Nuance Speech Recognition System
Version 8.5

Installation Guide
## Chapter 3. Migrating existing applications .............................................. 15
- Overview ......................................................................................... 15
  - Overview of new master packages .................................................. 15
  - Choosing a master package ............................................................. 16
  - Accuracy versus speed .................................................................. 17
  - Memory usage ................................................................................ 18
- Upgrading existing applications .......................................................... 19
  - Upgrading voice enrollments ............................................................ 19
  - Upgrading to Verifier 3.5 ................................................................. 20
    - Compatibility between verification packages ................................. 20

## Chapter 4. Licensing the Nuance System software ..................................... 23
- Types of licenses .............................................................................. 23
  - Obtaining a license ........................................................................ 23
  - Developer license ........................................................................... 24
  - Load test license ............................................................................ 24
  - Deployment license ....................................................................... 24
- Starting a license manager ................................................................. 25
  - Multiple licenses ........................................................................... 25
  - Developer license ........................................................................... 25
  - Load test and deployment licenses ................................................... 26
- Setting parameters in a resource file .................................................... 27

## Chapter 5. Nuance audio providers .......................................................... 29
- Overview .......................................................................................... 29
- Audio providers ................................................................................ 29
- Using the native audio provider ......................................................... 30
  - Windows ......................................................................................... 30
  - Unix ............................................................................................... 31
  - Barge-in ......................................................................................... 31

## Chapter 6. Additional software components ............................................ 33
- Databases ......................................................................................... 33
  - Oracle ............................................................................................ 33
  - ODBC ............................................................................................ 34
File system ................................................................. 34
Text-to-speech .............................................................. 34
About this guide

This guide describes how to install the Nuance® Speech Recognition System software (or Nuance System) and test the installation.

Audience

This guide is intended for application developers and integration partners installing the Nuance System. If you are using a different deployment option such as a third-party toolkit or custom platform, see the vendor’s documentation for the installation procedure.

Organization

The guide is organized as follows:

- **Chapter 1** describes the supported configurations and hardware requirements.
- **Chapter 2** describes installing and configuring the Nuance System on Windows and Unix platforms, and includes instructions on how to use the provided demos to test your installation.
- **Chapter 4** provides licensing information required to run Nuance in development, testing, and deployment environments.
- **Chapter 5** provides information about the Nuance audio providers.
- **Chapter 6** describes the database and text-to-speech components that can be used with the Nuance System.

Related documentation

The Nuance System documentation contains a set of developer guides as well as comprehensive online reference documentation. See *Introduction to the Nuance System* for the list of available documents.
Typographical conventions

Nuance manuals use the following text conventions:

italic text Indicates variables, file and path names, program names, program options, web and email addresses, as well as terms introduced for the first time. For example:

Edit the ag.cfg configuration file.

Courier New Indicates method/member functions, parameter names and values, program constants, and onscreen program output. For example:

Nuance recommends that you set the parameter audio.OutputVolume to 255.

> Courier New Indicates commands or characters you type in and the responses that appear on the screen. The > character indicates the MS-DOS command prompt or Unix shell. Everything after this character is intended to be typed in. For example:

> resource-manager

In this example, the text that you actually type at the command line is “resource-manager”

Courier New Indicates a value that you replace. For example:

The usage for the nlm utility is:

> nlm license_key

In this example, license_key is a value that you replace, so the text you actually type could be, for example:

> nlm ncr8-16-100-a-b22-333c4d55eff

Note: The Nuance System runs on both Windows and Unix platforms. Windows syntax is typically used throughout the documentation. If you are running on a Unix platform, substitute Unix syntax. For example, use $NUANCE wherever %NUANCE% appears, and use “/” in place of “\” in path names. Differences in usage or functionality on Windows and Unix platforms are noted where relevant.
Where to get help

If you have questions or problems, technical support is provided through Nuance Technical Support Online, a web-based resource center that includes technical support guides on specific topics, access to software updates, and technical support news. Access to Nuance Technical Support Online is available to members of our partner programs and direct customers of Nuance. If you are a member, go to support.nuance.com to log on.

To submit comments on the documentation, please send email directly to techdoc@nuance.com. Note that no technical support is provided through this email address. Technical support is provided through the Nuance Technical Support Online.
This chapter describes the requirements to install and run the Nuance System.

Supported configurations

The Nuance System supports many configurations—a precise combination of Nuance version, operating system, processor, compiler, audio provider, and if applicable, database, text-to-speech (TTS), and Java implementation.

Table 1 lists the operating systems and telephony environment combinations supported by Nuance. The telephony environment refers to Nuance audio provider software, and specific telephony hardware and software. Nuance audio providers are shipped with the Nuance System. See Chapter 5 for more information.

**Note:** These configurations are available at different times during the release cycle. See Nuance Technical Support Online for what is currently available.

### Table 1: Operating systems and telephony environments

<table>
<thead>
<tr>
<th>Operating system</th>
<th>Intel Dialog</th>
<th>NMS</th>
<th>RTP</th>
<th>SIP</th>
<th>Aculab¹</th>
<th>Brooktrout¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 2000</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>SPARC Solaris 2.8</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Intel (i386) Solaris 2.8</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. Distributed and supported by third-party vendor.

Table 2 lists supported processors, compilers, databases, TTS, and Java implementation.
Hardware requirements

Nuance recommends these minimum requirements for running the Nuance System in a development environment:

- 500 MHz Pentium II or higher processor—Windows 2000
- 400 MHz processor—Unix
- Memory:

Table 2: Supported configurations

<table>
<thead>
<tr>
<th>Operating system (processor)</th>
<th>Compiler</th>
<th>Audio provider</th>
<th>Database</th>
<th>TTS</th>
<th>Java Speech Channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows 2000 with Service Pack 4 (Pentium II or above; non-Pentium CPUs are not supported)</td>
<td>Intel Compiler C++ 5.0</td>
<td>Intel Dialogic</td>
<td>Oracle 9.2</td>
<td>X</td>
<td>Java 2 SDK (v1.3)</td>
</tr>
<tr>
<td></td>
<td>Microsoft Visual C++ v6.x</td>
<td>NMS</td>
<td>Oracle 8.1.7</td>
<td>File system</td>
<td>ODBC: MSQL 2000, IBM DB2 v7.1 (FixPack 3)</td>
</tr>
<tr>
<td></td>
<td>Intel Dialogic</td>
<td>RTP</td>
<td>File system</td>
<td>File system</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aculab&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brooktrout&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intel Dialogic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>NMS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SIP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aculab&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brooktrout&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPARC Solaris 2.8 32-bit plus latest recommended patch cluster shipped by Sun support (Ultra 80/60 and SunBlade 1000)</td>
<td>Sun Forte C++ Edition 6 update 1</td>
<td>NMS</td>
<td>Oracle 9.2</td>
<td>X</td>
<td>Java 2 SDK (v1.3)</td>
</tr>
<tr>
<td></td>
<td>gcc 2.95.2</td>
<td>RTP</td>
<td>Oracle 8.1.7</td>
<td>File system</td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SIP</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aculab&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brooktrout&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intel Dialogic</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NMS</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTP</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SIP</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aculab&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brooktrout&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Intel Dialogic</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NMS</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTP</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SIP</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aculab&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brooktrout&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td>Intel (i386) Solaris 2.8 plus latest recommended patch cluster shipped by Sun support (Pentium II or above)</td>
<td>gcc 2.95.2</td>
<td>NMS</td>
<td>Oracle 9.2</td>
<td>X</td>
<td>Java 2 SDK (v1.3)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTP</td>
<td>Oracle 8.1.7</td>
<td>File system</td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SIP</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aculab&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Brooktrout&lt;sup&gt;1&lt;/sup&gt;</td>
<td></td>
<td></td>
<td>File system</td>
</tr>
</tbody>
</table>

1. Distributed and supported by third-party vendor.
- **Recognition server (standard master packages):** 128 MB RAM for simple grammars and applications, 256 MB or higher RAM for large grammars, complex applications, multiple ports, and so on.

- **Recognition server (extended or enhanced wireless master packages):** 256 MB RAM for simple grammars and applications, 768 MB or higher RAM for large grammars, complex applications, multiple ports, and so on.

- **Recognition client:** Depends on the number of recognition clients and applications running concurrently. You should estimate 4 MB per channel. For example, a 24-channel system would need 96 MB of RAM, in addition to the memory needed by other processors and recognition servers running on the same host.

- **Disk storage:**
  - 300 to 500 MB—Installation and simple applications
  - 1 GB or higher—Large complex applications
  - 9 GB—Development, testing, and tuning large applications (includes recording utterances and logs to improve grammars and diagnose problems)

**Note:** Please contact your Nuance Sales Engineer for help with provisioning before deploying a production system.
Chapter 2
Installation

This chapter describes installing and configuring the Nuance System on Windows and Unix platforms.

Overview

Nuance 8.5 is designed for full backwards compatibility with Nuance 8.0, and is available in two installation packages:

- **Service pack installation**—To be installed on hosts currently running Nuance 8.0.
- **Complete installation package**—To be installed on hosts with no Nuance products previously installed

For procedures on upgrading your applications after installation, see Chapter 3.

Service pack installation

If you are currently running Nuance 8.0, you can install this package by:

1. **Delete these folders:**
   - **On Windows**
     - `%NUANCE%\doc`
     - `%NUANCE%\programming-examples\JSC`
   - **On Unix**
     - `$NUANCE/doc`
     - `$NUANCE/programming-examples\JSC`

2. **Extract the contents of the installation package to your %NUANCE% (on Unix, $NUANCE) directory**
Note: The new master packages are not included with the Service Pack installation but can be downloaded from Nuance Technical Support Online.

Complete installation package

Successfully installing and running the Nuance System on hosts without Nuance 8 requires these basic steps:

1. Install the Nuance System following the instructions for your operating system (see “Installing the Nuance System on Windows” on page 6 or “Installing the Nuance System on Unix” on page 7)

2. Set up your environment (see “Configuring the Nuance System” on page 8)

3. Run the Nuance demos (see “Testing your installation” on page 10)

4. Set up a license manager for running Nuance in a development, testing or deployment environment (see Chapter 4)

5. Choose your audio provider (see Chapter 5)

6. If you are upgrading your installation, migrate your applications (see Chapter 3)

Installing the Nuance System on Windows

To install the Nuance System complete installation package on Windows, follow these steps:

1. If you are upgrading from Nuance 7 or earlier, you need to uninstall the old version before installing the new one:
   a. Save any changes you made to Nuance directories that you want to preserve.
   b. Uninstall the Nuance System using the Control Panel, Add/Remove Programs option.
   c. Delete the old Nuance installation directory.
   d. If you are upgrading from Nuance 7 or earlier, you must manually remove all Nuance settings from the CLASSPATH, INCLUDE, LIB, and PATH system environment variables.

2. If you are installing on a host where no Nuance products have been installed, download the Nuance System ZIP file from Nuance Technical Support Online (support.nuance.com).
3 Extract the zipped files to a temporary directory.

   **Note:** If you are using *WinZip* to unzip this file, make sure that the CR/LF at the end of the lines doesn’t include an extra ^M:

   - From the *WinZip Options* menu, select Configuration→Miscellaneous→Other.
   - Make sure that “Tar file smart CR/LF conversion” is NOT checked. If it is, some binary files will be corrupted with an extra ^M at the ends of lines. Some executables will then crash or not execute properly.

4 Run *Setup.exe* from the directory where you extracted the Nuance files and follow the instructions in the install screens.

5 Towards the end of the installation, you will be offered the option of installing the Nuance Watcher as a service. The Watcher is a tool for monitoring processes across a network and is described in the *Nuance System Administrator’s Guide*.

   If you accept, the Watcher will be installed as a manual service.

6 Restart your PC after installing the Nuance System. You can delete the zip file after completing the installation.

### Installing the Nuance System on Unix

To install the Nuance System complete installation package on Unix:

1 Download the compressed file from the Nuance Technical Support Online website ([support.nuance.com](http://support.nuance.com)).

2 Uncompress this file. If, for example, you downloaded *NUANCE-8-5-0-SPARC.tar.Z* to `/home/downloads`, run:

   ```
   > uncompress /home/downloads/NUANCE-8-5-0-SPARC.tar.Z
   ```

3 Untar the file, for example:

   ```
   > tar xvf /home/downloads/NUANCE-8-5-0-SPARC.tar
   ```

4 Ensure that you have root permission.

5 Run the *INSTALL* script, for example:

   ```
   > /home/downloads/INSTALL
   ```

   You can delete the tar file after completing the installation.
6 After installation, you must source the Nuance System SETUP file to configure your environment. You should set the $NUANCE environment variable in your shell’s .rc file, for example:

a In a C shell, execute:

```bash
setenv NUANCE install_dir
source $NUANCE/SETUP
```

b In a Bourne shell, execute:

```bash
NUANCE=install_dir
export NUANCE; . $NUANCE/SETUP.sh
```

7 Make sure that you have the DISPLAY variable correctly set, as this is necessary to run Nuance graphic utilities.

**Start menu**

The installation updates the Start menu to include shortcuts to the Nuance product documentation and several executables, among them `check-license`, `nuance-master-packages`, `nuance-version`, and `Xwavedit`. These shortcuts are accessible under Start→Programs→Nuance→v8.5.0.

**Installing Java**

If you intend to use NuanceSpeechChannel, you need to install Java 2:

1 Install Sun’s Java 2 Platform, Standard Edition, version 1.3 (Sun product downloads are available at java.sun.com/products/).

2 After installation, make sure that your PATH environment variable includes the Java bin directory (for example, `c:\jdk1.3\bin`). Nuance also recommends that your CLASSPATH environment variable includes the current directory (".").

**Configuring the Nuance System**

The Nuance System installation automatically sets required environment variables for you. You should not need to verify these settings. You should make
sure that you have a temporary directory set up, and under certain conditions, verify your locale setting. These topics are described next.

**Environment variables**

The install process sets the following variables:

- **NUANCE**—To the location where you installed the Nuance System
- **MACHINE_TYPE**—To indicate the platform you are running, for example:
  - win32
  - i386-solaris
  - sparc-solaris
- **CLASSPATH**—To include the following paths:
  - %NUANCE%\java\nsc.jar
  - %NUANCE%\java\vcomsc.jar
  - %NUANCE%\java\swingall.jar
- **PATH, INCLUDE, and LIB** (LD_LIBRARY_PATH on Unix)—To include the necessary Nuance resources

**Temporary directory**

Several Nuance processes use /tmp on Unix platforms or the directory set by the environment variable %TEMP% on Windows to store temporary files. Make sure the temporary directory is set up for your system.

**Locales**

The Nuance System uses your system locale setting to determine how it interprets character data. It does this based on the settings provided by the currently active locale, regardless of the encoding being used to represent the data. (Nuance software does not use other settings provided by the system locale.) This means that you can use any character encoding supported by your environment to generate text files used by the Nuance System, for example:

- Grammar and dictionary files passed to nuance-compile
- nuance-resources files
- Configuration files such as those used by the watcher-daemon or Windows Watcher service
Make sure that the locale setting for your environment is appropriately set for the language of the grammar you’re compiling. For example, if you are compiling a Spanish grammar, the locale should not be set to English.

The Nuance System uses the locale specified by the `LC_CTYPE` environment variable, if defined. If `LC_CTYPE` is not defined, it uses the default locale for the machine on which it is running:

- On Windows, as specified through the Regional Options or Keyboard Control Panel
- On Unix, as specified by the default locale in the environment

If necessary, set `LC_CTYPE` to the correct locale string. On Windows, Nuance still recommends you set this variable, rather than changing the Control Panel setting. `LC_CTYPE` lets you selectively compile certain languages, whereas the Control Panel setting affects all programs.

### Installing language modules

When you install the Nuance System, only the North American English language module is installed. You can download other language modules—they’re packaged in Language Packs—from Nuance Technical Support Online.

The language modules contain the master packages, dictionaries, and verification packages (if supported) for that language. Some language modules include recorded prompts and sample grammars.

The language modules are installed in language-specific directories in:

```
%NUANCE%\data\lang
```

For example:

```
%NUANCE%\data\lang\English.America
```

### Testing your installation

After you have installed the Nuance System, you can run two demos to test your installation. The Nuance System includes a sample application, `Xapp`, that performs simple speech recognition and natural language interpretation.

These demos assume you are using the native audio provider that came with the Nuance System. The native audio provider processes audio data with no additional software or hardware other than what is provided with the platform.
(except a microphone and external speakers if desired). It is a quick way to test your installation.

If you are using a telephony provider, you must initialize the application and then call into it to run the demos.

Before you begin, you’ll need:

- Your Nuance developer license key (the one used in the demos below is not a valid license). You can obtain a developer license from Nuance Technical Support Online.
- A computer with speakers and a microphone (for the Nuance audio provider). PCs also need a Sound Blaster card or other compatible card.

**Demo 1—Simple speech recognition**

In this first demo, you’ll use the Xapp application with the sample digits recognition package, located in %NUANCE%\sample-packages. This package contains a grammar for recognizing numbers 0 to 9, including both “oh” and “zero” for 0.

**Note:** You might want to test that your microphone is working properly in the Sound Recorder window (Start→Programs→Accessories→Entertainment).

**Starting required processes**

Before running Xapp, you need to start two processes: a license manager and a recognition server. Since the license manager assigns licenses to Nuance processes, you’ll start that process first. You can start each process from a separate command prompt window.

1. To start the license manager, run the nlm process passing your developer license key as an argument. The command might look like this:
   
   > nlm ntk85-1000-a-9a99-99aaa9fa9999

2. Start a recognition server specifying the package to be used in the recognition process. Use the Nuance parameter, lm.Addresses, to tell the recognition server where the nlm process is running. For example:
   
   > recserver -package %NUANCE%\sample-packages\digits
   lm.Addresses=machine

   The parameter `machine` identifies the computer running the nlm process on the network, for example, `mickey`. Specify `localhost` if you’re running all processes on the same machine (as this demo assumes).

**Running Xapp**

3. Run Xapp with the digits package:
   
   > Xapp -package %NUANCE%\sample-packages\digits
   lm.Addresses=localhost
The Xapp window appears.

4 When the application has initialized (check the status area), click LISTEN and say a phone number into your microphone or telephone. As you speak, Xapp prints the words you utter in the Xapp Recognized String area. Once you finish speaking, Xapp displays the confidence score of the recognition result.

5 Click PLAY to hear a recording of your utterance or LISTEN to input more utterances. Click QUIT to exit Xapp, and—if you are using a telephone for audio input—hang up.

Demo 2—Natural language understanding

This demo demonstrates speech recognition and natural language understanding based on the voice-mail sample recognition package. The voice-mail package recognizes and interprets simple utterances used to interact with a voice-mail system, such as “call home” or “play message two.”

Starting required processes

The nlm process should still be running from the first demo. If not, start one before continuing.

1 Start a recognition server specifying the voice-mail recognition package.

```bash
> recserver -package %NUANCE%\sample-packages\voice-mail
lm.Addresses=localhost
```

Running Xapp

2 Once the recognition server initializes, run Xapp with the voice-mail package:

```bash
> Xapp -package %NUANCE%\sample-packages\voice-mail
lm.Addresses=localhost
```

3 When the application has initialized, click LISTEN, and try out the grammar by saying into the microphone:

“Get me George on the line!”

Because the voice-mail package provides language understanding support, Xapp prints out an interpretation for the utterance as well as the recognized string. The interpretation is specified by filling predefined slots (such as a person, place, or message number) with values (such as “John,” “home,” or “6”).

For example, the interpretation of the utterance “Get me George on the line” is created from the specification:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>dial</td>
</tr>
</tbody>
</table>
This is how the recognizer interprets the utterance to mean that you want to make a call to a person called George.

To try another utterance, click LISTEN and say:

“Play message number five please.”

The interpretation now looks something like this:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>command</td>
<td>play_msg</td>
</tr>
<tr>
<td>polite</td>
<td>yes</td>
</tr>
<tr>
<td>which_msg</td>
<td>5</td>
</tr>
</tbody>
</table>

The slot/value pairs for this interpretation differ from the interpretation of the previous utterance. The value of the command slot is `play_msg` rather than `dial`. There is no `person` slot because it is irrelevant for the “play a message” operation, but there is a `which_msg` slot, filled with the value “5.” The recognizer also understands that you said “please,” and the interpretation indicates that you were polite.
This release of the Nuance System ships with new and improved American English master packages for speech recognition and verification. To take advantage of these new packages, you will need to migrate your existing applications as described in this chapter.

Overview

This section provides an overview of the new master packages. It also provides guidelines for choosing a new master package.

Overview of new master packages

The Nuance System ships with the following American English master packages for speech recognition and verification:

For speech recognition:

- \textit{English.America.3.3}—Multipass master package, default for the Nuance System
- \textit{English.America.1.3}—Compact model for systems with restricted memory usage

For speaker verification:

- \textit{English.America.digit.1}
- \textit{English.America.digit.1.1}
- \textit{English.America.text.1}
- \textit{English.America.text.1.1}
The English.America.3.3 master package is the default package for the Nuance System. However, both English.America.3.2 and English.America.1.2 master packages will continue to be supported. See the Nuance Grammar Developer’s Guide for information on compiling recognition packages with a master package other than English.America.3.3.

English.America.3.3 and English.America.1.3 are not compatible with previous versions of the Nuance System. Other language modules are available for download from Nuance Technical Support Online.

**Note:** The English.America.2.0 master package is no longer available in Nuance 8.5. Any start-up scripts that refer specifically to English.America.2 or English.America.2.x must be updated. Nuance recommends that you use the latest version of English.America.3.3 shipped with this release. Run `nuance-master-packages` to determine what is installed on your system.

### Choosing a master package

Nuance recommends upgrading to the English.America.3.3 master package, which is the default for Nuance 8.5. The English.America.3.3 multipass master package provides a significant increase in accuracy, performing at half the error rate as English.America.3.2. The English.America.3.3 master package is also suitable for use with large statistical language models, such as those required by Say Anything™ or AccuRoute™.

The enhanced accuracy impacts CPU and RAM utilization, requiring approximately twice the CPU computation as English.America.3.2 and slightly more memory, when comparing default settings. For many applications operating at pruning value higher than the default value for English.America.3.2, the new English.America.3.3 master package can provide better accuracy at the same CPU consumption.

If you are currently using a compact model, such as English.America.1.2, you can upgrade to the English.America.1.3 master package, which performs at about one-quarter fewer errors as English.America.1.2, and even provides accuracy improvements over English.America.3.2.

The English.America.1.3 is appropriate for systems with these constraints:

- Memory usage is constrained to the same footprint of English.America.1.2
- CPU consumption must remain equivalent to that used with either:
  - English.America.1.2 with `rec.Prune < 950`
  - English.America.3.2 with `rec.Prune < 850`
Note: If your deployment uses custom acoustic models produced by Nuance Professional Services, Nuance recommends you compare their performance to English.America.3.3 before upgrading. The comparison can be done by running offline experiments with batchrec on recent data collected from the deployment. In addition, you can re-engage Nuance Professional Services to adapt the English.America.3.3 models to your deployment.

Accuracy versus speed

The plot below shows roughly how error rates for the different packages change as a function of CPU computation when the parameter rec.Pruning is changed, averaged over a number of applications. (The X-axis is in arbitrary units, but can be used to estimate relative computation changes: the fourth vertical grid line represents twice the computation of the second).

The legacy packages, English.America.1.2 and English.America.3.2, are shown as dashed lines, and the new ones, English.America.1.3 and English.America.3.3, are shown as solid lines. The blue lines are the compact models, and the red lines are the extended (default) models. The pruning values are labeled on the curves, and the default pruning values are shown with an asterix (*).
Performance in any specific task will vary, but Nuance recommends the following when migrating applications to the new models:

- **Upgrading to `English.America.3.3` from `English.America.3.2`:**
  - Out-of-the-box, `English.America.3.3` requires about twice the CPU, which doubles the LU requirement
  - At a fixed `rec.Pruning` value, `English.America.3.3` is about 25% slower
  - To retain the same CPU consumption, reduce the `rec.Pruning` value tuned for `English.America.3.2` by 150

- **Upgrading to `English.America.3.3` from `English.America.1.2`:**
  - Out-of-the-box, `English.America.3.3` requires 2.5 times more CPU
  - At a fixed `rec.Pruning` value, `English.America.3.3` is about 33% slower
  - To retain the same CPU consumption, reduce the `rec.Pruning` value tuned for `English.America.1.2` by 200

- **Upgrading to `English.America.1.3` from `English.America.1.2`:**
  - At a fixed `rec.Pruning` value, `English.America.1.3` is about 12% slower
  - To retain the same CPU consumption, reduce the `rec.Pruning` value tuned for `English.America.1.2` by 50

**Note:** If you are using `rec.Pruning > 900` with `English.America.1.3` then you can likely find a more accurate operating point with the same CPU computation by using `English.America.3.3`.

### Memory usage

The table below shows approximate memory usage for the master packages. These calculations are based on a simple `yes/no` grammar, including a dynamic grammar (using the `-enable_jit` option during compilation) and using default parameter settings. With larger grammars, memory usage for the recserver process will be higher depending on the size of the grammar (an additional 150 MB is not uncommon).

The memory requirement for the `recserver` process is based on an 8-connection setup, the default maximum number of concurrent connections. The `compilation-server` process can also handle concurrent connections (default is 5), and its memory used is a function of the type of grammars it compiles.
Upgrading existing applications

Follow these guidelines to upgrade existing applications:

- Update all voice enrollments using the `update-dgdb` tool. See “Upgrading voice enrollments” on page 19 for details.

- Migrate Verifier tuning settings using the import-tuning tool. See “Upgrading to Verifier 3.5” on page 20.

- Recompile all custom recognition packages with the new American English master packages with `nuance-compile` as described in the Nuance Grammar Developer’s Guide.

**Note:** If you want to continue using your current master packages (such as `English.America.3.2`), make sure you specify the version number of the master package when compiling the recognition package. If you do not specify the version number (for example, if you only specify `English.America.3`), the new `English.America.3.3` master package will be used by default. See the Nuance Grammar Developer’s Guide for more information on compiling a recognition package.

Upgrading voice enrollments

Voice enrollments are compatible across releases of a language module. For example, voice enrollments created with `English.America.1.2` or `English.America.3.2` can be used with the new `English.America.3.3` and `English.America.1.3` language modules. For the voice enrollment database records created with one language module to be readable by another, use the `update-dgdb` tool.

The `update-dgdb` tool converts enrollments generated using Nuance 8.0 to be readable with Nuance 8.5 and `English.America.3.3`. Make sure the new language module has been installed before running this tool. The syntax of the tool is described in the Nuance API Reference. The `package` argument on the command line refers to the package which contains the enrollment grammar.
which has been compiled with the new language module. The old language module is not needed.

Upgrading to Verifier 3.5

This section describes how to upgrade to the new verification packages.

Compatibility between verification packages

The new verification packages are available for North American English include:

- English.America.digit.1
- English.America.digit.1.1
- English.America.text.1
- English.America.text.1.1

The English.America.digit/text.1.1 models are a superset of the English.America.digit/text.1 channels, and include landline hands-free and cellular hands-free channels for both genders.

The speaker models (voiceprints) are compatible if you are following one of these upgrade paths:

<table>
<thead>
<tr>
<th>Upgrading from</th>
<th>Upgrading to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Model</td>
</tr>
<tr>
<td>8.0</td>
<td>language.digit.1</td>
</tr>
<tr>
<td>8.0</td>
<td>language.digit.1</td>
</tr>
<tr>
<td>8.0</td>
<td>language.text.1</td>
</tr>
<tr>
<td>8.0</td>
<td>language.text.1</td>
</tr>
<tr>
<td>8.5</td>
<td>English.America.digit.1</td>
</tr>
<tr>
<td>8.5</td>
<td>English.America.digit.1</td>
</tr>
<tr>
<td>8.5</td>
<td>English.America.text.1</td>
</tr>
<tr>
<td>8.5</td>
<td>English.America.text.1</td>
</tr>
</tbody>
</table>

To migrate tuning settings from Verifier 3.0 to Verifier 3.5, you can use the new import-tuning offline tool. While this tool should enable a problem-free
migration. For optimal performance, Nuance recommends re-tuning the system on Verifier 3.5 whenever possible.

**Note:** Using `import-tuning` is required *only* if `sv.SecurityLevel` was previously set to something other than "OFF".

Using Verifier 3.5 with a package that has not been upgraded will produce error messages in the `recserver` logs saying that the package is down-level and should be upgraded. Verification will proceed but there may be some performance degradation.

The usage statement for this tool is:

```
import-tuning
   -package tuned_package
   -sv_package new_package
   -db* database_descriptors
   -voiceprint_list list_of_voiceprints
   -password_length average_password_length
```

Where:

- `tuned_package` is the name of the compiled verification package tuned for Verifier 3.0
- `new_package` is the name of the new verification package for Verifier 3.5 using 3.0 tuning settings
- `database_descriptors` are the relevant options for the source database, depending on the type of database provider
- `list_of_voiceprints` points to a file listing one voiceprint ID per line
- `average_password_length` is the average length of passwords used in the application (typical length is 0.37 seconds per digit)

To use the tool, enter the above input on the command line as described in the online API reference. For example:

```
import-tuning -package MyPackage -sv_package MyUpdatedPackage
   -dbprovider fs -dbclass svdb -dbname database -dbroot
   -voiceprint_list MyVPlist -password_length
   myAveragePasswordLength
```

For more information on the `import-tuning` tool, see the online API reference.
Licensing the Nuance System software

This chapter describes the types of licenses provided by Nuance and how to set them up.

The Nuance System includes a license manager that allows licenses to float across multiple machines on a network. The license manager, or \texttt{nlm} process, runs on a single machine and issues licenses to Nuance processes distributed across the network.

You must run the license manager before you can start other Nuance processes such as recognition servers. You identify the location of the \texttt{nlm} process to other processes using the \texttt{lm.Addresses} parameter.

Types of licenses

Nuance provides three types of licenses:

- Developer
- Load test
- Deployment

The licenses differ from each other primarily in the number of recognition servers and speech recognition channels they support, and the environment they are used in.

Obtaining a license

To obtain a license:
- **Developer license**: See the license information available from Nuance Technical Support Online (support.nuance.com). You must obtain a Nuance 8.5 key—keys issued for Nuance 8.0 or earlier will not work with Nuance 8.5.

- **Load test license**: Send an e-mail to orders@nuance.com and specify “Load test license request” in the subject field.

- **Deployment license**: Request a Nuance Standard License Agreement from your sales or partner manager. To upgrade your Nuance 8.0 for Nuance 8.5, send an email to orders@nuance.com, and include:
  - Either a copy of your key, the purchase order number for current Technical Support, or the purchase order number for the license purchase.
  - The key configuration (for example, 24 ports on one key or 12 ports on 2 keys)

### Developer license

The developer license, also called a *toolkit license*, lets you run the Nuance System in a development environment. This license supports:

- Specific number of speech recognition channels
- Specific number of speaker verification channels
- One instance of executables in the toolkit (such as nuance-compile and batchrec)
- Unlimited number of recognition servers (up to 99999)

The developer license is a permanent anyhost license, meaning you can use it indefinitely on any host machine.

### Load test license

The load test license is used to test your application before going to production. This license supports the number of channels and recognition servers you expect to need for your deployed application. It should be used only in the last stages of development—never with a deployed system. You can also use a load test license to evaluate Nuance System software.

The load-test license is a temporary license, valid for up to three months.
Deployment license

The deployment license is used in a production environment, after you have completed developing and testing your application. The license is generated based on the number of recognition servers, speech channels, and verification channels you require.

The deployment license is a permanent anyhost license.

Starting a license manager

A license manager (nlm process) must be running before you start other Nuance processes. There are two ways to start one, depending on the type of license you have:

- **Developer license**: Specify the license key (a string containing a sequence of letters and numbers) directly on the nlm command line, for example:

  ```shell
  > nlm ntk85-1000-a-9a99-99aa9fa9999
  ```

- **Load test and deployment licenses**: Specify a file containing required license information on the nlm command line, for example:

  ```shell
  > nlm my_license_file
  ```

For instructions, read the appropriate section below.

Multiple licenses

You can also run multiple licenses on one nlm process, by specifying several license strings on the nlm command line. For example:

```shell
> nlm license_code1 license_code2 license_code3 license_code1
```

For an explanation of how multiple licenses are concatenated, see the *Nuance API Reference*.

Developer license

To start a license manager:

1. Run the nlm process at the command prompt, passing your development license key as an argument. For example:

   ```shell
   > ntk85-1000-a-9a99-99aa9fa9999
   ```
The license manager displays the number of licenses it has, followed by the message, **Nuance License Manager ready**, meaning it’s now ready to hand out licenses.

The Nuance license manager **must** be running whenever someone wants to use Nuance software on your network. Nuance recommends that you include the `nlm` command in a startup script (or Windows service) so that it will always be running.

2. Add the `lm.Addresses` parameter to your `nuance-resources.site` file to tell other Nuance processes the location of the machine running the `nlm` process. See “Setting parameters in a resource file” on page 27 for more information.

3. To ensure that everything is working, run the Nuance utility `check-license` on any machine on your network:

```
> check-license
```

For example, `check-license` might display something like this:

```
The following Nuance v8.5.0 licenses are available:

<table>
<thead>
<tr>
<th>product</th>
<th>available/total</th>
<th>expiration date</th>
</tr>
</thead>
<tbody>
<tr>
<td>sp-chan</td>
<td>99999/99999</td>
<td>18-Jan-2038</td>
</tr>
<tr>
<td>dru</td>
<td>99999/99999</td>
<td>18-Jan-2038</td>
</tr>
<tr>
<td>v-chan</td>
<td>99999/99999</td>
<td>18-Jan-2038</td>
</tr>
<tr>
<td>tool</td>
<td>99999/99999</td>
<td>18-Jan-2038</td>
</tr>
<tr>
<td>server</td>
<td>99999/99999</td>
<td>18-Jan-2038</td>
</tr>
</tbody>
</table>
```

**Speaker verification**

If you start a Nuance server with a recognition package that supports verification, the license manager checks out a verification channel license automatically. If your application does not use speaker verification, use recognition packages that do not support verification to prevent licenses from being allocated unnecessarily or causing your Nuance server to fail if the licenses are unavailable.

**Load test and deployment licenses**

Starting a license manager with these licenses requires that you first save the license information in a license file, and then specify that file on the `nlm` command line. After you request a license, Nuance sends you a text file with the license information. It might look something like this:

```
SN:             100
HostLock:       anyhost
Port:           8470
               1/1          server         8500    8599      1-dec-2003
```

---

26     **Nuance Speech Recognition System 8.5**

**Installation Guide**
Nuance License Manager ready to accept requests on port 8470.

The license manager listens for connections on port 8470. “SN” is a Nuance serial number associated with this license code.

To start these licenses:

1. Save the license information in an ASCII text file.

2. Make sure that there is a new line after the last line of code, otherwise the license manager won’t be able to read the file.

   For example, the file must look like this:

   1     server  600 899 02-jan-2002
   <End of File>

   not like this:

   1     server  600 899 02-jan-2002<End of File>

3. Run the nlm command with the location of that file as the argument:

   > nlm license_file

   The license manager will open the file, read the contents, and then be ready to issue these licenses.

4. Add the lm.Addresses parameter to your nuance-resources.site file to tell other Nuance processes the location of the machine running the nlm process. See “Setting parameters in a resource file” on page 27 for more information.

   Before you purchase a deployment license, you need to consider the characteristics of your local network where the Nuance license is scheduled to work. Depending on your requirements, you may want to configure your LAN into subnets, or, as in the case of critical applications, to create some redundancy so that the Nuance Speech Recognition System is available even if one license server goes down.

   Nuance recommends you run a license manager wherever you have a Nuance resource manager running. This will localize all the Nuance resources used by the system in “clusters” without degrading the performance of other utilities, as the nlm process is not CPU intensive.
Setting parameters in a resource file

Using a resource file, you can associate Nuance parameter settings with specific processes. For example, the `lm.Addresses` identifies the machine running the license manager process to other Nuance processes. Instead of setting the parameter on the command line when starting each process, you can set it once in the `nuance-resources.site` file.

To do this, edit the file `%NUANCE%\data\nuance-resources.site` with any text editor and set the `lm.Addresses` line to the machine on your network running the `nlm` process. For example:

```
lm.Addresses=mickey
```

If you are running multiple license managers, list them as a comma-separated list of machines. For example:

```
lm.Addresses=mickey,mouse
```
Chapter 5

Nuance audio providers

This chapter provides a brief introduction to the Nuance audio providers.

Overview

The Nuance System uses software components called audio providers to interface with specific types of audio hardware. There are two basic categories of Nuance audio providers:

- Those that provide audio via a telephony-based interface. These audio providers support telephony actions like answering the phone, detecting hangup, and transferring a call. Most Nuance audio providers fall into this category.

- Those that provide audio via an internal device like the microphone and speaker on a computer, and do not provide telephony features.

The audio provider works as a plug-in component to the Nuance recognition client (which performs recording and audio playback). This enables you to write applications that are independent of the physical audio/telephony interface.

Audio providers

The Nuance System includes the native audio provider, which allows you to use a sound card, microphone, and speakers on your desktop system, and the simulator audio provider, used for testing purposes.

Before you can run the Nuance System with other telephony hardware, you need to configure the appropriate Nuance audio provider. Nuance audio providers are shipped with the Nuance System, including:
- Intel Dialogic
- NMS
- SIP (Session Initiation Protocol)
- RTP (Real-Time Transport Protocol)

For information on configuring required hardware and software, using the audio provider, and specifying configuration parameters, see the *Nuance Telephony Guide*.

**Note:** Aculab also supplies audio providers for integration with Nuance software. For more information, see the Aculab website (www.aculab.com).

### Using the native audio provider

The Nuance System comes with a native audio provider that supports audio input and output through your computer’s audio interface, microphone, and speakers. It requires no additional hardware or software other than what comes with your computer (except a microphone and external speakers if desired—PCs also need a Sound Blaster card). The native audio provider provides a quick way to develop, prototype, and demonstrate applications.

#### Windows

The native audio device on Windows requires a Creative Labs Sound Blaster 16 (SB-16) card or other compatible sound card.

The Windows volume settings and some Nuance audio parameters interact with each other as follows:

- The Nuance `audio.InputVolume` parameter overrides the Microphone volume system setting
- The `audio.OutputVolume` parameter overrides the master wave volume setting and interacts with the master volume as follows:
  - If you set `audio.OutputVolume=0`, you will never hear the output even if the Master Volume is turned all the way up
  - Conversely, if you set `audio.OutputVolume=255` and turn the Master Volume all the way down, you will not hear the output either
  - Nuance recommends that you set `audio.OutputVolume=255` (you can include it in the file `%NUANCE%\data\nuance-resources.site`) and adjust the output with the Master Volume slider
To open the Volume Control dialog, select Start→Programs→Accessories→Entertainment.

**Unix**

On computers running Solaris, you can set up the audio characteristics with the desktop audio interface. However, if you’re running i386 Solaris, you also need to install a Creative Labs Sound Blaster (SB-16) card or other compatible card.

Only the user logged into the console can access the audio device. To give all users access to your Solaris machine’s audio device, regardless of the login user, follow these steps:

1. Log in as root.
2. Edit the file `/etc/logindevperm`.
3. Change `0600` to `0666` in the line:

   ```
   /dev/console 0600 /dev/sound/*
   ```

   You don’t have to restart the machine for this change to take effect. However, the person logged in at the console must log out and log in again.

**Barge-in**

Because it is not telephony-based, the native audio provider does not support barge-in. Make sure that you disable barge-in by setting `client.AllowBargeIn` to `FALSE`. See the *Nuance API Reference* for more information.
Chapter 6

Additional software components

This chapter describes the database and text-to-speech components supported by the current version of the Nuance System.

Databases

The Nuance System uses databases to store dynamic grammars and speaker verification voiceprints. If you plan to use these features, you must install and configure a database. If you are not using these features, no database is required.

For more information about configuring an Oracle, ODBC, or file system database, refer to the *Nuance Application Developer’s Guide*. This guide also describes the *nuance-database-admin* tool which you can use to create or delete a local file system database. You can also use it to configure an existing database from a supported provider to store Nuance data.

Oracle

The Nuance System supports Oracle server and client combinations. Please note these requirements:

- Install the Oracle client shared libraries on all platforms
- On Windows, add the required shared libraries (*oci.dll*, etc.) to your *PATH*
- On Solaris, add the required shared libraries (*libclntsh.so.8.0* and *libwtc8.so*) to your *LD_LIBRARY_PATH*

**Note:** On Solaris, if *libclntsh.so.8.0* is not generated by the Oracle installation process, you’ll need to generate manually in `$ORACLE_HOME/rdbms/demo` with `make -f demo_rdbms.mk generate_sharedlib`. 


ODBC

The Nuance System supports these ODBC databases on Windows:

- Microsoft SQL 2000 Server
- IBM DB2 v7.1 (FixPack 3)

Please note these requirements:

- ODBC 3.0 or higher driver manager (part of the Microsoft Data Access Components SDK 2.5)
- ODBC 2.0 or higher driver (the driver must support the ODBC core API and level 1 API)
- Microsoft ODBC Administrator for configuring the ODBC data source for your database

Your data source must also support:

- ISO 92 standard SQL
- Variable length data columns that can be fetched and updated in multiple pieces
- Row-based record locking via the `SELECT...FOR UPDATE` statement

File system

The Nuance System includes a file system database provider for prototyping and development, enabling you to begin application development without having to purchase and configure a third-party database. Nuance recommends that you deploy your applications with a commercial database such as Oracle or ODBC.

Text-to-speech

The Nuance System architecture is integrated with the Nuance text-to-speech (TTS) product, Nuance Vocalizer. Vocalizer is a separately-licensed software product that provides a sophisticated TTS system, allowing you to create speech applications that synthesize speech from a text stream. See the `Nuance Vocalizer Developer’s Guide` for more information.
Index

A

Aculab audio providers 30
audio providers
  Aculab 30
    using Nuance native 30
audio.InputVolume parameter
  (Nuance) 30
audio.OutputVolume parameter
  (Nuance) 30

B

barge-in
  native audio provider 31

C

character data (interpretation) 9
check-license utility 26
choosing a master package 16
CLASSPATH variable 8, 9
compilers (supported) 2
configurations (supported) 1
Creative Labs Sound Blaster card 30

databases (supported) 2
demos
  natural language understanding 12
  speech recognition 11
deployment license 24
  obtaining 24
  starting 26
developer license 24
  obtaining 23
  starting 25
disk space requirements 2
display variable 8

E

English.America.1.2 16, 17, 18, 19
English.America.1.3 15, 16, 17, 18, 19
English.America.3.2 16, 17, 18, 19
English.America.3.3 15, 16, 17, 18, 19
environment variables
  setting after installation 9
  See also variables

H

help with Nuance ix

I

i386 Solaris
  installing Nuance System 7
  native audio provider 30
installing Nuance System
  Unix 7
  Windows 6
installing Watcher service 7

L

language modules
  downloading 10
    North American English 10
LC_CTYPE variable 10
LD_LIBRARY_PATH variable 9, 33
LIB variable 9
license manager 23
  See also nlm process
licenses 23-27
  deployment 24, 26
  developer 24, 25
  load test 24, 26
  multiple 25
  obtaining 23
lm.Addresses parameter (Nuance) 11, 23
load test license 24
obtaining 24
starting 26
locale
and nuance-compile 9
LC_CTYPE variable 10
setting 10

M
MACHINE_TYPE variable 9
master packages
choosing 16
English.America.1.2 16, 17, 18, 19
English.America.1.3 15, 16, 17, 18, 19
English.America.3.2 16, 17, 18, 19
English.America.3.3 15, 16, 17, 18, 19
overview 15
memory requirements 2
migrating existing NVP applications 15

N
native audio provider
barge-in 31
Creative Labs Sound Blaster card 30
using with Nuance System 30
natural language understanding
demo 12
nlm process 23
check-license utility 26
starting 25
startup script 26
Nuance System
getting help ix
licensing 23-27
testing installation 10-13
NUANCE variable 9
nuance-resources.site file 28, 30

O
obtaining licenses 23
ODBC database 34
operating system requirements 1-3
Oracle database 33

P
parameters
audio.InputVolume (Nuance) 30
audio.OutputVolume (Nuance) 30
lm.Addresses (Nuance) 11, 23
PATH variable 8, 9, 33
processor requirements 2

R
recognition client
memory requirements 3
recognition server
and verification 26
memory requirements 3
starting 11, 12
requirements
audio providers 2
compilers 2
databases 2
disk space 2
memory 2
operating systems 1-3
processors 2
telephony environments 1

S
SETUP file (Unix)
Bourne shell 8
C shell 8
Sound blaster card 30
SPARC Solaris
installing Nuance System 7
native audio provider 30
speaker verification licenses 26
speech recognition demo 11
Standard License Agreement 24
T

technical support ix
telephony environments 1
  See also audio providers
testing Nuance System installation 10-13
text-to-speech 34
third-party software (supported) 33

U

Unix
  installing Nuance System 7
  native audio provider 30
  SETUP file 8
  syntax conventions viii
upgrading
  existing NVP applications 19
  voice enrollments 19

V

variables
  CLASSPATH 8, 9
  DISPLAY 8
  LC_CTYPE 10
  LD_LIBRARY_PATH 9, 33
  LIB 9
  MACHINE_TYPE 9
  NUANCE 9
  PATH 8, 9, 33
verification. See speaker verification
  voice enrollments 19
volume, setting with the native audio
  provider 31

W

Watcher service
  installing 7
Windows
  installing Nuance System 6
  syntax notations viii

X

Xapp
  natural language understanding
demo 12
  speech recognition demo 11