Archived:NSSTTS Utility Symbian API

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Note: :This API is not part of the public SDK. It can be found in the SDK API Plug-in.

The `CTtsUtility` class is the main interface for speech synthesis. The Text-To-Speech (TTS) utility provides methods for opening and playing text files, descriptors, URLs, etc. It also provides methods for adjusting the volume and getting metadata and a list of supported languages, voices, etc.

Header files

```
#include <nssttsutility.h>
#include <nssttsutilityobserver.h>
```

Link against

```
LIBRARY nssttsutility.lib
```

Example code

1) Derive your class from `MTtsClientUtilityObserver` and implement all pure virtual functions.

```
 void MapcCustomCommandEvent( TInt aEvent, TInt aError );
 void MapcInitComplete( TInt aError, const TTimeIntervalMicroSeconds& aDuration );
 void MapcPlayComplete( TInt aError );
```

2) Initialise the TTS utility.

```
CTtsUtility* iTtsUtility = CTtsUtility::NewL( *this );
```

3) Add the style and data to be played.

```
if ( iTtsUtility->NumberOfStyles() > 0 )
  {
  // Delete style if exists
  iTtsUtility->DeleteStyle( iStyleId );
  }
if ( iParsedText->NumberOfSegments() > 0 )
  {
  // Remove existing segment from parsed text structure
  iParsedText->DeleteSegmentL( 0 );
  }
// Use high quality TTS
```


(C) Copyright Nokia 2014. All rights reserved.

```
iStyle.iQuality = ETtsQualityHighOnly;
// Use language and speaker set in Speech application.
// Also speaking rate and volume is set according to Speech app.
iStyle.iLanguage = KTtsUndefinedLanguage;
// Add style
iStyleId = iTtsUtility->AddStyleL( iStyle );
// Initialise with current text and settings.
iTtsUtility->OpenParsedTextL( *iParsedText );
//The command to start playing
iTtsUtility->Play();
```

3) You can also get and set the volume for `CTtsUtility`.

```
TInt volume;
iTtsUtility->GetVolume( volume );
iTtsUtility->SetVolume( volume );
```

4) Stop playing the text:

```
iTtsUtility->Stop();
```

Supported languages and voices

The following code sample demonstrates how to retrieve the names of languages and voices supported by the TTS engine.

```
#include <ptiengine.h>              // link against ptiengine.lib
```

`CPtiEngine` is used for retrieving the name of supported language (for instance, for display purposes).

Implementation

```
CPtiEngine* ptiEngine = CPtiEngine::NewL();
CleanupStack::PushL( ptiEngine );

RArray<TUid> pluginUids;
CleanupClosePushL( pluginUids );

RArray<TLanguage> languages;
CleanupClosePushL( languages );

RArray<TTtsStyle> voices;
CleanupClosePushL( voices );

// Lists UIDs of available Text-To-Speech plugins
iTtsUtility->ListPluginsL( pluginUids );

for ( TInt i = 0; i < pluginUids.Count(); i++ )
  {
  // pluginUids.operator[]( i ).iUid
  // contains the UID of an available TTS plugin (at index i)
```


(C) Copyright Nokia 2014. All rights reserved.

```
    iTtsUtility->OpenPluginL( pluginUids.operator[]( i ) );
    iTtsUtility->GetSupportedLanguagesL( languages );

    TBuf<64> languageName;
    for ( TInt langIdx = 0; langIdx < languages.Count(); ++langIdx )
      {
      languageName.Zero();
      ptiEngine->GetLocalizedLanguageName( languages[langIdx], languageName );

      // languageName now contains the name of language at index langIdx

      TRAPD( err, iTtsUtility->GetSupportedVoicesL( languages[langIdx], voices ) );
      if ( err == KErrNone )
        {
        for ( TInt voiceIdx = 0; voiceIdx < voices.Count(); ++voiceIdx )
          {
          // voices.operator[]( voiceIdx ).iVoice
          // contains the name of voice for language at index langIdx
          }
        }
      iTtsUtility->Close();
      }
    }
    CleanupStack::PopAndDestroy(4); // closes the 3 arrays, deletes ptiEngine
```

See also

Archived:NSSTTS Common Symbian API

(C) Copyright Nokia 2014. All rights reserved.