

Natural Voices JAVA Demonstration Tool

Contents

- Prerequisites 2
- Sample Applications 2
 - Installation 3
 - Contents 3
 - Running the Demo's 3
- The Code 4

Prerequisites

For this sample Natural Voices Java tool to function correctly the following prerequisites must be in place and available to the desktop running the sample tool.

For all versions of the Demonstration Code

- Java must be installed on the client machine, available from <http://www.java.com>
- Java directory must be included in the client machine PATH variable

For the Standalone Demonstration Code

Component	Action
Natural Voices SDK:	Install desktop\sdk.exe to the client machine
Natural Voices Runtime:	Install desktop\runtime.exe to the client machine
Voice KIT:	Install a voice Kit (att_naturalvoices_v5.1_mike8.exe) to the client machine

For the Client-Server Demonstrations

Component	Action
Server Client:	Install server\client.exe-sdk to a drive that the client machine has access to.
Server SDK:	Install server\SDK.exe-sdk to a drive that the client machine has access to.
Server:	Install server\server.exe to a drive that the client machine has access to.
Voice KIT:	Install a voice Kit (att_naturalvoices_v5.1_mike8.exe) to the client machine

Sample Applications

Folder **NV-API**

Installation

Aside from copying the above to a directory on the client machine, no further installation is required (provided the pre-requisites have been met).

It is recommended to copy the files to the following directories

For Windows:

- **C:** as such the demo tool will be available in **C:\NV-API**

For UNIX/LINUX:

- **Users home directory** as such the demo tool will be available in **~/NV-API**

Contents

Within the folder you will find the following folders and files

- **Naturalvoices:** Contains the compiled Java wrappers/API for Natural Voices, sample applications in both UNIX and Windows
 - **Naturalvoices** Compiled wrappers
 - **Naturalvoices\demos** Compiled sample applications (GUI and Command line)
 - **Naturalvoices\Source Code:** Contains the editable (via notepad or eclipse) source code for the UNIX and Windows Sample App's
- **DemoOutput:** Empty folder that will be used to save transcriptions or wav output files
- **Gui-demo-client.cmd:** Executable file for client-server version of the sample (in windows)
- **gui-demo-standalone.cmd:** Executable file for client-server version of the sample (in windows)

Running the Demo's

Two versions of the sample application are available GUI Based Sample

1. Command Line Sample
2. Client Server Mode

In both cases a standalone and client server version is available for BOTH Windows and UNIX.

	Windows	UNIX
GUI Samples		
Client Server Version	Execute (doubleclick): Gui-demo-client.cmd OR From a cmd line execute javaw naturalvoices.demo.gui.client.Main	From the nv-api folder execute (command is case sensitive): java naturalvoices.demo.gui.client.Main
Standalone Version	Execute (double click) gui-demo-standalone.cmd OR From a cmd line in the nv-api folder execute javaw naturalvoices.demo.gui.standalone.Main	From the nv-api folder execute (command is case sensitive): java naturalvoices.demo.gui.standalone.Main

Command Line Samples		
Client Server Version	From a cmd line in the nv-api folder execute: java naturalvoices.demo.client.Main <naturalvoices-client-directory>	From a cmd line in the nv-api folder execute: java naturalvoices.demo.client.Main <naturalvoices-client-directory>
Standalone Version	From a cmd line in the nv-api folder execute: java naturalvoices.demo.standalone.Main <naturalvoices-sdk-directory>	From a cmd line in the nv-api folder execute: java naturalvoices.demo.standalone.Main <naturalvoices-sdk-directory>

Assumes that the zip file has been extracted to C:\nv-api for windows and home/nvapi for unix
<naturalvoices-client-directory> is the folder into which the client-server version of Natural Voices was installed
<naturalvoices-sdk-directory> is the folder into which the standalone version of Natural Voices was installed
Each Command line version allows for text parameter to be passed by adding any text to the end of the command provided is it encapsulated by double quotes. E.g.: Java naturalvoices.demo.standalone.Main c:\nvsdk "Any custom text can be placed here as long as it is surrounded by double quotes"

The Code

The sample applications (command line and GUI) leverage the java wrappers to perform each of its action a as follows:

Action	Command
To initialize the wrappers	<pre>import naturalvoices.Player; import naturalvoices.StandalonePlayer; import naturalvoices.WordTranscription;</pre>

Action	Command
<p>Declare and initialize variables required by the wrappers</p> <ul style="list-style-type: none"> • textsample (the text to be spoken, transcribed and/or output to wav) • Player (the name of the player that you will use) 	<ul style="list-style-type: none"> • <code>String textSample = "Hi there";</code> • <code>Player player;</code>
<p>Create a new player object</p>	<ul style="list-style-type: none"> • <code>player = new StandalonePlayer(rootDir);</code>
<p>Enable standard error output</p>	<ul style="list-style-type: none"> • <code>player.Verbose = true;</code>
<p>Set player parameters as needed</p> <ul style="list-style-type: none"> • Voice • Latin • Text to play 	<ul style="list-style-type: none"> • <code>player.setVoice("mike16");</code> • <code>player.setLatin1(true);</code> • <code>player.setSourceText(textSample);</code>
<p>Play the Text</p>	<ul style="list-style-type: none"> • <code>player.Play();</code>
<p>Dump the output to a wav file</p>	<ul style="list-style-type: none"> • <code>player.Convert("DemoOutput/tts_standalone_demo.wav");</code>
<p>Transcribe the text with phonetics</p>	<ul style="list-style-type: none"> • <code>List<WordTranscription> transcription = player.Transcribe();</code>
<p>Show the transcription on screen</p>	<pre> PrintWriter transFile; for (int i = 0; i < transcription.size(); i++) { String s = "word: " + transcription.get(i).Word + "", transcription: " + transcription.get(i).Transcription + ""; System.out.println(s); transFile.println(s); } </pre>
<p>List the voices</p>	<pre> System.out.println("Voices:"); String[] voices = player.ListVoices(); if (voices != null) { for(int i = 0; i < voices.length; i++) { System.out.println(" " + voices[i]); } } </pre>