM_ENGINE_INITIALIZATIONS As Integer = 16

```

```

Public Enum DFTTSPredefinedSpeechParams

```

```

    DF_TTS_DEFAULT_PITCH = 100 '/*100%*/
    DF_TTS_DEFAULT_SPEED = 100 '/*100%*/
    DF_TTS_DEFAULT_VOLUME = 100 '/*100%*/
    DF_TTS_DEFAULT_PAUSE = 670 '/*670 msec*/
    DF_TTS_NEOSPEECH_MIN_PITCH = 50
    DF_TTS_NEOSPEECH_MAX_PITCH = 200
    DF_TTS_NEOSPEECH_MIN_SPEED = 50
    DF_TTS_NEOSPEECH_MAX_SPEED = 400
    DF_TTS_NEOSPEECH_MIN_VOLUME = 0
    DF_TTS_NEOSPEECH_MAX_VOLUME = 500
    DF_TTS_NEOSPEECH_MIN_PAUSE = 0
    DF_TTS_NEOSPEECH_MAX_PAUSE = 20000
    DF_TTS_CEPSTRAL_MIN_PITCH = 100
    DF_TTS_CEPSTRAL_MAX_PITCH = 500
    DF_TTS_CEPSTRAL_MIN_SPEED = 0
    DF_TTS_CEPSTRAL_MAX_SPEED = 400
    DF_TTS_CEPSTRAL_MIN_VOLUME = 0
    DF_TTS_CEPSTRAL_MAX_VOLUME = 500
    DF_TTS_ATTNV_MIN_SPEED = 13
    DF_TTS_ATTNV_MAX_SPEED = 800
    DF_TTS_ATTNV_MIN_VOLUME = 0
    DF_TTS_ATTNV_MAX_VOLUME = 500
    DF_TTS_ATTNV_MIN_PITCH = 0 '/*AT&T NV do not support this*/
    DF_TTS_ATTNV_MAX_PITCH = 0 '/*AT&T NV do not support this*/
    DF_TTS_MACOSXSPMAN_MIN_PITCH = 1
    DF_TTS_MACOSXSPMAN_MAX_PITCH = 1000
    DF_TTS_MACOSXSPMAN_MIN_SPEED = 1
    DF_TTS_MACOSXSPMAN_MAX_SPEED = 1000
    DF_TTS_MACOSXSPMAN_MIN_VOLUME = 100
    DF_TTS_MACOSXSPMAN_MAX_VOLUME = 500
    DF_TTS_MSSAPI_MIN_PITCH = 30
    DF_TTS_MSSAPI_MAX_PITCH = 350
    DF_TTS_MSSAPI_MIN_SPEED = 30
    DF_TTS_MSSAPI_MAX_SPEED = 350

```



```
DF_TTS_MSSAPI_MIN_VOLUME = 0
DF_TTS_MSSAPI_MAX_VOLUME = 100
```

End Enum

Public Enum DFTTSVoiceEngineType

```
NOENGINE = -4
ALLENGINES = -3
NEOSPEECHVOICETEXT = 0
CEPSTRAL
ATTNV
MSSAPI
MACOSXSPMAN
```

End Enum

Public Enum InitDFTTSEngineReturnValue

```
INIT_DFTTS_ENGINE_SUCCESS
INIT_DFTTS_ENGINE_ERROR_DB_PATH_DIFFERENT
INIT_DFTTS_ENGINE_ERROR_CHANNEL_MEM_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_MORPHEME_ANALYSIS_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_BREAK_INDEX_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_TEXT_PREP_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_ACOU_MODEL_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_UNIT_SEL_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_PROS_MODEL_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_SPEECH_DB_FAIL
INIT_DFTTS_ENGINE_ERROR_DB_PITCH_LOC_INFO_FAIL
INIT_DFTTS_ENGINE_ERROR_FAILED
INIT_DFTTS_ENGINE_ERROR_INVALIDARG
INIT_DFTTS_ENGINE_ERROR_OUTOFMEMORY
INIT_DFTTS_ENGINE_ERROR_NOTIMPL
INIT_DFTTS_ENGINE_ERROR_ABORT
INIT_DFTTS_ENGINE_ERROR_UNKNOWN
INIT_DFTTS_ENGINE_ERROR_BADHANDLE
INIT_DFTTS_ENGINE_ERROR_EXCEPTION
INIT_DFTTS_ENGINE_ERROR_EMPTY
INIT_DFTTS_ENGINE_ERROR_FULL
INIT_DFTTS_ENGINE_ERROR_INVALIDSTATE
INIT_DFTTS_ENGINE_ERROR_BADVERSION
INIT_DFTTS_ENGINE_ERROR_INSUFFICIENT_BUFFER
INIT_DFTTS_ENGINE_ERROR_UNSUPPORTED
INIT_DFTTS_ENGINE_ERROR_NOLICENSE
INIT_DFTTS_ENGINE_ERROR_CREATECHILDPROCESS_FAILED
INIT_DFTTS_ENGINE_ERROR_NOENVIRONMENTPATH
INIT_DFTTS_ENGINE_ERROR_TIMEOUT
INIT_DFTTS_ENGINE_ERROR_OUTOFRESOURCES
INIT_DFTTS_ENGINE_ERROR_NOVOICES
INIT_DFTTS_ENGINE_ERROR_CREATEFAIL
INIT_DFTTS_ENGINE_ERROR_CONNECTFAIL
INIT_DFTTS_ENGINE_ERROR_BINDFAIL
INIT_DFTTS_ENGINE_ERROR_LISTENFAIL
INIT_DFTTS_ENGINE_ERROR_CONNECTIONCLOSED
INIT_DFTTS_ENGINE_ERROR_ACCEPTFAIL
INIT_DFTTS_ENGINE_ERROR_SOCKETTIMEOUT
INIT_DFTTS_ENGINE_ERROR_SOCKETERROR
INIT_DFTTS_ENGINE_ERROR_NOMORESERVERS
INIT_DFTTS_ENGINE_ERROR_SOCKET_EWOULDBLOCK
INIT_DFTTS_ENGINE_ERROR_SOCKET_EINPROGRESS
INIT_DFTTS_ENGINE_ERROR_SOCKET_EALREADY
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOTSOCK
INIT_DFTTS_ENGINE_ERROR_SOCKET_EDESTADDRREQ
INIT_DFTTS_ENGINE_ERROR_SOCKET EMSGSIZE
INIT_DFTTS_ENGINE_ERROR_SOCKET_EPROTOTYPE
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOPROTOOPT
INIT_DFTTS_ENGINE_ERROR_SOCKET_EPROTONOSUPPORT
INIT_DFTTS_ENGINE_ERROR_SOCKET_ESOCKTNOSUPPORT
INIT_DFTTS_ENGINE_ERROR_SOCKET_EOPNOTSUPP
INIT_DFTTS_ENGINE_ERROR_SOCKET_EPFNOSUPPORT
```

```

INIT_DFTTS_ENGINE_ERROR_SOCKET_EAFNOSUPPORT
INIT_DFTTS_ENGINE_ERROR_SOCKET_EADDRINUSE
INIT_DFTTS_ENGINE_ERROR_SOCKET_EADDRNOTAVAIL
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENETDOWN
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENETUNREACH
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENETRESET
INIT_DFTTS_ENGINE_ERROR_SOCKET_ECONNABORTED
INIT_DFTTS_ENGINE_ERROR_SOCKET_ECONNRESET
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOBUFS
INIT_DFTTS_ENGINE_ERROR_SOCKET_EISCONN
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOTCONN
INIT_DFTTS_ENGINE_ERROR_SOCKET_ESHUTDOWN
INIT_DFTTS_ENGINE_ERROR_SOCKET_ETOOMANYREFS
INIT_DFTTS_ENGINE_ERROR_SOCKET_ECONNREFUSED
INIT_DFTTS_ENGINE_ERROR_SOCKET_ELOOP
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENAMETOOLONG
INIT_DFTTS_ENGINE_ERROR_SOCKET_EHOSTDOWN
INIT_DFTTS_ENGINE_ERROR_SOCKET_EHOSTUNREACH
INIT_DFTTS_ENGINE_ERROR_SOCKET_ENOTEMPTY
INIT_DFTTS_ENGINE_ERROR_SOCKET_EPROCLIM
INIT_DFTTS_ENGINE_ERROR_THREADSTARTED
INIT_DFTTS_ENGINE_ERROR_THREADNOTSTARTED
INIT_DFTTS_ENGINE_ERROR_THREADCOULDNOTCREATE
INIT_DFTTS_ENGINE_ERROR_BADNVFILE
INIT_DFTTS_ENGINE_ERROR_NOAUDIODRIVER
INIT_DFTTS_ENGINE_ERROR_DICTNOTFOUND
INIT_DFTTS_ENGINE_ERROR_OTHER

```

End Enum

Public Enum UninitDFTTSEngineReturnValue

```

UNINIT_DFTTS_ENGINE_SUCCESS
UNINIT_DFTTS_ENGINE_FAIL

```

End Enum

Public Enum LoadDFTTSUserDictReturnValue

```

LOAD_USER_DICT_SUCCESS
LOAD_USER_DICT_ERROR_DICTIDX_NOT_VALID
LOAD_USER_DICT_ERROR_DICT_ALREADY_LOADED
LOAD_USER_DICT_ERROR_DICT_WORD_ALREADY_LOADED
LOAD_USER_DICT_ERROR_NO_DICT_FILE_OR_ENTRY
LOAD_USER_DICT_ERROR_INVALID_DICT_FILE
LOAD_USER_DICT_ERROR_INVALID_VOICE
LOAD_USER_DICT_ERROR_INVALID_ARG
LOAD_USER_DICT_ERROR_INVALID_PHONEME_SET
LOAD_USER_DICT_ERROR_INVALID_PHONEME
LOAD_USER_DICT_ERROR_INVALID_PARTOFSP
LOAD_USER_DICT_ERROR_DENIED
LOAD_USER_DICT_ERROR_OUTOFMEMORY
LOAD_USER_DICT_ERROR_INTERNAL
LOAD_USER_DICT_ERROR_OTHER

```

End Enum

Public Enum UnloadDFTTSUserDictReturnValue

```

UNLOAD_USER_DICT_SUCCESS
UNLOAD_USER_DICT_ERROR_DICTIDX_NOT_VALID
UNLOAD_USER_DICT_ERROR_DICT_UNLOADED
UNLOAD_USER_DICT_ERROR_NO_DICT_FILE_OR_ENTRY
UNLOAD_USER_DICT_ERROR_INVALID_PARTOFSP
UNLOAD_USER_DICT_ERROR_INVALID_PHONEME_SET
UNLOAD_USER_DICT_ERROR_INTERNAL
UNLOAD_USER_DICT_ERROR_OTHER

```

End Enum

Public Enum DFTTSTextType

DFTTS_TEXT_TYPE_PLAIN
DFTTS_TEXT_TYPE_XML

End Enum

Public Enum DFTTSSpeakReturnValue

DFTTS_SPEAK_SUCCESS
DFTTS_SPEAK_ERROR_CHANNEL_MEM_FAIL
DFTTS_SPEAK_ERROR_TEXT_NULL
DFTTS_SPEAK_ERROR_TEXT_ZERO_LEN
DFTTS_SPEAK_ERROR_DB_NOT_LOADED
DFTTS_SPEAK_ERROR_SET_SOUND_CARD_FAIL
DFTTS_SPEAK_ERROR_UNIMPLEMENTED
DFTTS_SPEAK_ERROR_INTERNAL
DFTTS_SPEAK_ERROR_INVALID_PARAM
DFTTS_SPEAK_ERROR_INVALID_POINTER
DFTTS_SPEAK_ERROR_UNKNOWN_ENCODING
DFTTS_SPEAK_ERROR_OBJECT_NOT_FOUND
DFTTS_SPEAK_ERROR_INTERRUPTED
DFTTS_SPEAK_ERROR_INVALID_VOICE
DFTTS_SPEAK_ERROR_WRONG_EVENT
DFTTS_SPEAK_ERROR_ENGINE_INUSE
DFTTS_SPEAK_ERROR_NETWORK_ERROR
DFTTS_SPEAK_ERROR_INVALID_KEY
DFTTS_SPEAK_ERROR_QUEUE_FULL
DFTTS_SPEAK_ERROR_TOKEN_TIMEOUT
DFTTS_SPEAK_ERROR_FAILED
DFTTS_SPEAK_ERROR_INVALIDARG
DFTTS_SPEAK_ERROR_OUTOFMEMORY
DFTTS_SPEAK_ERROR_NOTIMPL
DFTTS_SPEAK_ERROR_ABORT
DFTTS_SPEAK_ERROR_UNKNOWN
DFTTS_SPEAK_ERROR_BADHANDLE
DFTTS_SPEAK_ERROR_EXCEPTION
DFTTS_SPEAK_ERROR_EMPTY
DFTTS_SPEAK_ERROR_FULL
DFTTS_SPEAK_ERROR_INVALIDSTATE
DFTTS_SPEAK_ERROR_BADVERSION
DFTTS_SPEAK_ERROR_INSUFFICIENT_BUFFER
DFTTS_SPEAK_ERROR_UNSUPPORTED
DFTTS_SPEAK_ERROR_NOLICENSE
DFTTS_SPEAK_ERROR_CREATECHILDPROCESS_FAILED
DFTTS_SPEAK_ERROR_NOENVIRONMENTPATH
DFTTS_SPEAK_ERROR_TIMEOUT
DFTTS_SPEAK_ERROR_OUTOFRESOURCES
DFTTS_SPEAK_ERROR_NOVOICES
DFTTS_SPEAK_ERROR_CREATEFAIL
DFTTS_SPEAK_ERROR_CONNECTFAIL
DFTTS_SPEAK_ERROR_BINDFAIL
DFTTS_SPEAK_ERROR_LISTENFAIL
DFTTS_SPEAK_ERROR_CONNECTIONCLOSED
DFTTS_SPEAK_ERROR_ACCEPTFAIL
DFTTS_SPEAK_ERROR_SOCKETTIMEOUT
DFTTS_SPEAK_ERROR_SOCKETERROR
DFTTS_SPEAK_ERROR_NOMORESERSERS
DFTTS_SPEAK_ERROR_SOCKET_EWOULDBLOCK
DFTTS_SPEAK_ERROR_SOCKET_EINPROGRESS
DFTTS_SPEAK_ERROR_SOCKET_EALREADY
DFTTS_SPEAK_ERROR_SOCKET_ENOTSOCK
DFTTS_SPEAK_ERROR_SOCKET_EDESTADDRREQ
DFTTS_SPEAK_ERROR_SOCKET_MSGSIZE
DFTTS_SPEAK_ERROR_SOCKET_EPROTOTYPE
DFTTS_SPEAK_ERROR_SOCKET_ENOPROTOPT
DFTTS_SPEAK_ERROR_SOCKET_EPROTONOSUPPORT
DFTTS_SPEAK_ERROR_SOCKET_ESOCKTNOSUPPORT
DFTTS_SPEAK_ERROR_SOCKET_EOPNOTSUPP
DFTTS_SPEAK_ERROR_SOCKET_EPFNOSUPPORT
DFTTS_SPEAK_ERROR_SOCKET_EAFNOSUPPORT

```

DFTTS_SPEAK_ERROR_SOCKET_EADDRINUSE
DFTTS_SPEAK_ERROR_SOCKET_EADDRNOTAVAIL
DFTTS_SPEAK_ERROR_SOCKET_ENETDOWN
DFTTS_SPEAK_ERROR_SOCKET_ENETUNREACH
DFTTS_SPEAK_ERROR_SOCKET_ENETRESET
DFTTS_SPEAK_ERROR_SOCKET_ECONNABORTED
DFTTS_SPEAK_ERROR_SOCKET_ECONNRESET
DFTTS_SPEAK_ERROR_SOCKET_ENOBUFS
DFTTS_SPEAK_ERROR_SOCKET_EISCONN
DFTTS_SPEAK_ERROR_SOCKET_ENOTCONN
DFTTS_SPEAK_ERROR_SOCKET_ESHUTDOWN
DFTTS_SPEAK_ERROR_SOCKET_ETOOMANYREFS
DFTTS_SPEAK_ERROR_SOCKET_ECONNREFUSED
DFTTS_SPEAK_ERROR_SOCKET_ELOOP
DFTTS_SPEAK_ERROR_SOCKET_ENAMETOOLONG
DFTTS_SPEAK_ERROR_SOCKET_EHOSTDOWN
DFTTS_SPEAK_ERROR_SOCKET_EHOSTUNREACH
DFTTS_SPEAK_ERROR_SOCKET_ENOTEMPTY
DFTTS_SPEAK_ERROR_SOCKET_EPROCLIM
DFTTS_SPEAK_ERROR_THREADSTARTED
DFTTS_SPEAK_ERROR_THREADNOTSTARTED
DFTTS_SPEAK_ERROR_THREADCOULDNOTCREATE
DFTTS_SPEAK_ERROR_BADNVFILE
DFTTS_SPEAK_ERROR_NOAUDIODRIVER
DFTTS_SPEAK_ERROR_DICTNOTFOUND
DFTTS_SPEAK_ERROR_ALREADYPLAYING
DFTTS_SPEAK_ERROR_AUDIOFORMATNOTSUPPORTED
DFTTS_SPEAK_ERROR_XML_INVALID
DFTTS_SPEAK_ERROR_WL_INVALID
DFTTS_SPEAK_ERROR_INVALIDPHONESET
DFTTS_SPEAK_ERROR_INVALIDPHONEME
DFTTS_SPEAK_ERROR_MSGQ_CREATEFAILED
DFTTS_SPEAK_ERROR_MSGQ_ALREADYEXISTS
DFTTS_SPEAK_ERROR_MSGQ_NOTFOUND
DFTTS_SPEAK_ERROR_MSGQ_INVALIDOP
DFTTS_SPEAK_ERROR_MSGQ_NOTOPEN
DFTTS_SPEAK_ERROR_MSGQ_LOCKFAILED
DFTTS_SPEAK_ERROR_MSGQ_ABANDONED
DFTTS_SPEAK_ERROR_MSGQ_OPENFAILED
DFTTS_SPEAK_ERROR_OTHER

```

End Enum

Public Enum DFTTSPauseReturnValue

```

DFTTS_PAUSE_SUCCESS
DFTTS_PAUSE_ERROR_UNIMPLEMENTED
DFTTS_PAUSE_ERROR_INTERNAL
DFTTS_PAUSE_ERROR_INVALID_PARAM
DFTTS_PAUSE_ERROR_INVALID_POINTER
DFTTS_PAUSE_ERROR_OBJECT_NOT_FOUND
DFTTS_PAUSE_ERROR_UNKNOWN_ENCODING
DFTTS_PAUSE_ERROR_INTERRUPTED
DFTTS_PAUSE_ERROR_INVALID_VOICE
DFTTS_PAUSE_ERROR_WRONG_EVENT
DFTTS_PAUSE_ERROR_ENGINE_INUSE
DFTTS_PAUSE_ERROR_NETWORK_ERROR
DFTTS_PAUSE_ERROR_INVALID_KEY
DFTTS_PAUSE_ERROR_QUEUE_FULL
DFTTS_PAUSE_ERROR_TOKEN_TIMEOUT
DFTTS_PAUSE_ERROR_OTHER

```

End Enum

Public Enum DFTTSResumeReturnValue

```

DFTTS_RESUME_SUCCESS
DFTTS_RESUME_ERROR_UNIMPLEMENTED
DFTTS_RESUME_ERROR_INTERNAL

```

DFTTS_RESUME_ERROR_INVALID_PARAM
DFTTS_RESUME_ERROR_INVALID_POINTER
DFTTS_RESUME_ERROR_OBJECT_NOT_FOUND
DFTTS_RESUME_ERROR_UNKNOWN_ENCODING
DFTTS_RESUME_ERROR_INTERRUPTED
DFTTS_RESUME_ERROR_INVALID_VOICE
DFTTS_RESUME_ERROR_WRONG_EVENT
DFTTS_RESUME_ERROR_ENGINE_INUSE
DFTTS_RESUME_ERROR_NETWORK_ERROR
DFTTS_RESUME_ERROR_INVALID_KEY
DFTTS_RESUME_ERROR_QUEUE_FULL
DFTTS_RESUME_ERROR_TOKEN_TIMEOUT
DFTTS_RESUME_ERROR_FAILED
DFTTS_RESUME_ERROR_INVALIDARG
DFTTS_RESUME_ERROR_OUTOFMEMORY
DFTTS_RESUME_ERROR_NOTIMPL
DFTTS_RESUME_ERROR_ABORT
DFTTS_RESUME_ERROR_UNKNOWN
DFTTS_RESUME_ERROR_BADHANDLE
DFTTS_RESUME_ERROR_EXCEPTION
DFTTS_RESUME_ERROR_EMPTY
DFTTS_RESUME_ERROR_FULL
DFTTS_RESUME_ERROR_INVALIDSTATE
DFTTS_RESUME_ERROR_BADVERSION
DFTTS_RESUME_ERROR_INSUFFICIENT_BUFFER
DFTTS_RESUME_ERROR_UNSUPPORTED
DFTTS_RESUME_ERROR_NOLICENSE
DFTTS_RESUME_ERROR_CREATECHILDPROCESS_FAILED
DFTTS_RESUME_ERROR_NOENVIRONMENTPATH
DFTTS_RESUME_ERROR_TIMEOUT
DFTTS_RESUME_ERROR_OUTOFRESOURCES
DFTTS_RESUME_ERROR_NOVOICES
DFTTS_RESUME_ERROR_CREATEFAIL
DFTTS_RESUME_ERROR_CONNECTFAIL
DFTTS_RESUME_ERROR_BINDFAIL
DFTTS_RESUME_ERROR_LISTENFAIL
DFTTS_RESUME_ERROR_CONNECTIONCLOSED
DFTTS_RESUME_ERROR_ACCEPTFAIL
DFTTS_RESUME_ERROR_SOCKETTIMEOUT
DFTTS_RESUME_ERROR_SOCKETERROR
DFTTS_RESUME_ERROR_NOMORESERVERS
DFTTS_RESUME_ERROR_SOCKET_EWOULDBLOCK
DFTTS_RESUME_ERROR_SOCKET_EINPROGRESS
DFTTS_RESUME_ERROR_SOCKET_EALREADY
DFTTS_RESUME_ERROR_SOCKET_ENOTSOCK
DFTTS_RESUME_ERROR_SOCKET_EDESTADDRREQ
DFTTS_RESUME_ERROR_SOCKET EMSGSIZE
DFTTS_RESUME_ERROR_SOCKET_EPROTOTYPE
DFTTS_RESUME_ERROR_SOCKET_ENOPROTOOPT
DFTTS_RESUME_ERROR_SOCKET_EPROTONOSUPPORT
DFTTS_RESUME_ERROR_SOCKET_ESOCKTNOSUPPORT
DFTTS_RESUME_ERROR_SOCKET_EOPNOTSUPP
DFTTS_RESUME_ERROR_SOCKET_EPFNOSUPPORT
DFTTS_RESUME_ERROR_SOCKET_EAFNOSUPPORT
DFTTS_RESUME_ERROR_SOCKET_EADDRINUSE
DFTTS_RESUME_ERROR_SOCKET_EADDRNOTAVAIL
DFTTS_RESUME_ERROR_SOCKET_ENETDOWN
DFTTS_RESUME_ERROR_SOCKET_ENETUNREACH
DFTTS_RESUME_ERROR_SOCKET_ENETRESET
DFTTS_RESUME_ERROR_SOCKET_ECONNABORTED
DFTTS_RESUME_ERROR_SOCKET_ECONNRESET
DFTTS_RESUME_ERROR_SOCKET_ENOBUFS
DFTTS_RESUME_ERROR_SOCKET_EISCONN
DFTTS_RESUME_ERROR_SOCKET_ENOTCONN
DFTTS_RESUME_ERROR_SOCKET_ESHUTDOWN
DFTTS_RESUME_ERROR_SOCKET_ETOOMANYREFS
DFTTS_RESUME_ERROR_SOCKET_ECONNREFUSED
DFTTS_RESUME_ERROR_SOCKET_ELOOP
DFTTS_RESUME_ERROR_SOCKET_ENAMETOOLONG
DFTTS_RESUME_ERROR_SOCKET_EHOSTDOWN
DFTTS_RESUME_ERROR_SOCKET_EHOSTUNREACH

```

DFTTS_RESUME_ERROR_SOCKET_ENOTEMPTY
DFTTS_RESUME_ERROR_SOCKET_EPROCLIM
DFTTS_RESUME_ERROR_THREADSTARTED
DFTTS_RESUME_ERROR_THREADNOTSTARTED
DFTTS_RESUME_ERROR_THREADCOULDNOTCREATE
DFTTS_RESUME_ERROR_BADNVFILE
DFTTS_RESUME_ERROR_NOAUDIODRIVER
DFTTS_RESUME_ERROR_DICTNOTFOUND
DFTTS_RESUME_ERROR_ALREADYPLAYING
DFTTS_RESUME_ERROR_AUDIOFORMATNOTSUPPORTED
DFTTS_RESUME_ERROR_XML_INVALID
DFTTS_RESUME_ERROR_WL_INVALID
DFTTS_RESUME_ERROR_INVALIDPHONESET
DFTTS_RESUME_ERROR_INVALIDPHONEME
DFTTS_RESUME_ERROR_MSGQ_CREATEFAILED
DFTTS_RESUME_ERROR_MSGQ_ALREADYEXISTS
DFTTS_RESUME_ERROR_MSGQ_NOTFOUND
DFTTS_RESUME_ERROR_MSGQ_INVALIDOP
DFTTS_RESUME_ERROR_MSGQ_NOTOPEN
DFTTS_RESUME_ERROR_MSGQ_LOCKFAILED
DFTTS_RESUME_ERROR_MSGQ_ABANDONED
DFTTS_RESUME_ERROR_MSGQ_OPENFAILED
DFTTS_RESUME_ERROR_NOTHING_TO_RESUME
DFTTS_RESUME_ERROR_OTHER

```

End Enum

Public Enum DFTTSStopReturnValue

```

DFTTS_STOP_SUCCESS
DFTTS_STOP_ERROR_INVALID_POINTER
DFTTS_STOP_ERROR_INVALID_ARG
DFTTS_STOP_ERROR_OUT_OF_MEMORY
DFTTS_STOP_ERROR_INVALID_FLAGS
DFTTS_STOP_ERROR_ENGINE_INUSE
DFTTS_STOP_ERROR_OTHER

```

End Enum

Public Enum DFTTSExportReturnValue

```

DFTTS_EXPORT_SUCCESS
DFTTS_EXPORT_ERROR_FORMAT_NOT_SUPPORTED
DFTTS_EXPORT_ERROR_CHANNEL_MEM_FAIL
DFTTS_EXPORT_ERROR_TEXT_NULL
DFTTS_EXPORT_ERROR_TEXT_ZERO_LEN
DFTTS_EXPORT_ERROR_DB_NOT_LOADED
DFTTS_EXPORT_ERROR_GEN_FILE_FAIL
DFTTS_EXPORT_ERROR_BUFFER_NULL
DFTTS_EXPORT_ERROR_THREAD_IN_USE
DFTTS_EXPORT_ERROR_UNIMPLEMENTED
DFTTS_EXPORT_ERROR_INTERNAL
DFTTS_EXPORT_ERROR_INVALID_PARAM
DFTTS_EXPORT_ERROR_INVALID_POINTER
DFTTS_EXPORT_ERROR_OBJECT_NOT_FOUND
DFTTS_EXPORT_ERROR_UNKNOWN_ENCODING
DFTTS_EXPORT_ERROR_INTERRUPTED
DFTTS_EXPORT_ERROR_INVALID_VOICE
DFTTS_EXPORT_ERROR_WRONG_EVENT
DFTTS_EXPORT_ERROR_ENGINE_INUSE
DFTTS_EXPORT_ERROR_NETWORK_ERROR
DFTTS_EXPORT_ERROR_INVALID_KEY
DFTTS_EXPORT_ERROR_QUEUE_FULL
DFTTS_EXPORT_ERROR_TOKEN_TIMEOUT
DFTTS_EXPORT_ERROR_FAILED
DFTTS_EXPORT_ERROR_INVALIDARG
DFTTS_EXPORT_ERROR_OUTOFMEMORY
DFTTS_EXPORT_ERROR_NOTIMPL
DFTTS_EXPORT_ERROR_ABORT
DFTTS_EXPORT_ERROR_UNKNOWN

```

DFTTS_EXPORT_ERROR_BADHANDLE
DFTTS_EXPORT_ERROR_EXCEPTION
DFTTS_EXPORT_ERROR_EMPTY
DFTTS_EXPORT_ERROR_FULL
DFTTS_EXPORT_ERROR_INVALIDSTATE
DFTTS_EXPORT_ERROR_BADVERSION
DFTTS_EXPORT_ERROR_INSUFFICIENT_BUFFER
DFTTS_EXPORT_ERROR_UNSUPPORTED
DFTTS_EXPORT_ERROR_NOLICENSE
DFTTS_EXPORT_ERROR_CREATECHILDPROCESS_FAILED
DFTTS_EXPORT_ERROR_NOENVIRONMENTPATH
DFTTS_EXPORT_ERROR_TIMEOUT
DFTTS_EXPORT_ERROR_OUTOFRESOURCES
DFTTS_EXPORT_ERROR_NOVOICES
DFTTS_EXPORT_ERROR_CREATEFAIL
DFTTS_EXPORT_ERROR_CONNECTFAIL
DFTTS_EXPORT_ERROR_BINDFAIL
DFTTS_EXPORT_ERROR_LISTENFAIL
DFTTS_EXPORT_ERROR_CONNECTIONCLOSED
DFTTS_EXPORT_ERROR_ACCEPTFAIL
DFTTS_EXPORT_ERROR_SOCKETTIMEOUT
DFTTS_EXPORT_ERROR_SOCKETERROR
DFTTS_EXPORT_ERROR_NOMORESERVERS
DFTTS_EXPORT_ERROR_SOCKET_EWOULDBLOCK
DFTTS_EXPORT_ERROR_SOCKET_EINPROGRESS
DFTTS_EXPORT_ERROR_SOCKET_EALREADY
DFTTS_EXPORT_ERROR_SOCKET_ENOTSOCK
DFTTS_EXPORT_ERROR_SOCKET_EDESTADDRREQ
DFTTS_EXPORT_ERROR_SOCKET EMSGSIZE
DFTTS_EXPORT_ERROR_SOCKET_EPROTOTYPE
DFTTS_EXPORT_ERROR_SOCKET_ENOPROTOOPT
DFTTS_EXPORT_ERROR_SOCKET_EPROTONOSUPPORT
DFTTS_EXPORT_ERROR_SOCKET_ESOCKTNOSUPPORT
DFTTS_EXPORT_ERROR_SOCKET_EOPNOTSUPP
DFTTS_EXPORT_ERROR_SOCKET_EPFNOSUPPORT
DFTTS_EXPORT_ERROR_SOCKET_EAFNOSUPPORT
DFTTS_EXPORT_ERROR_SOCKET_EADDRINUSE
DFTTS_EXPORT_ERROR_SOCKET_EADDRNOTAVAIL
DFTTS_EXPORT_ERROR_SOCKET_ENETDOWN
DFTTS_EXPORT_ERROR_SOCKET_ENETUNREACH
DFTTS_EXPORT_ERROR_SOCKET_ENETRESET
DFTTS_EXPORT_ERROR_SOCKET_ECONNABORTED
DFTTS_EXPORT_ERROR_SOCKET_ECONNRESET
DFTTS_EXPORT_ERROR_SOCKET_ENOBUFS
DFTTS_EXPORT_ERROR_SOCKET_EISCONN
DFTTS_EXPORT_ERROR_SOCKET_ENOTCONN
DFTTS_EXPORT_ERROR_SOCKET_ESHUTDOWN
DFTTS_EXPORT_ERROR_SOCKET_ETOOMANYREFS
DFTTS_EXPORT_ERROR_SOCKET_ECONNREFUSED
DFTTS_EXPORT_ERROR_SOCKET_ELOOP
DFTTS_EXPORT_ERROR_SOCKET_ENAMETOOLONG
DFTTS_EXPORT_ERROR_SOCKET_EHOSTDOWN
DFTTS_EXPORT_ERROR_SOCKET_EHOSTUNREACH
DFTTS_EXPORT_ERROR_SOCKET_ENOTEMPTY
DFTTS_EXPORT_ERROR_SOCKET_EPROCLIM
DFTTS_EXPORT_ERROR_THREADSTARTED
DFTTS_EXPORT_ERROR_THREADNOTSTARTED
DFTTS_EXPORT_ERROR_THREADCOULDNOTCREATE
DFTTS_EXPORT_ERROR_BADNFILE
DFTTS_EXPORT_ERROR_NOAUDIODRIVER
DFTTS_EXPORT_ERROR_DICTNOTFOUND
DFTTS_EXPORT_ERROR_ALREADYPLAYING
DFTTS_EXPORT_ERROR_AUDIOFORMATNOTSUPPORTED
DFTTS_EXPORT_ERROR_XML_INVALID
DFTTS_EXPORT_ERROR_WL_INVALID
DFTTS_EXPORT_ERROR_INVALIDPHONESET
DFTTS_EXPORT_ERROR_INVALIDPHONEME
DFTTS_EXPORT_ERROR_MSGQ_CREATEFAILED
DFTTS_EXPORT_ERROR_MSGQ_ALREADYEXISTS
DFTTS_EXPORT_ERROR_MSGQ_NOTFOUND
DFTTS_EXPORT_ERROR_MSGQ_INVALIDOP

```

DFTTS_EXPORT_ERROR_MSGQ_NOTOPEN
DFTTS_EXPORT_ERROR_MSGQ_LOCKFAILED
DFTTS_EXPORT_ERROR_MSGQ_ABANDONED
DFTTS_EXPORT_ERROR_MSGQ_OPENFAILED
DFTTS_EXPORT_ERROR_OTHER

End Enum

Public Enum DFTTSVoiceInfoReturnValue

    GET_DFTTS_VI_SUCCESS
    GET_DFTTS_VI_NO_MORE_ITEMS
    GET_DFTTS_VI_ERROR_BUFFER_TOO_SMALL
    GET_DFTTS_VI_ERROR_OTHER

End Enum

'
' Language IDs.
'
' The following two combinations of primary language ID and
' sublanguage ID have special semantics:
'
' Primary Language ID    Sublanguage ID    Result
' -----
' LANG_NEUTRAL           SUBLANG_NEUTRAL    Language neutral
' LANG_NEUTRAL           SUBLANG_DEFAULT    User default language
' LANG_NEUTRAL           SUBLANG_SYS_DEFAULT System default language
' LANG_INVARIANT         SUBLANG_NEUTRAL    Invariant locale
'
'
' Primary language IDs.
'

Public Const LANG_NEUTRAL As Integer = &H0
Public Const LANG_INVARIANT As Integer = &H7F

Public Const LANG_AFRIKAANS As Integer = &H36
Public Const LANG_ALBANIAN As Integer = &H1C
Public Const LANG_ALSATIAN As Integer = &H84
Public Const LANG_AMHARIC As Integer = &H5E
Public Const LANG_ARABIC As Integer = &H1
Public Const LANG_ARMENIAN As Integer = &H2B
Public Const LANG_ASSAMESE As Integer = &H4D
Public Const LANG_AZERI As Integer = &H2C
Public Const LANG_BASHKIR As Integer = &H6D
Public Const LANG_BASQUE As Integer = &H2D
Public Const LANG_BELARUSIAN As Integer = &H23
Public Const LANG_BENGALI As Integer = &H45
Public Const LANG_BRETON As Integer = &H7E
Public Const LANG_BOSNIAN As Integer = &H1A ' Use with
SUBLANG_BOSNIAN * Sublanguage IDs
Public Const LANG_BOSNIAN_NEUTRAL As Integer = &H781A ' Use with the
ConvertDefaultLocale function
Public Const LANG_BULGARIAN As Integer = &H2
Public Const LANG_CATALAN As Integer = &H3
Public Const LANG_CHINESE As Integer = &H4 ' Use with
SUBLANG_CHINESE * Sublanguage IDs
Public Const LANG_CHINESE_SIMPLIFIED As Integer = &H4 ' Use with the
ConvertDefaultLocale function
Public Const LANG_CHINESE_TRADITIONAL As Integer = &H7C04 ' Use with the
ConvertDefaultLocale function
Public Const LANG_CORSICAN As Integer = &H83
Public Const LANG_CROATIAN As Integer = &H1A
Public Const LANG_CZECH As Integer = &H5
Public Const LANG_DANISH As Integer = &H6
Public Const LANG_DARI As Integer = &H8C
Public Const LANG_DIVEHI As Integer = &H65
Public Const LANG_DUTCH As Integer = &H13
Public Const LANG_ENGLISH As Integer = &H9

```



```

Public Const LANG_ESTONIAN As Integer = &H25
Public Const LANG_FAEROESE As Integer = &H38
Public Const LANG_FARSI As Integer = &H29 ' Deprecated: use
LANG_PERSIAN instead
Public Const LANG_FILIPINO As Integer = &H64
Public Const LANG_FINNISH As Integer = &HB
Public Const LANG_FRENCH As Integer = &HC
Public Const LANG_FRISIAN As Integer = &H62
Public Const LANG_GALICIAN As Integer = &H56
Public Const LANG_GEORGIAN As Integer = &H37
Public Const LANG_GERMAN As Integer = &H7
Public Const LANG_GREEK As Integer = &H8
Public Const LANG_GREENLANDIC As Integer = &H6F
Public Const LANG_GUJARATI As Integer = &H47
Public Const LANG_HAUSA As Integer = &H68
Public Const LANG_HEBREW As Integer = &HD
Public Const LANG_HINDI As Integer = &H39
Public Const LANG_HUNGARIAN As Integer = &HE
Public Const LANG_ICELANDIC As Integer = &HF
Public Const LANG_IGBO As Integer = &H70
Public Const LANG_INDONESIAN As Integer = &H21
Public Const LANG_INUKTITUT As Integer = &H5D
Public Const LANG_IRISH As Integer = &H3C ' Use with the
SUBLANG_IRISH_IRELAND Sublanguage ID
Public Const LANG_ITALIAN As Integer = &H10
Public Const LANG_JAPANESE As Integer = &H11
Public Const LANG_KANNADA As Integer = &H4B
Public Const LANG_KASHMIRI As Integer = &H60
Public Const LANG_KAZAK As Integer = &H3F
Public Const LANG_KHMER As Integer = &H53
Public Const LANG_KICHE As Integer = &H86
Public Const LANG_KINYARWANDA As Integer = &H87
Public Const LANG_KONKANI As Integer = &H57
Public Const LANG_KOREAN As Integer = &H12
Public Const LANG_KYRGYZ As Integer = &H40
Public Const LANG_LAO As Integer = &H54
Public Const LANG_LATVIAN As Integer = &H26
Public Const LANG_LITHUANIAN As Integer = &H27
Public Const LANG_LOWER_SORBIAN As Integer = &H2E
Public Const LANG_LUXEMBOURGISH As Integer = &H6E
Public Const LANG_MACEDONIAN As Integer = &H2F ' the Former
Yugoslav Republic of Macedonia
Public Const LANG_MALAY As Integer = &H3E
Public Const LANG_MALAYALAM As Integer = &H4C
Public Const LANG_MALTESE As Integer = &H3A
Public Const LANG_MANIPURI As Integer = &H58
Public Const LANG_MAORI As Integer = &H81
Public Const LANG_MAPUDUNGUN As Integer = &H7A
Public Const LANG_MARATHI As Integer = &H4E
Public Const LANG_MOHAWK As Integer = &H7C
Public Const LANG_MONGOLIAN As Integer = &H50
Public Const LANG_NEPALI As Integer = &H61
Public Const LANG_NORWEGIAN As Integer = &H14
Public Const LANG_OCCITAN As Integer = &H82
Public Const LANG_ORIYA As Integer = &H48
Public Const LANG_PASHTO As Integer = &H63
Public Const LANG_PERSIAN As Integer = &H29
Public Const LANG_POLISH As Integer = &H15
Public Const LANG_PORTUGUESE As Integer = &H16
Public Const LANG_PUNJABI As Integer = &H46
Public Const LANG_QUECHUA As Integer = &H6B
Public Const LANG_ROMANIAN As Integer = &H18
Public Const LANG_ROMANSH As Integer = &H17
Public Const LANG_RUSSIAN As Integer = &H19
Public Const LANG_SAMI As Integer = &H3B
Public Const LANG_SANSKRIT As Integer = &H4F
Public Const LANG_SERBIAN As Integer = &H1A ' Use with the
SUBLANG_SERBIAN_* Sublanguage IDs
Public Const LANG_SERBIAN_NEUTRAL As Integer = &H7C1A ' Use with the
ConvertDefaultLocale function
Public Const LANG_SINDHI As Integer = &H59

```

```

Public Const LANG_SINHALESE As Integer = &H5B
Public Const LANG_SLOVAK As Integer = &H1B
Public Const LANG_SLOVENIAN As Integer = &H24
Public Const LANG_SOTHO As Integer = &H6C
Public Const LANG_SPANISH As Integer = &HA
Public Const LANG_SWAHILI As Integer = &H41
Public Const LANG_SWEDISH As Integer = &H1D
Public Const LANG_SYRIAC As Integer = &H5A
Public Const LANG_TAJIK As Integer = &H28
Public Const LANG_TAMAZIGHT As Integer = &H5F
Public Const LANG_TAMIL As Integer = &H49
Public Const LANG_TATAR As Integer = &H44
Public Const LANG_TELUGU As Integer = &H4A
Public Const LANG_THAI As Integer = &H1E
Public Const LANG_TIBETAN As Integer = &H51
Public Const LANG_TIGRIGNA As Integer = &H73
Public Const LANG_TSWANA As Integer = &H32
Public Const LANG_TURKISH As Integer = &H1F
Public Const LANG_TURKMEN As Integer = &H42
Public Const LANG_UGHUR As Integer = &H80
Public Const LANG_UKRAINIAN As Integer = &H22
Public Const LANG_UPPER_SORBIAN As Integer = &H2E
Public Const LANG_URDU As Integer = &H20
Public Const LANG_UZBEK As Integer = &H43
Public Const LANG_VIETNAMESE As Integer = &H2A
Public Const LANG_WELSH As Integer = &H52
Public Const LANG_WOLOF As Integer = &H88
Public Const LANG_XHOSA As Integer = &H34
Public Const LANG_YAKUT As Integer = &H85
Public Const LANG_YI As Integer = &H78
Public Const LANG_YORUBA As Integer = &H6A
Public Const LANG_ZULU As Integer = &H35

'
'   Sublanguage IDs.
'
'   The name immediately following SUBLANG_ dictates which primary
'   language ID that sublanguage ID can be combined with to form a
'   valid language ID.
'

Public Const SUBLANG_NEUTRAL As Integer = &H0
language neutral
Public Const SUBLANG_DEFAULT As Integer = &H1
default
Public Const SUBLANG_SYS_DEFAULT As Integer = &H2
system default
Public Const SUBLANG_CUSTOM_DEFAULT As Integer = &H3
default custom language/locale
Public Const SUBLANG_CUSTOM_UNSPECIFIED As Integer = &H4
custom language/locale
Public Const SUBLANG_UI_CUSTOM_DEFAULT As Integer = &H5
Default custom MUI language/locale

Public Const SUBLANG_AFIKAANS_SOUTH_AFRICA As Integer = &H1
Afrikaans (South Africa) As Integer = &H0436 af-ZA
Public Const SUBLANG_ALBANIAN_ALBANIA As Integer = &H1
Albanian (Albania) As Integer = &H041c sq-AL
Public Const SUBLANG_ALSATIAN_FRANCE As Integer = &H1
Alsatian (France) As Integer = &H0484
Public Const SUBLANG_AMHARIC_ETHIOPIA As Integer = &H1
Amharic (Ethiopia) As Integer = &H045e
Public Const SUBLANG_ARABIC_SAUDI_ARABIA As Integer = &H1
Arabic (Saudi Arabia)
Public Const SUBLANG_ARABIC_IRAQ As Integer = &H2
Arabic (Iraq)
Public Const SUBLANG_ARABIC_EGYPT As Integer = &H3
Arabic (Egypt)
Public Const SUBLANG_ARABIC_LIBYA As Integer = &H4
Arabic (Libya)

```

```

Public Const SUBLANG_ARABIC_ALGERIA As Integer = &H5
Arabic (Algeria)
Public Const SUBLANG_ARABIC_MOROCCO As Integer = &H6
Arabic (Morocco)
Public Const SUBLANG_ARABIC_TUNISIA As Integer = &H7
Arabic (Tunisia)
Public Const SUBLANG_ARABIC_OMAN As Integer = &H8
Arabic (Oman)
Public Const SUBLANG_ARABIC_YEMEN As Integer = &H9
Arabic (Yemen)
Public Const SUBLANG_ARABIC_SYRIA As Integer = &HA
Arabic (Syria)
Public Const SUBLANG_ARABIC_JORDAN As Integer = &HB
Arabic (Jordan)
Public Const SUBLANG_ARABIC_LEBANON As Integer = &HC
Arabic (Lebanon)
Public Const SUBLANG_ARABIC_KUWAIT As Integer = &HD
Arabic (Kuwait)
Public Const SUBLANG_ARABIC_UAE As Integer = &HE
Arabic (U.A.E)
Public Const SUBLANG_ARABIC_BAHRAIN As Integer = &HF
Arabic (Bahrain)
Public Const SUBLANG_ARABIC_QATAR As Integer = &H10
Arabic (Qatar)
Public Const SUBLANG_ARMENIAN_ARMENIA As Integer = &H1
Armenian (Armenia) As Integer = &H042b hy-AM
Public Const SUBLANG_ASSAMESE_INDIA As Integer = &H1
Assamese (India) As Integer = &H044d
Public Const SUBLANG_AZERI_LATIN As Integer = &H1
(Latin) Azeri
Public Const SUBLANG_AZERI_CYRILLIC As Integer = &H2
(Cyrillic) Azeri
Public Const SUBLANG_BASHKIR_RUSSIA As Integer = &H1
Bashkir (Russia) As Integer = &H046d ba-RU
Public Const SUBLANG_BASQUE_BASQUE As Integer = &H1
Basque (Basque) As Integer = &H042d eu-ES
Public Const SUBLANG_BELARUSIAN_BELARUS As Integer = &H1
Belarusian (Belarus) As Integer = &H0423 be-BY
Public Const SUBLANG_BENGALI_INDIA As Integer = &H1
Bengali (India)
Public Const SUBLANG_BENGALI_BANGLADESH As Integer = &H2
Bengali (Bangladesh)
Public Const SUBLANG_BOSNIAN_BOSNIA_HERZEGOVINA_LATIN As Integer = &H5
Bosnian (Bosnia and Herzegovina - Latin) As Integer = &H141a bs-BA-Latn
Public Const SUBLANG_BOSNIAN_BOSNIA_HERZEGOVINA_CYRILLIC As Integer = &H8
Bosnian (Bosnia and Herzegovina - Cyrillic) As Integer = &H201a bs-BA-Cyrl
Public Const SUBLANG_BRETON_FRANCE As Integer = &H1
Breton (France) As Integer = &H047e
Public Const SUBLANG_BULGARIAN_BULGARIA As Integer = &H1
Bulgarian (Bulgaria) As Integer = &H0402
Public Const SUBLANG_CATALAN_CATALAN As Integer = &H1
Catalan (Catalan) As Integer = &H0403
Public Const SUBLANG_CHINESE_TRADITIONAL As Integer = &H1
Chinese (Taiwan) As Integer = &H0404 zh-TW
Public Const SUBLANG_CHINESE_SIMPLIFIED As Integer = &H2
Chinese (PR China) As Integer = &H0804 zh-CN
Public Const SUBLANG_CHINESE_HONGKONG As Integer = &H3
Chinese (Hong Kong S.A.R., P.R.C.) As Integer = &H0c04 zh-HK
Public Const SUBLANG_CHINESE_SINGAPORE As Integer = &H4
Chinese (Singapore) As Integer = &H1004 zh-SG
Public Const SUBLANG_CHINESE_MACAU As Integer = &H5
Chinese (Macau S.A.R.) As Integer = &H1404 zh-MO
Public Const SUBLANG_CORSICAN_FRANCE As Integer = &H1
Corsican (France) As Integer = &H0483
Public Const SUBLANG_CZECH_CZECH_REPUBLIC As Integer = &H1
(Czech Republic) As Integer = &H0405
Public Const SUBLANG_CROATIAN_CROATIA As Integer = &H1
Croatian (Croatia)
Public Const SUBLANG_CROATIAN_BOSNIA_HERZEGOVINA_LATIN As Integer = &H4
Croatian (Bosnia and Herzegovina - Latin) As Integer = &H101a hr-BA

```

```

    Public Const SUBLANG_DANISH_DENMARK As Integer = &H1
Danish (Denmark) As Integer = &H0406
    Public Const SUBLANG_DARI_AFGHANISTAN As Integer = &H1
(Afghanistan)
    Public Const SUBLANG_DIVEHI_MALDIVES As Integer = &H1
Divehi (Maldives) As Integer = &H0465 div-MV
    Public Const SUBLANG_DUTCH As Integer = &H1
    Public Const SUBLANG_DUTCH_BELGIAN As Integer = &H2
(Belgian)
    Public Const SUBLANG_ENGLISH_US As Integer = &H1
English (USA)
    Public Const SUBLANG_ENGLISH_UK As Integer = &H2
English (UK)
    Public Const SUBLANG_ENGLISH_AUS As Integer = &H3
English (Australian)
    Public Const SUBLANG_ENGLISH_CAN As Integer = &H4
English (Canadian)
    Public Const SUBLANG_ENGLISH_NZ As Integer = &H5
English (New Zealand)
    Public Const SUBLANG_ENGLISH_EIRE As Integer = &H6
English (Irish)
    Public Const SUBLANG_ENGLISH_SOUTH_AFRICA As Integer = &H7
English (South Africa)
    Public Const SUBLANG_ENGLISH_JAMAICA As Integer = &H8
English (Jamaica)
    Public Const SUBLANG_ENGLISH_CARIBBEAN As Integer = &H9
English (Caribbean)
    Public Const SUBLANG_ENGLISH_BELIZE As Integer = &HA
English (Belize)
    Public Const SUBLANG_ENGLISH_TRINIDAD As Integer = &HB
English (Trinidad)
    Public Const SUBLANG_ENGLISH_ZIMBABWE As Integer = &HC
English (Zimbabwe)
    Public Const SUBLANG_ENGLISH_PHILIPPINES As Integer = &HD
English (Philippines)
    Public Const SUBLANG_ENGLISH_INDIA As Integer = &H10
English (India)
    Public Const SUBLANG_ENGLISH_MALAYSIA As Integer = &H11
English (Malaysia)
    Public Const SUBLANG_ENGLISH_SINGAPORE As Integer = &H12
English (Singapore)
    Public Const SUBLANG_ESTONIAN_ESTONIA As Integer = &H1
Estonian (Estonia) As Integer = &H0425 et-EE
    Public Const SUBLANG_FAEROESE_FAROE_ISLANDS As Integer = &H1
Faroeese (Faroe Islands) As Integer = &H0438 fo-FO
    Public Const SUBLANG_FILIPINO_PHILIPPINES As Integer = &H1
Filipino (Philippines) As Integer = &H0464 fil-PH
    Public Const SUBLANG_FINNISH_FINLAND As Integer = &H1
Finnish (Finland) As Integer = &H040b
    Public Const SUBLANG_FRENCH As Integer = &H1
    Public Const SUBLANG_FRENCH_BELGIAN As Integer = &H2
French (Belgian)
    Public Const SUBLANG_FRENCH_CANADIAN As Integer = &H3
French (Canadian)
    Public Const SUBLANG_FRENCH_SWISS As Integer = &H4
French (Swiss)
    Public Const SUBLANG_FRENCH_LUXEMBOURG As Integer = &H5
French (Luxembourg)
    Public Const SUBLANG_FRENCH_MONACO As Integer = &H6
French (Monaco)
    Public Const SUBLANG_FRISIAN_NETHERLANDS As Integer = &H1
Frisian (Netherlands) As Integer = &H0462 fy-NL
    Public Const SUBLANG_GALICIAN_GALICIAN As Integer = &H1
Galician (Galician) As Integer = &H0456 gl-ES
    Public Const SUBLANG_GEORGIAN_GEORGIA As Integer = &H1
Georgian (Georgia) As Integer = &H0437 ka-GE
    Public Const SUBLANG_GERMAN As Integer = &H1
    Public Const SUBLANG_GERMAN_SWISS As Integer = &H2
German (Swiss)
    Public Const SUBLANG_GERMAN_AUSTRIAN As Integer = &H3
German (Austrian)

```

```

Public Const SUBLANG_GERMAN_LUXEMBOURG As Integer = &H4
German (Luxembourg)
Public Const SUBLANG_GERMAN_LIECHTENSTEIN As Integer = &H5
German (Liechtenstein)
Public Const SUBLANG_GREEK_GREECE As Integer = &H1
(Greece)
Public Const SUBLANG_GREENLANDIC_GREENLAND As Integer = &H1
Greenlandic (Greenland) As Integer = &H046f kl-GL
Public Const SUBLANG_GUJARATI_INDIA As Integer = &H1
Gujarati (India (Gujarati Script)) As Integer = &H0447 gu-IN
Public Const SUBLANG_HAUSA_NIGERIA_LATIN As Integer = &H1
(Latin, Nigeria) As Integer = &H0468 ha-NG-Latn
Public Const SUBLANG_HEBREW_ISRAEL As Integer = &H1
Hebrew (Israel) As Integer = &H040d
Public Const SUBLANG_HINDI_INDIA As Integer = &H1
(India) As Integer = &H0439 hi-IN
Public Const SUBLANG_HUNGARIAN_HUNGARY As Integer = &H1
Hungarian (Hungary) As Integer = &H040e
Public Const SUBLANG_ICELANDIC_ICELAND As Integer = &H1
Icelandic (Iceland) As Integer = &H040f
Public Const SUBLANG_IGBO_NIGERIA As Integer = &H1
(Nigeria) As Integer = &H0470 ig-NG
Public Const SUBLANG_INDONESIAN_INDONESIA As Integer = &H1
Indonesian (Indonesia) As Integer = &H0421 id-ID
Public Const SUBLANG_INUKTITUT_CANADA As Integer = &H1
Inuktitut (Syllabics) (Canada) As Integer = &H045d iu-CA-Cans
Public Const SUBLANG_INUKTITUT_CANADA_LATIN As Integer = &H2
Inuktitut (Canada - Latin)
Public Const SUBLANG_IRISH_IRELAND As Integer = &H2
(Ireland)
Public Const SUBLANG_ITALIAN As Integer = &H1
Italian
Public Const SUBLANG_ITALIAN_SWISS As Integer = &H2
Italian (Swiss)
Public Const SUBLANG_JAPANESE_JAPAN As Integer = &H1
Japanese (Japan) As Integer = &H0411
Public Const SUBLANG_KANNADA_INDIA As Integer = &H1
Kannada (India (Kannada Script)) As Integer = &H044b kn-IN
Public Const SUBLANG_KASHMIRI_SASIA As Integer = &H2
Kashmiri (South Asia)
Public Const SUBLANG_KASHMIRI_INDIA As Integer = &H2
app compatibility only
Public Const SUBLANG_KAZAK_KAZAKHSTAN As Integer = &H1
Kazakh (Kazakhstan) As Integer = &H043f kk-KZ
Public Const SUBLANG_KHMER_CAMBODIA As Integer = &H1
(Cambodia) As Integer = &H0453 kh-KH
Public Const SUBLANG_KICHE_GUATEMALA As Integer = &H1
K'iche (Guatemala)
Public Const SUBLANG_KINYARWANDA_RWANDA As Integer = &H1
Kinyarwanda (Rwanda) As Integer = &H0487 rw-RW
Public Const SUBLANG_KONKANI_INDIA As Integer = &H1
Konkani (India) As Integer = &H0457 kok-IN
Public Const SUBLANG_KOREAN As Integer = &H1
Korean (Extended Wansung)
Public Const SUBLANG_KYRGYZ_KYRGYZSTAN As Integer = &H1
Kyrgyz (Kyrgyzstan) As Integer = &H0440 ky-KG
Public Const SUBLANG_LAO_LAO As Integer = &H1
(Lao PDR) As Integer = &H0454 lo-LA
Public Const SUBLANG_LATVIAN_LATVIA As Integer = &H1
Latvian (Latvia) As Integer = &H0426 lv-LV
Public Const SUBLANG_LITHUANIAN As Integer = &H1
Lithuanian
Public Const SUBLANG_LOWER_SORBIAN_GERMANY As Integer = &H2
Sorbian (Germany) As Integer = &H082e wee-DE
Public Const SUBLANG_LUXEMBOURGISH_LUXEMBOURG As Integer = &H1
Luxembourgish (Luxembourg) As Integer = &H046e lb-LU
Public Const SUBLANG_MACEDONIAN_MACEDONIA As Integer = &H1
Macedonian (Macedonia (FYROM)) As Integer = &H042f mk-MK
Public Const SUBLANG_MALAY_MALAYSIA As Integer = &H1
(Malaysia)

```

```

Public Const SUBLANG_MALAY_BRUNEI_DARUSSALAM As Integer = &H2      ' Malay
(Brunei Darussalam)
Public Const SUBLANG_MALAYALAM_INDIA As Integer = &H1              '
Malayalam (India (Malayalam Script)) As Integer = &H044c ml-IN
Public Const SUBLANG_MALTESE_MALTA As Integer = &H1                '
Maltese (Malta) As Integer = &H043a mt-MT
Public Const SUBLANG_MAORI_NEW_ZEALAND As Integer = &H1            ' Maori
(New Zealand) As Integer = &H0481 mi-NZ
Public Const SUBLANG_MAPUDUNGUN_CHILE As Integer = &H1            '
Mapudungun (Chile) As Integer = &H047a arn-CL
Public Const SUBLANG_MARATHI_INDIA As Integer = &H1               '
Marathi (India) As Integer = &H044e mr-IN
Public Const SUBLANG_MOHAWK_MOHAWK As Integer = &H1               '
Mohawk (Mohawk) As Integer = &H047c moh-CA
Public Const SUBLANG_MONGOLIAN_CYRILLIC_MONGOLIA As Integer = &H1  '
Mongolian (Cyrillic, Mongolia)
Public Const SUBLANG_MONGOLIAN_PRC As Integer = &H2               '
Mongolian (PRC)
Public Const SUBLANG_NEPALI_INDIA As Integer = &H2                '
Nepali (India)
Public Const SUBLANG_NEPALI_NEPAL As Integer = &H1                '
Nepali (Nepal) As Integer = &H0461 ne-NP
Public Const SUBLANG_NORWEGIAN_BOKMAL As Integer = &H1            '
Norwegian (Bokmal)
Public Const SUBLANG_NORWEGIAN_NYNORSK As Integer = &H2           '
Norwegian (Nynorsk)
Public Const SUBLANG_OCCITAN_FRANCE As Integer = &H1              '
Occitan (France) As Integer = &H0482 oc-FR
Public Const SUBLANG_ORIYA_INDIA As Integer = &H1                 ' Oriya
(India (Oriya Script)) As Integer = &H0448 or-IN
Public Const SUBLANG_PASHTO_AFGHANISTAN As Integer = &H1          '
Pashto (Afghanistan)
Public Const SUBLANG_PERSIAN_IRAN As Integer = &H1                '
Persian (Iran) As Integer = &H0429 fa-IR
Public Const SUBLANG_POLISH_POLAND As Integer = &H1              '
Polish (Poland) As Integer = &H0415
Public Const SUBLANG_PORTUGUESE As Integer = &H2                  '
Portuguese
Public Const SUBLANG_PORTUGUESE_BRAZILIAN As Integer = &H1         '
Portuguese (Brazilian)
Public Const SUBLANG_PUNJABI_INDIA As Integer = &H1               '
Punjabi (India (Gurmukhi Script)) As Integer = &H0446 pa-IN
Public Const SUBLANG_QUECHUA_BOLIVIA As Integer = &H1             '
Quechua (Bolivia)
Public Const SUBLANG_QUECHUA_ECUADOR As Integer = &H2             '
Quechua (Ecuador)
Public Const SUBLANG_QUECHUA_PERU As Integer = &H3                '
Quechua (Peru)
Public Const SUBLANG_ROMANIAN_ROMANIA As Integer = &H1            '
Romanian (Romania) As Integer = &H0418
Public Const SUBLANG_ROMANSH_SWITZERLAND As Integer = &H1         '
Romansh (Switzerland) As Integer = &H0417 rm-CH
Public Const SUBLANG_RUSSIAN_RUSSIA As Integer = &H1              '
Russian (Russia) As Integer = &H0419
Public Const SUBLANG_SAMI_NORTHERN_NORWAY As Integer = &H1         '
Northern Sami (Norway)
Public Const SUBLANG_SAMI_NORTHERN_SWEDEN As Integer = &H2        '
Northern Sami (Sweden)
Public Const SUBLANG_SAMI_NORTHERN_FINLAND As Integer = &H3       '
Northern Sami (Finland)
Public Const SUBLANG_SAMI_LULE_NORWAY As Integer = &H4            ' Lule
Sami (Norway)
Public Const SUBLANG_SAMI_LULE_SWEDEN As Integer = &H5            ' Lule
Sami (Sweden)
Public Const SUBLANG_SAMI_SOUTHERN_NORWAY As Integer = &H6         '
Southern Sami (Norway)
Public Const SUBLANG_SAMI_SOUTHERN_SWEDEN As Integer = &H7        '
Southern Sami (Sweden)
Public Const SUBLANG_SAMI_SKOLT_FINLAND As Integer = &H8          ' Skolt
Sami (Finland)

```



```

Public Const SUBLANG_SAMI_INARI_FINLAND As Integer = &H9           ' Inari
Sami (Finland)
Public Const SUBLANG_SANSKRIT_INDIA As Integer = &H1               '
Sanskrit (India) As Integer = &H044f sa-IN
Public Const SUBLANG_SERBIAN_BOSNIA_HERZEGOVINA_LATIN As Integer = &H6 '
Serbian (Bosnia and Herzegovina - Latin)
Public Const SUBLANG_SERBIAN_BOSNIA_HERZEGOVINA_CYRILLIC As Integer = &H7 '
Serbian (Bosnia and Herzegovina - Cyrillic)
Public Const SUBLANG_SERBIAN_CROATIA As Integer = &H1              '
Croatian (Croatia) As Integer = &H041a hr-HR
Public Const SUBLANG_SERBIAN_LATIN As Integer = &H2                 '
Serbian (Latin)
Public Const SUBLANG_SERBIAN_CYRILLIC As Integer = &H3              '
Serbian (Cyrillic)
Public Const SUBLANG_SINDHI_INDIA As Integer = &H1                  '
Sindhi (India) reserved As Integer = &H0459
Public Const SUBLANG_SINDHI_PAKISTAN As Integer = &H2               '
Sindhi (Pakistan) reserved As Integer = &H0859
Public Const SUBLANG_SINDHI_AFGHANISTAN As Integer = &H2            ' For
app compatibility only
Public Const SUBLANG_SINHALESE_SRI_LANKA As Integer = &H1           '
Sinhalese (Sri Lanka)
Public Const SUBLANG_SOTHO_NORTHERN_SOUTH_AFRICA As Integer = &H1   '
Northern Sotho (South Africa)
Public Const SUBLANG_SLOVAK_SLOVAKIA As Integer = &H1              '
Slovak (Slovakia) As Integer = &H041b sk-SK
Public Const SUBLANG_SLOVENIAN_SLOVENIA As Integer = &H1           '
Slovenian (Slovenia) As Integer = &H0424 sl-SI
Public Const SUBLANG_SPANISH As Integer = &H1                       '
Spanish (Castilian)
Public Const SUBLANG_SPANISH_MEXICAN As Integer = &H2              '
Spanish (Mexican)
Public Const SUBLANG_SPANISH_MODERN As Integer = &H3                '
Spanish (Modern)
Public Const SUBLANG_SPANISH_GUATEMALA As Integer = &H4             '
Spanish (Guatemala)
Public Const SUBLANG_SPANISH_COSTA_RICA As Integer = &H5            '
Spanish (Costa Rica)
Public Const SUBLANG_SPANISH_PANAMA As Integer = &H6                '
Spanish (Panama)
Public Const SUBLANG_SPANISH_DOMINICAN_REPUBLIC As Integer = &H7    '
Spanish (Dominican Republic)
Public Const SUBLANG_SPANISH_VENEZUELA As Integer = &H8             '
Spanish (Venezuela)
Public Const SUBLANG_SPANISH_COLOMBIA As Integer = &H9              '
Spanish (Colombia)
Public Const SUBLANG_SPANISH_PERU As Integer = &HA                  '
Spanish (Peru)
Public Const SUBLANG_SPANISH_ARGENTINA As Integer = &HB              '
Spanish (Argentina)
Public Const SUBLANG_SPANISH_ECUADOR As Integer = &HC                '
Spanish (Ecuador)
Public Const SUBLANG_SPANISH_CHILE As Integer = &HD                 '
Spanish (Chile)
Public Const SUBLANG_SPANISH_URUGUAY As Integer = &HE                '
Spanish (Uruguay)
Public Const SUBLANG_SPANISH_PARAGUAY As Integer = &HF               '
Spanish (Paraguay)
Public Const SUBLANG_SPANISH_BOLIVIA As Integer = &H10              '
Spanish (Bolivia)
Public Const SUBLANG_SPANISH_EL_SALVADOR As Integer = &H11          '
Spanish (El Salvador)
Public Const SUBLANG_SPANISH_HONDURAS As Integer = &H12             '
Spanish (Honduras)
Public Const SUBLANG_SPANISH_NICARAGUA As Integer = &H13            '
Spanish (Nicaragua)
Public Const SUBLANG_SPANISH_PUERTO_RICO As Integer = &H14          '
Spanish (Puerto Rico)
Public Const SUBLANG_SPANISH_US As Integer = &H15                   '
Spanish (United States)

```

```

Public Const SUBLANG_SWAHILI_KENYA As Integer = &H1
Swahili (Kenya) As Integer = &H0441 sw-KE
Public Const SUBLANG_SWEDISH As Integer = &H1
Swedish
Public Const SUBLANG_SWEDISH_FINLAND As Integer = &H2
Swedish (Finland)
Public Const SUBLANG_SYRIAC_SYRIA As Integer = &H1
Syriac (Syria) As Integer = &H045a syr-SY
Public Const SUBLANG_TAJIK_TAJIKISTAN As Integer = &H1
(Tajikistan) As Integer = &H0428 tg-TJ-Cyrl
Public Const SUBLANG_TAMAZIGHT_ALGERIA_LATIN As Integer = &H2
Tamazight (Latin, Algeria) As Integer = &H085f tmz-DZ-Latn
Public Const SUBLANG_TAMIL_INDIA As Integer = &H1
(India)
Public Const SUBLANG_TATAR_RUSSIA As Integer = &H1
(Russia) As Integer = &H0444 tt-RU
Public Const SUBLANG_TELUGU_INDIA As Integer = &H1
Telugu (India (Telugu Script)) As Integer = &H044a te-IN
Public Const SUBLANG_THAI_THAILAND As Integer = &H1
(Thailand) As Integer = &H041e th-TH
Public Const SUBLANG_TIBETAN_PRC As Integer = &H1
Tibetan (PRC)
Public Const SUBLANG_TIGRIGNA_ERITREA As Integer = &H2
Tigrigna (Eritrea)
Public Const SUBLANG_TSWANA_SOUTH_AFRICA As Integer = &H1
Setswana / Tswana (South Africa) As Integer = &H0432 tn-ZA
Public Const SUBLANG_TURKISH_TURKEY As Integer = &H1
Turkish (Turkey) As Integer = &H041f tr-TR
Public Const SUBLANG_TURKMEN_TURKMENISTAN As Integer = &H1
Turkmen (Turkmenistan) As Integer = &H0442 tk-TM
Public Const SUBLANG_UGHUR_PRC As Integer = &H1
Uighur (PRC) As Integer = &H0480 ug-CN
Public Const SUBLANG_UKRAINIAN_UKRAINE As Integer = &H1
Ukrainian (Ukraine) As Integer = &H0422 uk-UA
Public Const SUBLANG_UPPER_SORBIAN_GERMANY As Integer = &H1
Sorbian (Germany) As Integer = &H042e wen-DE
Public Const SUBLANG_URDU_PAKISTAN As Integer = &H1
(Pakistan)
Public Const SUBLANG_URDU_INDIA As Integer = &H2
(India)
Public Const SUBLANG_UZBEK_LATIN As Integer = &H1
(Latin)
Public Const SUBLANG_UZBEK_CYRILLIC As Integer = &H2
(Cyrillic)
Public Const SUBLANG_VIETNAMESE_VIETNAM As Integer = &H1
Vietnamese (Vietnam) As Integer = &H042a vi-VN
Public Const SUBLANG_WELSH_UNITED_KINGDOM As Integer = &H1
(United Kingdom) As Integer = &H0452 cy-GB
Public Const SUBLANG_WOLOF_SENEGAL As Integer = &H1
(Senegal)
Public Const SUBLANG_XHOSA_SOUTH_AFRICA As Integer = &H1
isiXhosa / Xhosa (South Africa) As Integer = &H0434 xh-ZA
Public Const SUBLANG_YAKUT_RUSSIA As Integer = &H1
(Russia) As Integer = &H0485 sah-RU
Public Const SUBLANG_YI_PRC As Integer = &H1
(PRC) As Integer = &H0478
Public Const SUBLANG_YORUBA_NIGERIA As Integer = &H1
Yoruba (Nigeria) 046a yo-NG
Public Const SUBLANG_ZULU_SOUTH_AFRICA As Integer = &H1
isiZulu / Zulu (South Africa) As Integer = &H0435 zu-ZA

```

```

'
' A language ID is a 16 bit value which is the combination of a
' primary language ID and a secondary language ID. The bits are
' allocated as follows:
'

```

```

'      +-----+-----+
'      | Sublanguage ID | Primary Language ID |
'      +-----+-----+
'      15             10 9             0 bit

```



```

'

'Neutral language

Public Const LANG_SUBLANG_NEUTRAL As Integer = &H0

Public Enum NeoSpeechVoiceTextAudioFormat

    VT_FILE_API_FMT_S16PCM = 0
    VT_FILE_API_FMT_ALAW = 1
    VT_FILE_API_FMT_MULAW = 2
    VT_FILE_API_FMT_DADPCM = 3
    VT_FILE_API_FMT_S16PCM_WAVE = 4
    VT_FILE_API_FMT_U08PCM_WAVE = 5
    VT_FILE_API_FMT_IMA_WAVE = 6
    VT_FILE_API_FMT_ALAW_WAVE = 7
    VT_FILE_API_FMT_MULAW_WAVE = 8
    VT_FILE_API_FMT_MULAW_AU = 9

End Enum

'Export formats for Cepstral voices

'You need to set the following instead:

'szAudioEncoding [possible values:
' * "riff": Microsoft RIFF (WAV) file
' * "pcm16", "pcm8" PCM 16 bit/8 bit WAV
' * "ulaw" - u-Law (8-bit by definition), "alaw" - A-Law (8-bit by definition)
' * "snd": Sun/NeXT .au (SND) format.
' * "raw": unheadered audio data, native byte order
' * "le": unheadered audio data, little-endian (LSB first)
' * "be": unheadered audio data, big-endian (MSB first)
' * NOTE: that not all encoding types may be supported by all formats. For
instance, SND doesn't support A-Law. ], _
'iAudioSamplingRate [possible values: 8000 (8 KHz), 16000 (16 KHz), 11025 (11.025
kHz), etc ], _
'iAudioChannels [possible values: 1 (mono), 2 (stereo)]

'Export formats for AT&T NV (INTERNAL USE ONLY) _
'Currently only PCM WAV supported (all audio format arguments will be ignored)


Public Enum NeoSpeechVoiceID

    NEOSPEECH_KATE_ENG = 0

    NEOSPEECH_PAUL_ENG = 1


    NEOSPEECH_MIYU_JPN = 0

    NEOSPEECH_SHOW_JPN = 1

    NEOSPEECH_MISAKI_JPN = 2


    NEOSPEECH_LILY_CHI = 0

    NEOSPEECH_WANG_CHI = 1


    NEOSPEECH_JUNWOO_KOR = 3

    NEOSPEECH_SUJIN_KOR = 8

    NEOSPEECH_YUMI_KOR = 10

```

```

    NEOSPEECH_GYURI_KOR = 11

    NEOSPEECH_DAYOUNG_KOR = 12

End Enum

Public Const NEOSPEECH_KATE_ENG_NAMESTR As String = "Kate"
Public Const NEOSPEECH_PAUL_ENG_NAMESTR As String = "Paul"

Public Const NEOSPEECH_MIYU_JPN_NAMESTR As String = "Miyu"
Public Const NEOSPEECH_SHOW_JPN_NAMESTR As String = "Show"
Public Const NEOSPEECH_MISAKI_JPN_NAMESTR As String = "Misaki"

Public Const NEOSPEECH_LILY_CHI_NAMESTR As String = "Lily"
Public Const NEOSPEECH_WANG_CHI_NAMESTR As String = "Wang"

Public Const NEOSPEECH_JUNWOO_KOR_NAMESTR As String = "Junwoo"
Public Const NEOSPEECH_SUJIN_KOR_NAMESTR As String = "Sujin"
Public Const NEOSPEECH_YUMI_KOR_NAMESTR As String = "Yumi"
Public Const NEOSPEECH_GYURI_KOR_NAMESTR As String = "Gyuri"
Public Const NEOSPEECH_DAYOUNG_KOR_NAMESTR As String = "Dayoung"

'Messages used for speech events
Public Const WM_USER As Integer = 1024

Public Const WIN_SPEAK_MSG As Integer = WM_USER + 12359

Public Const WIN_EXPORT_MSG As Integer = WM_USER + 12360

#End Region

#Region "P/Invoke declarations"

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function GetDFTTSVoice(ByVal iVoiceInfoIdIn As Integer, _
    ByRef pvetOut As Integer, _
    ByVal szVoiceNameOut() As Byte, _
    ByRef piVoiceNameLenOut As Integer, _
    ByRef piVoiceIDOut As Integer, _
    ByRef plangOut As Integer _
    ) As Integer

End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Sub InitDFTTSEngineEx3(ByVal hwndWinOwner As IntPtr, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathKate As
System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal zNeoSpeechDBFolderPathPaul As
System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathMiyu As
System.String, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathShow As
System.String, _

```

```

        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathMisaki As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathLily As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathWang As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathJunwoo As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathSujin As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathYumi As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathGyuri As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechDBFolderPathDayoung As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathKate As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathPaul As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathMiyu As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathShow As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathMisaki As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathLily As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathWang As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathJunwoo As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathSujin As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathYumi As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathGyuri As
System.String, _
        <MarshalAs (UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePathDayoung As
System.String, _
        ByVal psiLoadedEngines() As Integer, _
        ByVal psiLoadedEnginesReturnValues() As Integer _
    )

End Sub

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function UninitDFTTSEngine() As Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function MakeLanguage(ByVal mainlang As Integer, _
    ByVal sublang As Integer _
) As Integer

End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function GetMainLanguage(ByVal lang As Integer) As Integer

End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function GetSubLanguage(ByVal lang As Integer) As Integer

End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function LoadDFTTSUserDict(ByVal iDictIndex As Integer, _
    <MarshalAs (UnmanagedType.LPStr)> ByVal szDictName As System.String, _
    <MarshalAs (UnmanagedType.LPStr)> ByVal szDictFileName As System.String, _

```

```

        ByVal vet As Integer, _
        ByVal lang As Integer, _
        <MarshalAs(UnmanagedType.LPStr)> ByVal szVoiceName As System.String, _
        <MarshalAs(UnmanagedType.LPStr)> ByVal szDictContents As System.String) As Integer
    End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function UnloadDFTTSUserDict(ByVal iDictIndex As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szDictName As System.String, _
    ByVal vet As Integer, _
    ByVal lang As Integer) As Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSSpeak(ByVal hwndWinOwner As IntPtr, _
    ByVal vet As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szVoiceName As System.String, _
    ByVal iVoiceID As Integer, ByVal lang As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szText As System.String, _
    ByVal iPitch As Integer, ByVal iSpeed As Integer, _
    ByVal iVolume As Integer, _
    ByVal iPause As Integer, _
    ByVal iDictID As Integer, _
    ByVal ttTextType As Integer, _
    ByVal ofOutPutFormat As Integer) As Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSPause(ByVal vet As Integer, ByVal lang As Integer) As
Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSResume(ByVal vet As Integer, ByVal lang As Integer) As
Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSStop(ByVal vet As Integer, ByVal lang As Integer) As
Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function DFTTSExportToFileEx(ByVal vet As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szVoiceName As System.String, _
    ByVal iVoiceID As Integer, _
    ByVal lang As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szText As System.String, _
    ByVal iPitch As Integer, _
    ByVal iSpeed As Integer, _
    ByVal iVolume As Integer, _
    ByVal iPause As Integer, _
    ByVal iDictID As Integer, _
    ByVal ttTextType As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szFilePath As System.String, _
    ByVal ffmpegFormat As Integer, _
    <MarshalAs(UnmanagedType.LPStr)> ByVal szAudioEncoding As System.String, _
    ByVal iAudioSamplingRate As Integer, _
    ByVal iAudioChannels As Integer) As Integer
End Function

<DllImport("dftts.dll", CharSet:=CharSet.Auto)> _
Public Shared Function GetDFTTSEngineInfo(ByVal vetVType As Integer, ByVal lang As
Integer, ByVal ttsParam As Integer, ByVal iValue As Integer(),
    <MarshalAs(UnmanagedType.LPStr)> ByVal szNeoSpeechLicFilePath As System.String) As Integer
End Function

#End Region

```

Appendix 2

Controlling Speech with a Standard XML Tag Set

VTML for NeoSpeech VoiceText [™]

(See vtml.pdf or vtml.ps)

SSML

(See <http://www.w3.org/TR/speech-synthesis/>)

Note: Not all tags are supported by all engines.