NUANCE PROFESSIONAL SERVICES I Version 2.5 I May 2, 2008

SREC and UAPI on Android RC3 Release

Release Notes

RELEASE NOTES

1 DELIVERABLES

- SREC source code
- Open-source grammar compiler for Ubuntu 6.06 Linux\x86
- Example application source code and makefiles
- Test case with acoustic model, dictionaries, grammar, audio files and a test script
- API reference documentation, SREC User Guide and Release Notes
- Java API, also known as Unified API (UAPI), with reference documentation
- UAPI source code
- Example application SpeakContact based on Google's deprecated SpeechTest demo (dial by name)

2 INSTALLATION PROCEDURE

The package contents can extracted and copied to the "device" directory since the files are in the standard locations. Please refer to the next section for a description of the directories and their contents.

The following should be reviewed first:

- 'SpeakContact' demo application, and source code, in `apps/SpeakContact`
- Docs folder `extlibs/srec/doc`
- For the lower level SREC engine, see source code and makefiles for the `SRecTest` and `SRecTestAudio` sample programs are in `extlibs/srec/srec/test/`
- The data files for the sample programs are in `extlibs/srec/config/`. This directory contains acoustic models, dictionaries, a grammar, audio files and test scripts to run `SRecTest` and `SRecTestAudio`. The contents of this directory are installed on the device in `/system/usr/srec/`.

The Java related files are located:

- Source code: `java/android/android/speech/recognition`
- Command-line, event-driven test applications are under `extlibs/srec/`:
  `java/android/android/speech/recognition/test/EmbeddedRecognizerTest.java`
  `java/android/android/speech/recognition/test/contacts/Main.java`
  `java/android/android/speech/recognition/test/parameters/Parameters.java`
  `java/android/android/speech/recognition/test/robustness1/Robustness1.java`
  `java/android/android/speech/recognition/test/robustness1/Robustness2.java`
  `java/android/android/speech/recognition/test/robustness1/Robustness3.java`
  `java/android/android/speech/recognition/test/voicetags/Voicetags1.java`
- Sample application `apps/SpeakContact/`

3 PACKAGE CONTENTS

The major components are:

`apps/SpeakContact`	SpeakContact sample application (based on Google's app)
`extlibs/srec/`	SREC source code
`extlibs/srec/audio/`	AudioIn audio driver
`extlibs/srec/config/`	See below
`extlibs/srec/doc/`	SREC manuals
`extlibs/srec/java/`	Java test programs
`extlibs/srec/make/`	Common makefiles for SREC examples
`extlibs/srec/portable/`	Portability library
`extlibs/srec/seti/`	SETI runtime G2p module source files
`extlibs/srec/shared/`	Common library
`extlibs/srec/srec/`	Core SREC engine
`extlibs/srec/srec/test/SRecTest/`	Example program
`extlibs/srec/srec/test/SRecTestAudio/`	Example program (interactive with live audio)
`extlibs/srec/tools/`	Utility programs such as grammar compiler, dictionary test, etc
`extlibs/srec/uapi/`	UAPI source files (C++, Java, makefiles)
`extlibs/srec/uapi/doc`	UAPI change log
`extlibs/srec/config/en.us/audio/`	Audio files for tests (training and recognition)
`extlibs/srec/config/en.us/dictionary/`	Dictionaries
`extlibs/srec/config/en.us/g2p/`	Grapheme-to-Phoneme models
`extlibs/srec/config/en.us/grammars/`	Grammars: GRXML and compiled
`extlibs/srec/config/en.us/models/`	Acoustic models
`extlibs/srec/config/en.us/tcp/`	Test scripts (speech files and commands)
`extlibs/srec/config/en.us/wave/`	Audio files for tests
`extlibs/srec/config/en.us/baseline*.par`	Parameter file to configure `SRecTest`
`extlibs/srec/config/en.us/run*.sh`	Example for running `SRecTest` on ARM device
`extlibs/srec/config/shared/`	Common files
`java/android/android/speech/recognition/`	Java source code for UAPI

TEST CASES

[SREC C API]

SRecTest

SRecTest is an application that executes a series of speech recognition operations from pre-recorded audio files and prints relevant recognition results to the console. It is described in more detail in the User Guide. It can be executed using the shell scripts that are installed on the device under `/system/usr/srec/config/en.us`. The shell script `run-bothtags5.sh` shows how to run the program on the Android device.

The provided test case illustrates how to add voicetags (enrolled by voice), texttags (enrolled by text) and perform recognition. In bothtags5, 5 new voicetags are enrolled and added to a slot, then 5 new texttags (with prons) are added to the same slot, and finally recognition is performed using audio files. The grammar also demonstrates the use of the semantic interpreter because two different orthographies are mapped to the same semantic meaning ("traffic" is a synonym for "traffic information").

Build:

```
make SRecTest srec_test_files
```

Execution:

From the `/system/usr/srec/config/en.us` directory run SRecTest as follows:

```
/system/bin/SRecTest -parfile baseline11k.par -tcp tcp/bothtags5.tcp
-datapath audio/
```

Parameters:

`-parfile baseline11k.par`:

- Parameters that can be overridden by the command line (acoustic model, dictionary, TCP file)
- Recognizer parameters (`SREC.* and CREC.*`)

`-tcp tcp/bothtags5.tcp`:

- Commands such as adding voicetags, adding texttags, and performing recognition
- Also lists speech files used for voicetag enrolment and speech recognition

`-datapath audio/`:

- This is the path to the audio files used for voice tag enrolment and speech recognition.
- These are NIST audio files (1024 byte header which can simply be skipped)
- The filenames listed in the TCP script are appended to the data path. It is up to the user to provide a trailing forward slash to the data path, if necessary.

Output:

The output, redirected from stdout and stderr, is available in the following file:

```
out_SHIP_bothtags5.txt
```

Prior to the Alpha 3 release the grammar was specified on the command line:

`-grammar grammars/bothtags5.g2g`:

- This is a compiled grammar created from `bothtags5.grxml` using the grammar compiler tools whose source code is under extlibs/srec/tools (`grxmlcompile` and `make_g2g`).
- Refer to `grammars/run_compile_grammars.sh` to see how this was compiled.

This is now contained in the TCP file, as described in the SREC User Guide.

There is a related shell script `run-bothtags5-from-saved.sh` which runs a variant of the above described test.

SRecTestAudio

SRecTestAudio is an interactive version of SRecTest which works with live audio from the microphone.

Build:

```
make SRecTestAudio srec_test_files

adb sync
```

Execution:

It can be invoked as follows or using run-liveaudio.sh

```
/system/bin/SRecTestAudio -parfile baseline11k.par -tcp tcp/recognize_10_live.tcp
```

where `recognize_10_live.tcp` is a script that performs recognition 10 times. Another single recognition script is provided (recognize_1_live.tcp).

[UAPI Java API]

The Java UAPI is tested by using the EmbeddedRecognizerTest, which is an event-driven Java program that is executed from the command line.

To build:

```
make srec-java-tests srec_test_files
```

To run EmbeddedRecognizerTest:

NOTE: see Appendix A for important information about an change that was required in the command-line invoked Dalvik VM.

- Install files onto device

  ```
  $ adb sync
  ```

- Open a shell to the device (`adb shell`) and go to the test directory to run a shell script.

  ```
  $ adb shell

  # cd /system/usr/srec
  # ./run_chmod.sh
  # ./run_ERT.sh
  ```

- Program output:

  ```
  Create an EmbeddedRecognizer
  Load a grammar
  Add word to grammar slot
  Save the grammar
  Grammar saved
  Recognizing against the first grammar (dynamic add-word)
  Meaning1=DEL Jen_Parker
  RecognizerListener1.onStopped()
  Load the second grammar (pre-populated slot)
  Grammar loaded
  Recognizing against the pre-populated grammar
  Meaning2=DEL Jen_Parker
  ```

The Java UAPI can also be exercised by running the following scripts from /system/usr/srec:

run_contacts.sh
run_parameters.sh
run_robustness1.sh
run_robustness2.sh
run_robustness3.sh
run_voicetags1.sh

or the GUI application in `apps/SpeakContact`, which is based on Google's demo application (formerly SpeechTest).

EXAMPLE GRAMMARS

Example grammars are provided under extlibs/srec/config/en.us/grammars. Brief descriptions are provided at the top of each grammar file.

```
boolean.grxml
bothtags5.grxml
digits.grxml
dynamic-test.grxml
enroll.grxml
homonym_test1.grxml
homonym_test2.grxml
homonym_test3.grxml
homonym_test4.grxml
ipaq_commands.grxml
lookup.grxml
rootslot.grxml
slot_test1.grxml
slot_test2.grxml
```

GRAMMAR COMPILER AND UTILITIES

The grammar compiler and utilities are located in extlibs/srec/tools. Refer to SREC_User_Guide.pdf for more information. Several utilities are included:

- `dictTest`

  Look up prons in huge offline dictionary. Returns long form prons as used with the grammar compiler. Uses G2P as a backup.

- grxmlcompile

  Open-source grammar compiler

- make_cfst

  Creates the grammar compiler's generic.C file. This file is already supplied and does not need to be regenerated unless modifications to the grammar compiler call for such changes.

- make_g2g

  Creates a .g2g grammar binary from grxmlcompile output

- make_ve_grammar (optional, can also have .grxml enrolment grammars)

  Creates a .oice-enrollment grammar

- parseStringTest

  Tests semantic interpretation scripts

- test_g2g

  Tests a .g2g grammar binary created by make_g2g

- test_swiarb

  A tool to examine triphones

Perl script:

- srecres2utd.pl

  Processes output of SRecTest

4 DELIVERY METHOD

Packages will be released through the following FTP account:

```
FTP Site:  ftp://ftp.speechworks.com  or  ftp://ftp3.scansoft.com

Username:  qproject
Password:  5wDFhq9q7

 - Download files from the pub folder,
 - Upload files to the incoming folder.
 - File names should not contain spaces
 - or punctuation other than the period (.)
 - dash (-) or underscore (_) characters.
```

5 RELEASE HISTORY

Detailed Description
Version 1.0.0 03/30/2007 Alpha Release Features: • C API • 11 kHz acoustic models • OSR 1.1 grammar compiler (not open source) • Example shows file-based recognition (no audio driver integration) • Uses system heap for dynamic memory allocation. • Ported to and tested on simulator (Linux\x86) and Sooner phone (Linux\ARM) Limitations: • Grapheme-to-Phoneme (G2P) module for runtime addition of words that are not in dictionaries is not yet included • The API manual may still describe components that have been removed from SREC for this alpha release (threads, memory files, etc). • The API will change between the alpha and beta releases. • SRecTest.c has been restructured and may differ slightly from the code described in the User Guide. • SRecTest might not properly handle abnormal conditions (missing files, invalid settings) FILE: `Nuance_SREC_2007_03_30_alpha.tar.gz`

Version 1.0.1
05/04/2007

Alpha 2 Release

Features:

- Same as previous release but recompiled using May 1, 2007 snapshot of Android code
- Included audio library (libASR_AudioIn.so) for testing SREC with live audio. An alternate version (libASR_AudioInFix.so) gets samples by directly reading /dev/eac.
- Additional test programs SRecTestAudio (interactive test), AudioInRecord (uses audioin library) and AudioTestRecord (uses Android's AudioHardware).

Limitations:

- libASR_AudioIn.so only supports 11.025 kHz.
- Low-level audio driver still hangs.

FILE: `Nuance_SREC_2007_05_04_alpha_2.tar.gz`

Version 1.0.2
05/30/2007

Alpha 3 Release

Features:

- Now includes Grapheme-to-Phoneme (G2P) module for runtime addition of words that are not in dictionaries.
- Modified SRecTest.c and corresponding TCP script. Refer to User Guide.
- Robustness improvements.

Limitations:

- libASR_AudioIn.so only supports 11.025 kHz.
- Low-level audio driver still hangs.
- The API manual may still describe components that have been removed from SREC for this release (threads, memory files, etc).
- The API will change between the alpha and beta releases.
- SRecTest might not properly handle abnormal conditions (missing files, invalid settings).
- G2P does not support digit strings (i.e., "200" is not supported but "two hundred" is).

FILE: `Nuance_SREC_2007_05_30_alpha_3.tar.gz`

Version 1.1.0
06/29/2007

Beta 1 Release

Features:

- Includes Grapheme-to-Phoneme (G2P) module for runtime addition of words that are not in dictionaries.
- New G2P models
- Includes digit string preprocessor (creates phonetic transcription for individual digits in digit string)
- Modified SRecTest.c and corresponding TCP script. Refer to User Guide.
- Robustness improvements.
- Includes draft version of Nuance Unified API (UAPI) Java interface.
- Grammar compiler uses G2P module for pron guessing.
- New utility dicttest_sgc in tools/swi_linux/bin

Limitations:

- libASR_AudioIn.so only supports 11.025 kHz.
- The Java API is subject to change
- SRecTest might not properly handle abnormal conditions (missing files, invalid settings).
- UAPI Speaker class not implemented

Notes:

- Grammar compiler tools moved from tools/swi to tools/swi_linux
- Simulator build no longer included due to lack of thread and audio support

FILE: `Nuance_SREC_2007_06_29_beta.tar.gz`

Version 1.2.0
08/14/2007

Beta 1b Release

Notes:

- Recompiled using 08/10/2007 Android code (BUILD_ID=nuance-27454)
- libSR_AudioIn.so now works with 44.1 kHz mono input audio

FILE: `Nuance_SREC_2007_08_14_beta_1b.tar.gz`

Version 1.3.0

Beta 2 Release

Notes

- Date of release : September 7, 2007
- CVS tag for SREC: Q-BETA-2
- SVN tag for UAPI: beta2
- A complete changelog, with all changes since the Beta release of June 29, 2007, is located at docs/UAPI/changelog.txt
- The binary grammar format changed so the .grxml files need to be recompiled using the current grammar compiler

Limitations

- Does not support Voice Tags
- Does not support recognition parameters
- Some methods do not declare all the exceptions they may throw
- The specification does not dictate the component state if the onError() event occurs
- The recognizer will crash if a grammar is unloaded in the middle of recognition.
- Microphone.setCodec(), DeviceSpeaker.setCodec() and Recognizer.configure() are currently synchronous functions but will become asynchronous in a future release.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.

Known Issues

- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure.

FILE: `Nuance_SREC_2007_09_07_beta_2.tar.gz`

Version 1.4.0

Beta 3 Release

Notes

- Date of release : October 5, 2007
- CVS tag for SREC: Q-BETA-3
- SVN tag for UAPI: beta3
- A complete changelog is located at extlibs/srec/uapi/doc/changelog.txt
- UAPI Java source is now in java/android/android/speech/recognition rather than java/android/uapi/.
- The SREC C and UAPI C++ source is now in extlibs/srec/
- The SREC configuration files (models, grammars) are in extlibs/srec/config/
- The package name was changed from com.nuance.uapi to android.speech.recognition, per Google's request
- Includes several bug fixes for improved stability (see changelog.txt)

Limitations

- Does not support Voice Tags
- Does not support recognition parameters
- Some methods do not declare all the exceptions they may throw
- The specification does not dictate the component state if the onError() event occurs
- The recognizer will crash if a grammar is unloaded in the middle of recognition.
- Microphone.setCodec(), DeviceSpeaker.setCodec() and Recognizer.configure() are currently synchronous functions but will become asynchronous in a future release.
- Recognizer.configure() should not be called more than once.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.

Known Issues

- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure.
- The DeviceSpeaker may crash if an application shuts down while the playback is in progress.
- A crash or missing logging is more likely if the logging level is set to LEVEL_TRACE.
- Possible memory leak of a few 10's of KBytes.

FILE: `Nuance_SREC_2007_10_05_beta_3.tar.gz`

Version 1.5.0

Beta 3.1 Release

Notes

- Date of release : October 5, 2007
- CVS tag for SREC: Q-BETA-3-1
- SVN tag for UAPI: beta3-1
- <same as Beta 3>
- Modified for consolidated shared libraries. All libESR_*.so, libSR_*.so and libUAPI_*.so merged into libUAPI_jni.so based on Michael Glover's changes.

Limitations

- <same as Beta 3>

Known Issues

- <same as Beta 3>

FILE: `Nuance_SREC_2007_10_05_beta_3.1.tar.gz`

Version 1.6.0

Beta 3.2 Release

Notes

- Date of release : October 25, 2007
- CVS tag for SREC: Q-BETA-3-2
- SVN tag for UAPI: beta3-2
- Merged SREC and UAPI code changes from 10/19 Android SDK snapshot (BUILD_ID=nuance-35060)
- Renamed data directory /system/usr/srec.config to /system/usr/srec/config
- Included VoiceDialer.grxml
- Reduced compiler warnings
- Improved stability
- C and C++ test programs SRecTest, SRecTestAudio, UAPI_test and UAPI_SrecTest must be explicitly built.
- Java test programs must be explicitly built (make srec-java-tests)
- Java test programs moved from java/android/android/speech/recognition to extlibs/srec/java.
- Fixed race conditions in Microphone and DeviceSpeaker classes

Limitations

- Does not support Voice Tags
- Does not support recognition parameters
- Some methods do not declare all the exceptions they may throw
- The specification does not dictate the component state if the onError() event occurs
- The recognizer will crash if a grammar is unloaded in the middle of recognition.
- Microphone.setCodec(), DeviceSpeaker.setCodec() and Recognizer.configure() are currently synchronous functions but will become asynchronous in a future release.
- Recognizer.configure() should not be called more than once.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- The grammars are not recompiled during the build. They are prebuilt .g2g's but this should change in future releases.

Known Issues

- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure.
- Possible memory leak of a few 10's of KBytes.
- Local references overflow. A fix is being tested and could be released in Beta 4.0.

FILE: `Nuance_SREC_2007_10_25_beta_3.2.tar.gz`

Version 1.7.0

Beta 4.0 Release

Notes

- Date of release : November 9, 2007
- CVS tag for SREC: Q-BETA-4
- SVN tag for UAPI: beta4
- Merged SREC and UAPI code changes from 11/08 Android SDK snapshot (version 38709 from Michael Glover)

Changes:

- Speed and stability improvements
- Fixed local reference table overflow
- Fixed crash on 2nd Recognizer.configure() call
- Fixed crash on calling Recognizer.stop()
- Fixed Microphone race condition
- Replaced inefficient logger introduced in Beta 3.2
- Disabled JNIKeepAlive for improved efficiency. Console apps must now keep main thread alive to prevent the JVM from shutting down
- SpeechRecorder no longer crashes
- Implemented Recognizer getParam()/setParam() and provided sample Java app Parameters.java
- Robustness1 ran for over 2 days (250,000+ iterations) before it was manually stopped.
- New Robustness2 application for microphone + recognizer testing
- Reduced compiler warnings

Limitations

- Does not support Voice Tags
- Some methods do not declare all the exceptions they may throw
- The specification does not dictate the component state if the onError() event occurs
- The recognizer will crash if a grammar is unloaded in the middle of recognition.
- Microphone.setCodec(), DeviceSpeaker.setCodec() and Recognizer.configure() are currently synchronous functions but might become asynchronous in a future release.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- The grammars are not recompiled during the build. They are prebuilt .g2g's but this should change in future releases.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive was disabled).

Known Issues

- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure.

FILE: `Nuance_SREC_2007_11_09_beta_4.0.tar.gz`

Version 1.8.0

Beta 5.0 Release

Notes

- Date of release : November 30, 2007
- CVS tag for SREC: Q-BETA-5
- SVN tag for UAPI: beta5
- Merged SREC and UAPI changes from 11/26 Android SDK snapshot (version 41004)

Changes:

- Speed and stability improvements
- Includes open-source grammar compiler (extlibs/srec/tools/grxmlcompile) for Ubuntu 6.06 Linux. The SpeechWorks grammar compiler is no longer supported.
- New Robustness3 application for more thorough exercise of API
- Includes Voicetags support
- New Voicetags1 application to exercise voicetags
- New API function to add a list of items to a slot (see addItemList method)
- Added a work around for audio driver bug: unable to set input format immediately after closing the driver since DMA is still active, according to HTC. See ANDROID_AUDIODRIVER_WORKAROUND in device/extlibs/srec/uapi/cpp/audio/linux/source/MicrophoneLINUX.cpp and device/extlibs/srec/uapi/make/uapi/Makefile.common

Limitations

- Voicetags end of speech detection is late. This will be fixed in the next release. It only affects enrollment, not recognition.
- Some methods do not declare all the exceptions they may throw
- The specification does not dictate the component state if the onError() event occurs
- Microphone.setCodec(), DeviceSpeaker.setCodec() and Recognizer.configure() are currently synchronous functions but might become asynchronous in a future release.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- The grammars are not recompiled during the build. They are prebuilt .g2g's but this should change in future releases with Google's help.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive was disabled).

Known Issues

- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure.

FILE: `Nuance_SREC_2007_11_30_beta_5.0.tar.gz`

Version 2.0.0

RC-1 (renamed to Beta 6.0) Release

Notes

- Date of release : December 20, 2007
- CVS tag for SREC: Q-RC-1
- SVN tag for UAPI: rc1
- Merged SREC and UAPI changes from 12/10 Android SDK snapshot (version 42763)

Changes:

- Speed and stability improvements
- Includes new open-source grammar compiler (extlibs/srec/tools/grxmlcompile) for Ubuntu 6.06 Linux\x86. This grammar compiler no longer needs cppdom binaries. Instead, tinyxml is used.
- Faster grammar compilation
- New models (cross-word triphones)
- Voicetags end of speed detection fix.
- Added getAPIversion() to hidden System class (see limitations below)
- Faster audio input processing
- Test programs bug fixes
- Nuance no longer includes VoiceDialer.grxml which is maintained by Google
- xtapxmlconsole test program no longer included in release

Limitations

- Some methods do not declare all the exceptions they may throw. For example the OUT_OF_MEMORY return code will trigger a Java exception that is not documented in the API.
- The specification does not dictate the component state if the onError() event occurs. Example: When onError is called in the Microphone and MFR the sendOnErrorToListener()is executed this function calls the runStopMicrophoneTask() which sets the state of microphone and mfr to IDLE.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- The grammars are not recompiled during the build. They are prebuilt .g2g's but Google can recompile them from source.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive was disabled).
- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback. As soon as the function is out of scope the JNI will delete the results.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure to reset all memory via System.dispose().
- Only remaining compiler warnings are for a package used by the grammar compiler: extlibs/srec/tools/OpenFst. This library was developed at Google Research and is available under an Apache license. See http://www.openfst.org.

FILE: `Nuance_SREC_2007_12_20_rc_1.tar.gz`

Version 2.1.0

Beta 7.0 Release

Notes

- Date of release : January 23, 2008
- CVS tag for SREC: Q-BETA-7
- SVN tag for UAPI: beta7
- Merged SREC and UAPI changes from 01/10 Android SDK version 45418

Changes:

- SREC and UAPI now use Android logger on device
- Delay loading of SREC user dictionary until it is needed
- Merged Mike Glover's optimized dictionary load code
- Added # prefix to *.ok language header (#LANG=EN-US)
- Reduced compiler warnings for third-party library OpenFst
- Reduced /system/usr/srec footprint from 12 MB to 1.9 MB. Now one must "make srec_test_files" to install optional test files.
- More complete checking of dictionary for non-lower-case words
- Speed improvement: compute log tables offline (log_tabl.c, log_add.c)
- Fix: Crash when null string: SrecGrammar.Item(WordItem, 1, (String)null)
- Fix: Crash when null AudioStream passed to EmbeddedRecognizer.recognize
- Fix: Crash on long list passed to addItemList(). Local references were previously leaked.
- Fix: Mispronunciation when there are special characters or two embedded blanks in a name passed to WordItem
- Fix: Pronunciation involving unprintable characters
- Fix: Pronunciation for "oh", now should get fewer insertions
- Fix: Homonym collision problem. No longer abort processing.
- Fix: Some warnings and error messages in SREC core

Limitations

- Some methods do not declare all the exceptions they may throw. For example the OUT_OF_MEMORY return code will trigger a Java exception that is not documented in the API.
- The specification does not dictate the component state if the onError() event occurs. Example: When onError is called in the Microphone and MFR the sendOnErrorToListener()is executed this function calls the runStopMicrophoneTask() which sets the state of microphone and mfr to IDLE.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- The grammars are not recompiled during the build. They are prebuilt .g2g's but Google can recompile them from source.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive is disabled).
- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback. As soon as the function is out of scope the JNI will delete the results.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure to reset all memory via System.dispose().

FILE: `Nuance_SREC_2008_01_23_beta_7.0.tar.gz`

Version 2.2.0

Beta 8.0 Release

Notes

- Date of release : February 14, 2008
- CVS tag for SREC: Q-BETA-8
- SVN tag for UAPI: beta8
- Merged SREC and UAPI changes from 01/10 Android SDK version 45418

Changes:

- Speed improvement: compute log tables offline (log_tabl.c, log_add.c)
- Speed improvement: G2P lookup
- Speed improvement: strcmp for dictionary lookup
- Defer G2P creation until it is needed
- Use run-time (dynamic) allocations for slot memory when needed
  (so that ,addWords=N is now optional)
- Removed G2P fsm_dictionary (large.ok enlarged as necessary)
- Expanded large.ok SREC dictionary
- Grammar compiler grxmlcompile now has –outpath command-line option to specify the output directory.
- Grammars are compiled during build time (make srec_test_files). The release no longer includes pre-compiled grammars *.g2g
- Grammar compiler builds under Mac
- Removed G2GConfiguration
- Updated some Android applications to new Java API (due to G2GConfiguration removal)
- MediaFileReader and MediaFileWriter now operate on WAV files
- Improved audio efficiency
- Defined LOG_TAG “Srec” and “Uapi” for Android logger
- Support for Nuance Mobile Speech Platform (NMSP) network recognizer. libUAPI_nmsp.so should be added to device/config/prelink-linux-arm.map.
- Fix: “pau” and other problems in pronunciations related to periods and quotes
- Fix: SRecTest now uses proper API SR_Grammar and SR_Semantic for checkparse
- Fix: OpenFst compiler warnings

Limitations

- Some methods do not declare all the exceptions they may throw. For example the OUT_OF_MEMORY return code will trigger a Java exception that is not documented in the API.
- The specification does not dictate the component state if the onError() event occurs. Example: When onError is called in the Microphone and MFR the sendOnErrorToListener()is executed this function calls the runStopMicrophoneTask() which sets the state of microphone and mfr to IDLE.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive is disabled).
- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback. As soon as the function is out of scope the JNI will delete the results.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure to reset all memory via System.dispose().

FILE: `Nuance_SREC_2008_02_14_beta_8.0.tar.gz`

Version 2.3.0

Beta 9.0 Release

Notes

- Date of release : March 3, 2008
- CVS tag for SREC: Q-BETA-9
- SVN tag for UAPI: beta9
- Still using 01/10 Android SDK version 45418

Changes:

- Modified grammar makefile so .g2g depends on .grxml (no separate rule for .map)
- Removed network recognizer
- Updated some Android applications (due to removal of network recognizer)
- Fix: SIGSEGV on shutdown
- Fix: unhandled exceptions in callback

Limitations

- Some methods do not declare all the exceptions they may throw. For example the OUT_OF_MEMORY return code will trigger a Java exception that is not documented in the API.
- The specification does not dictate the component state if the onError() event occurs. Example: When onError is called in the Microphone and MFR the sendOnErrorToListener()is executed this function calls the runStopMicrophoneTask() which sets the state of microphone and mfr to IDLE.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive is disabled).
- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback. As soon as the function is out of scope the JNI will delete the results.
- The System class is not meant to be part of the public API. It is used exclusively by the test infrastructure to reset all memory via System.dispose().

FILE: `Nuance_SREC_2008_03_03_beta_9.0.tar.gz`

Version 2.4.0

RC2 Release

Notes

- Date of release : March 14, 2008
- CVS tag for SREC: Q-RC-2
- SVN tag for UAPI: rc2
- Still using 01/10 Android SDK version 45418

Changes:

- Fix: Meaning string max length (349 chars). The engine now reports an error if the string will be truncated.
- Support for changing the enrollment grammar (i.e., enrolling a name in a carrier phrase).
- Code clean-up (including removal of useless files latlib.h lat_desc.h syntax.h acc_sub.c)
- SREC_User_Guide.pdf was updated to reflect min and max values for parameters.

Limitations

- Some methods do not declare all the exceptions they may throw. For example the OUT_OF_MEMORY return code will trigger a Java exception that is not documented in the API.
- The specification does not dictate the component state if the onError() event occurs. Example: When onError is called in the Microphone and MFR the sendOnErrorToListener()is executed this function calls the runStopMicrophoneTask() which sets the state of microphone and mfr to IDLE.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive is disabled).
- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback. As soon as the function is out of scope the JNI will delete the results.
- The System class is not meant to be part of the public API. It can be used by the test infrastructure to reset all memory via System.dispose().

FILE: `Nuance_SREC_2008_03_14_rc_2.tar.gz`

Version 2.5.0

RC3 Release

Notes

- Date of release : May 2, 2008
- CVS tag for SREC: Q-RC-3
- SVN tag for UAPI: rc3
- Still using 01/10 Android SDK version 45418

Changes:

- Fix: slot nomenclature changes, fixes to more gracefully handle larger number or too many slots
- Fix: change associated with homonym bug
- Fix: throw exception on NULL configuration
- Fix: Renamed getBitRate to getBitsPerSample
- Fix: Corrected UALW to ULAW in Codec.java
- Remove obsolete comment about problem with dlopen() when UAPI_LINUX is defined (LibraryLoader.cpp:172)
- Changes associated with logging CMN vector
- The CMN vector can be retrieved by getting the parameter “CREC.Frontend.swicms.cmn”. The value is in the form of an opaque string. This string can be stored and later used by setting the same parameter to this value.

Limitations

- Some methods do not declare all the exceptions they may throw. For example the OUT_OF_MEMORY return code will trigger a Java exception that is not documented in the API.
- The specification does not dictate the component state if the onError() event occurs. Example: When onError is called in the Microphone and MFR the sendOnErrorToListener()is executed this function calls the runStopMicrophoneTask() which sets the state of microphone and mfr to IDLE.
- The recognizer will wait forever if the AudioSource does not feed it audio.
- The DeviceSpeaker class is only a placeholder and does not generate audio output on any device.
- UAPI assumes the application keeps a thread around to prevent the JVM from shutting down. This should be the case with GUI applications. Console applications need to do this themselves (now that JNIKeepAlive is disabled).
- Arguments of event listeners (such as recognition results) may not be used outside the scope of the callback. As soon as the function is out of scope the JNI will delete the results.
- The System class is not meant to be part of the public API. It can be used by the test infrastructure to reset all memory via System.dispose().

FILE: `Nuance_SREC_2008_05_02_rc_3.tar.gz`

6 SUPPORT

- For engineering support:

  Dennis Velasco dennis.velasco@nuance.com +1(781)565-5000

- Speech science:

  Jean Dahan jean.dahan@nuance.com +1(514)904-7800 x2470

- Project manager:

  Andy Wyatt andy.wyatt@nuance.com +1(607) 277-1647

- OHA Relationship manager:

  Nigel Burns nigel.burns@nuance.com +1(781)565-5000

A. APPENDIX A – COMMAND-LINE DALVIK CODE CHANGE

This appendix regards a crash that seems to need a code change in the Dalvik VM (under 9/4 Android SDK)

Nuance had to change *dalvik/cmd/Main.c* to get rid of a crash that prevented an event-driven Java test application (console) from running to completion. This crash only occurs with this "Command-line invocation of the Dalvik VM", as the file says at the top:

```
/*
 * Copyright 2006 Google Inc. All Rights Reserved.
 *
 * Command-line invocation of the Dalvik VM.
 */
```

Nuance commented out the statements related to the DetachCurrentThread() call around line 201:

```
/*printf("Shutting down Java VM\n");*/
if (vm != NULL) {
    /*
     * This allows join() and isAlive() on the main thread to work
     * correctly, and also provides uncaught exception handling.
     */
    ///if ((*vm)->DetachCurrentThread(vm) != JNI_OK) {
    ///    fprintf(stderr, "Warning: unable to detach main thread\n");
    ///    result = 1;
    ///}

    printf("\nJava VM is about to be destroyed\n");
    if ((*vm)->DestroyJavaVM(vm) != 0)
        fprintf(stderr, "Warning: Java VM did not shut down cleanly\n");
    printf("\nJava VM has exited\n");
}
```

Nuance provided some references to back this up.

> http://java.sun.com/javase/6/docs/technotes/guides/jni/spec/invocation.html#detach_current_thread
>
> *"As of JDK/JRE 1.2 , the main thread can be detached from the VM."*
>
> But if you go to the JDK 1.4.2 specification document, it reads:
>
> http://java.sun.com/j2se/1.4.2/docs/guide/jni/spec/invocation.html#wp16108
>
> *"The main thread, which is the thread that created the Java VM, cannot be detached from the VM. Instead, the main thread must call JNI_DestroyJavaVM() to unload the entire VM."*

Without this change Nuance saw that *dalvik/cmd/Main.c* crashed during DetachCurrentThread().

```
Stack Trace:
  ADDR      FUNCTION                FILE:LINE
  6183a6c8  dvmDetachCurrentThread  dalvik/vm/oo/Object.h:633
  6183cd93  dvmAddTrackedAlloc      dalvik/vm/alloc/Alloc.c:189
```

and more stack info:

```
      bebfcbe0  6183a6b5  dalvik/vm/oo/Object.h:633
                          dvmDetachCurrentThread
==>   bebfcbe4  0000c490
      bebfcbe8  00000001
      bebfcbec  00000000
      bebfcbf0  40008860
      bebfcbf4  0000b218
      bebfcbf8  00008cdc  ??:0
                          ??
      bebfcbfc  6186741c  ??:0
                          ??
      bebfcc00  61831b75  dalvik/vm/Jni.c:2640
                          DetachCurrentThread
      bebfcc04  00000000
      bebfcc08  000a0d38
      bebfcc0c  61831b8d  dalvik/vm/Jni.c:2648
                          DetachCurrentThread
      bebfcc10  000086e3  dalvik/cmd/Main.c:201
                          main
```

With the change, *Main.c* proceeeds to `DestroyJavaVM()` which should wait until no more normal (non-daemon) threads exist and the test program ran happily.