



Do you hear what I hear?

Create applications with speech recognition and synthesis using IBM Speech for Java

Level: Introductory

Satish Swaroop (sswaroop@sbionline.com), Senior Consultant, SBI Inc.

01 Nov 2001

Speech is becoming a new form of human interaction with computers. Adding speech recognition and speech synthesis to your Web applications could be crucial in this increasingly mobile world. Speech interaction allows hands-free computing, access to your computer even when away from your desk, and improves accessibility for disabled users. The Java Speech API is a freely available specification, and IBM Speech for Java is one such implementation. Speech for Java is based on IBM ViaVoice. This article describes Java Speech API and ViaVoice, and shows a real-time application that uses them effectively.

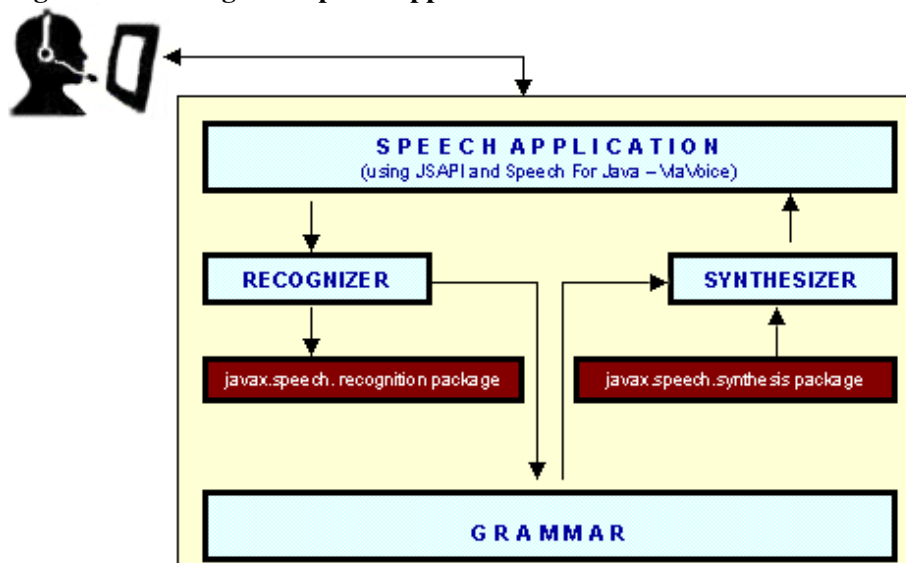
Java Speech API

The Java Speech 1.0 API (JSAPI) specification made it easy for Web developers to create applications that do speech synthesis and voice recognition. JSAPI is cross-platform and supports command and control recognizers, dictation systems, and speech synthesizers. The JSAPI specification is available on Sun's Web site; it includes the Javadoc-style API documentation containing about 70 classes and interfaces in the API (see [Resources](#) for a link).

You can use JSAPI in both applets and applications. Now you can direct a user interface by giving instructions through voice. For example, if you want to complete a form that needs demographic data, you can simply speak the values for different fields. Instead of typing your address, city, state, and ZIP code you can speak these values one by one and appropriate fields get filled in as you proceed. JSAPI can also be effectively used in eCommerce Web sites. If shoppers want to search for a specific product, they can speak the search criteria, and your application searches for and displays it in the browser (or any other device used for searching).

Figure 1 below shows the workings of a speech application. The speech synthesizer and speech recognizer are the instances of the `javax.speech.synthesis` package and `javax.speech.recognition` package. These packages have the basic function for speech synthesis and speech recognition.

Figure 1. Workings of a speech application



Another very important aspect of speech application is *grammar*. A grammar is an object in the JSAPI that controls the recognition process by telling the speaker what words they're expected to say and the patterns in which these words may occur. The biggest advantage of a grammar file is that it makes the recognition faster and more accurate. A sample grammar file is below.

Sample grammar file

```
grammar javax.speech.demo;  
public <sentence> = welcome | Hello | IBM | ViaVoice | Java | Good Job |  
                    Thank you very much | GoodBye;
```

You can add more words or sentences to the grammar file. Note that each word or sentence is separated by the "|" character.

IBM Speech for Java and ViaVoice

IBM implemented the specification of JSAPI and created Speech for Java, which is based on ViaVoice technology that provides continuous dictation (speech recognition) and text-to-speech conversion (speech synthesis). The latest version of ViaVoice includes a recognition engine with improved accuracy, expandable to an active vocabulary of two million words, and other features.

Speech for Java currently supports US English, UK English, Brazilian, Portuguese, French, German, Italian, and Spanish completely, and Japanese for recognition only. Speech for Java runs on Windows and Linux, and can be downloaded from the IBM [alphaWorks](#) Web site.

Your computer should meet the following minimum requirements to run IBM ViaVoice:

- 166MHz Pentium or 150MHz Pentium with MMX, running Windows 95 with 32MB of memory or Windows NT with 48MB, and Sun JDK 1.1.7 or 1.2, *or*
- 166MHz Pentium MMX with 32MB of memory running RedHat 6.1 Linux with IBM JDK 1.1.8 or BlackDown JDK 1.2.2 (with native thread support -- Speech for Java only works with native threads)

Also, be sure that you

- Have installed ViaVoice before unpacking the install package
 - After unpacking the package, set the CLASSPATH to include `\lib\ibmjs.jar`
 - Set PATH (or LD_LIBRARY_PATH on Linux) to include the `\lib` directory
 - Also execute install.bat (or sh install.sh on Linux) to register the IBM engines with the system.
-

Creating a speech application

This section explains the steps involved in creating any speech application. Once you have set up the environment, you are ready to write an application that does speech recognition and speech synthesis. The major tasks in creating any speech application are

- Create a speech synthesis method.
- Create a speech recognizer, which is basically an event listener.
- Create a grammar file that stores all the valid keywords your recognizer should accept.
- Optionally, create a user interface, depending on the type of application you are creating. For example, getting stock quotes might not require a user interface.

Before you run the sample application,

- Be sure Speech for Java and ViaVoice are installed.

- CLASSPATH AND PATH should be set in the environment. For information about setting these, see the README.HTML file provided with the Speech For Java install program.
- You should have a headset ready.
- Create a grammar file with all the valid names and order numbers that you want the Recognizer to capture when the application runs.
- Create an ORDERS table as shown below:

Column Name	Column Type	Description
ORRFNBR	INTEGER NOT NULL	Order Reference Number
SHOPPER_NAME	CHAR (50)	Full Name of the shopper
JOB_STATUS	SMALLINT (2)	Valid Values are:Y or N
ORSTAT	CHAR (1)	Valid values of Order status are: P - Order in pending state C - Order in completed state (order was placed) X - Order was canceled

The following "AllSpeechApp" application demonstrates speech synthesis and speech recognition using JSAPI and IBM Speech for Java. In this application the speaker completes the form by speaking the information required, the application processes the information, then the computer speaks the result.

The application shows a real-time requirement of an eCommerce Web site where a shopper wants to know the current status of the order placed. The shopper provides the information (name, order number, and confirmation to e-mail notification), and the application returns the order status. There are three stages in this application:

1. A data entry form is displayed in [Figure 2](#) below. At this point, the form takes voice input only, and can easily be enhanced for keyboard and mouse support. It is purposely set to voice-only mode so you can see the efficiency of Speech for Java with ViaVoice and JSAPI.

Figure 2. The input form

2. The shopper fills the form by speaking the full name first, followed by order number for which shopper wants the current status, and then the status of the order confirmation e-mail. Let's say the shopper name is *Bob Smith*, order number is *11* and he received the e-mail notification. After the shopper speaks the information the form looks like [Figure 3](#) below.

Figure 3. The speaker speaks, the fields get filled

The shopper can reset the form at any time by saying *Cancel*.

3. Finally, when the shopper says *Submit*, the information in the form is processed and the order status is searched in the database. When the status is found it is written and spoken by the computer for the shopper as shown in [Figure 4](#) below. The computer speaks *Order Number <order no> is <Pending, Cancelled, or Completed>*. In this case, it says "Order Number 11 is Completed" as shown in red.

Figure 4. The output form

Code samples

This section shows the basic framework of the code that performs the functions described above. The declarations and methods are then shown one by one to get a good understanding of the code. If desired, you can download the .jar file from the [Resources](#) section and see all of the code together.

The code framework

```
/**
 *File: AllSpeechApp.java 1.0 2001/09/01
 */
//Import statements
---
---
---

public class AllSpeechApp extends ResultAdapter
```

```

{
/**
 * Common Declarations
 */
    ---
    ---
    ---

/**
 * createComponents - creates a pane and add header label to it.
 */
    public Component createComponents(String printText)
    {
        ---
        ---
        ---
    }

/**
 * Creates a database connection, and gets the
 * order_status from the Orders table.
 * writes the text on the panel.
 * Speaks the text as
 * 'Order number <order number> is <order status>'.
 */
    public void getResult(String orderNum)
    {
        ---
        ---
        ---
    }

/**
 * Listens and stores the spoken text.
 * This is Speech Recognition method.
 * This method also writes the text on the screen using CreateForm() method.
 */
    public void resultAccepted(ResultEvent e)
    {
        ---
        ---
        ---
    }

/**
 * The parameter passed in this method is spoken by the computer.
 * This is Speech Synthesis (Text To Speech) method.
 */
    public void MySpeech(String SpeakText)
    {
        ---
        ---
        ---
    }

/**
 * Creates the form.
 */
    public boolean createForm(String fieldName,String PrintText)
    {
        ---
        ---
        ---
    }

/**
 * Main Method.
 */
    public static void main(String[] args)
    {
        ---
        ---
        ---
    }
}

```

The following section discusses all the commented blocks shown in the above framework.

Import all the necessary packages

```
//Import statements
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.speech.*;
import javax.speech.recognition.*;
import javax.speech.synthesis.*;
import java.util.Locale;
import java.io.FileReader;
import java.awt.Color;
import java.net.*;
import java.sql.*;
```

Common declarations and initializations

```
/**
 * Common declarations and initializations:
 */
JPanel pane;
JLabel label, head2, OrderRefNum_label,
        Name_label, email_label, label_status;
JTextField order_rn, Name;
JButton submit, cancel;
JRadioButton email_yes, email_no;
String newWord1="";
String orderStatus_DB="";
String email_status_y, email_status_n;
static JFrame frame =
        new JFrame("Speech-Text-Speech - Check order Status");
boolean success_Name=false, email_statusNO, email_statusYES;
int ordNum=0; String orderNumber="";

static Recognizer rec;

//Database Related Declarations
Connection theConnection;
ResultSet theResult1;
Statement theStatement1;

// Replace it with your JDBC driver name.
String driver="sun.jdbc.odbc.JdbcOdbcDriver";

// Replace it with your database login id.
String dbuser="satishs";

// Replace it with your database password.
String dbpasswd="SSSSSS";

// Replace it with your database name.
String db="MyLearning";
String driver_db="jdbc:odbc:";
String DRIVERDB=driver_db.concat(db);

// Fonts
Font HEADER_FONT= new Font("ARIAL", 1, 16);
Font NORMAL_FONT= new Font("ARIAL", 0, 6);
```

Create the pane and header label in the pane

```
/**
 * createComponents - creates a pane and add header label to it.
 */
public Component createComponents(String printText)
{
    pane = new JPanel();
    pane.setBorder(BorderFactory.createEmptyBorder(
        10, //top
```

```

        10, //left
        10, //bottom
        10) //right
    );
    label = new JLabel("Find Status of your order : ");
    label.setFont(HEADER_FONT);
    label.setForeground(new Color(0,0,238));
    pane.add(label);
    return pane;
}

```

Create database connection and get order status

```

/**
 * Creates a database connection, and gets the order_status
 * from the orders table.
 * Writes the text on the panel.
 * Speaks the text as
 * 'Order number <order number> is <order status>'.
 */
public void getResult(String orderNum)
{
    try
    {
        //Loading Sun's JDBC ODBC Driver
        Class.forName(driver);
        theConnection = DriverManager.getConnection(DRIVERDB,dbuser,dbpasswd);
        theStatement1=theConnection.createStatement();
        String query="SELECT orstat from orders where orrfnbr="+orderNum;
        theResult1=theStatement1.executeQuery(query);
        while(theResult1.next())
        {
            orderStatus_DB=theResult1.getString("orstat");
        }
        theResult1.close();    //Close the result set
        theStatement1.close(); //Close statement
        theConnection.close(); //Close the connection

        String part1= "Order Number "+orderNum+" is ";
        if(orderStatus_DB.equals("C"))
        {
            orderStatus_DB="Completed.";
        }
        else if(orderStatus_DB.equals("P"))
        {
            orderStatus_DB="Pending.";
        }
        else if(orderStatus_DB.equals("X"))
        {
            orderStatus_DB="Cancelled.";
        }
        String displayOrdStatus=part1.concat(orderStatus_DB);

        //Writes the text on the panel
        createForm("label_status",displayOrdStatus);

        //Speaks the text
        MySpeech(displayOrdStatus);

        //Set the focus back to the Name field.
        Name.requestFocus();

        //Repaint the panel
        pane.repaint();
    }
    catch(Exception e)
    {
        System.out.println("Exception in getResult : " +e);
    }
}

```

Speech recognition method

```

/**
 * Listens and stores the spoken text.
 * This is Speech Recognition method.
 * This method also writes the text on the screen using CreateForm() method.
 */
public void resultAccepted(ResultEvent e)
{
    try
    {
        Result r = (Result)(e.getSource());
        ResultToken tokens[] = r.getBestTokens();

        for(int i=0;i<tokens.length;i++)
        {
            newWord1 = newWord1.concat(tokens[i].getSpokenText());
            newWord1 = newWord1.concat(" ");
        }
        int len_tokens= tokens.length;
        if(len_tokens==2)
        {
            String name = tokens[0].getSpokenText().concat(" ");
            name=name.concat(tokens[1].getSpokenText());
            success_Name = createForm("Name",name);
        }
        if(success_Name)
        {
            order_rn.requestFocus();

            //Sets the order number in the form
            if(len_tokens == 1)
            {
                order_rn.setText("");
                int numstarts = newWord1.indexOf("1");
                orderNumber = newWord1.substring(numstarts,numstarts+2);
                order_rn.setText(orderNumber);
            }

            //Sets the order email notification flag to yes or no.
            int email_status_yes = newWord1.indexOf("Yes");
            int email_status_no = newWord1.indexOf("No");
            if(email_status_yes>0)
            {
                email_status_y = newWord1.substring(email_status_yes,newWord1.length());
                email_yes.setSelected(true);
                submit.requestFocus();
            }
            else if(email_status_no>0)
            {
                email_status_n = newWord1.substring(email_status_no,newWord1.length());
                email_no.setSelected(true);
                submit.requestFocus();
            }
            email_statusYES=email_yes.isSelected();
            email_statusNO =email_no.isSelected();

            /* Checks if shopper said 'Submit'.
             * If so, submit the form by calling getResult() method.
             */

            int submitStarts = newWord1.indexOf("Submit");
            if(submitStarts > 0)
            {
                int numstarts = newWord1.indexOf("1");
                orderNumber = newWord1.substring(numstarts,numstarts+2);

                //Get order status from the Database
                getResult(orderNumber);
                newWord1="";

                Name.setText("");
                order_rn.setText("");
                email_yes.setSelected(false);
                email_no.setSelected(false);
            }

            /**
             * Checks if shopper said 'Cancel'. If so, resets the form.
             */

```



```

        int cancelStarts = newWord1.indexOf("Cancel");
        if(cancelStarts > 0)
        {
            Name.requestFocus();
            newWord1="";
            Name.setText("");
            order_rn.setText("");
            email_yes.setSelected(false);
            email_no.setSelected(false);
        }
    }
}
catch (Exception e2)
{
    System.out.println("\n EXCEPTION in resultAccepted :\n"+e2);
}

//Add the window listener.
frame.addWindowListener(new WindowAdapter()
{
    public void windowClosing(WindowEvent e)
    {
        System.exit(0);
    }
});
frame.setSize(600,275);
frame.setVisible(true);
}

```

Speech Synthesis Method

```

/**
 * The parameter passed in this method is spoken by the computer.
 * This is Speech Synthesis (Text To Speech) method.
 */
public void MySpeech(String SpeakText)
{
    try
    {
        // Create a synthesizer for English
        Synthesizer synth = Central.createSynthesizer(
            new SynthesizerModeDesc(Locale.ENGLISH));

        // Get it ready to speak
        synth.allocate();
        synth.resume();

        //Speak Now...
        synth.speakPlainText(SpeakText, null);

        // wait till speaking is done
        synth.waitEngineState(Synthesizer.QUEUE_EMPTY);

        // Clean up
        synth.deallocate();
    }
    catch (Exception e1)
    {
        System.out.println("EXCEPTION in MySpeech :" + e1);
    }
}

```

To create the form

```

/**
 * Creates the form.
 */
public boolean createForm(String fieldName,String PrintText)
{
    Component contents = createComponents("");
    frame.getContentPane().add(contents, BorderLayout.CENTER);
}

```

```

pane.setLayout(new GridLayout(8,3,2,2));
//Instantiate all components
Name_label      = new JLabel("Enter Your Name :");
label_status    = new JLabel("label_status");
OrderRefNum_label = new JLabel("Enter Order Number :");
email_label     =
new JLabel("Did you get an email confirmation for your order?");
Name            = new JTextField("",30);
order_rn        = new JTextField("",5);
email_yes       = new JRadioButton("Yes");
email_no        = new JRadioButton("No");
ButtonGroup group = new ButtonGroup();
submit          = new JButton("Submit");
cancel          = new JButton("Cancel");
JLabel blank1   = new JLabel(" ");
JLabel head2    = new JLabel("[Voice Only Mode]");

//Set attributes
head2.setForeground(new Color(0,0,238));
submit.setBackground(new Color(0,0,128));
submit.setForeground(Color.white);
cancel.setBackground(new Color(0,0,128));
cancel.setForeground(Color.white);
group.add(email_yes);
group.add(email_no);
Name_label.setForeground(new Color(139,37,0));
OrderRefNum_label.setForeground(new Color(139,37,0));
email_label.setForeground(new Color(139,37,0));

//Add all components
head2.setFont(HEADER_FONT);
pane.add(head2);
pane.setFont(NORMAL_FONT);
pane.add(Name_label);
pane.add(Name);
pane.add(OrderRefNum_label);
pane.add(order_rn);

pane.add(email_label);
pane.add(email_yes);
pane.add(blank1);
pane.add(email_no);
pane.add(submit);
pane.add(cancel);

if(fieldName.equals("Name"))
{
    Name.setText("");
    Name.setText(PrintText);
    order_rn.setText("");
    email_yes.setSelected(false);
    email_no.setSelected(false);
    return true;
}
else if(fieldName.equals("order_rn"))
    order_rn.setText(PrintText);
else if(fieldName.equals("label_status"))
{
    pane.add(label_status);
    label_status.setText(PrintText);
    label_status.setForeground(Color.red);
    label_status.setFont(HEADER_FONT);
}
else
{
    return false;
}
return false;
}

```

Main Method

```

/**
 * Main Method.
 */

```

```

public static void main(String[] args)
{
    AllSpeechApp ASApp = new AllSpeechApp();
    ASApp.createForm("Name", "");
    try
    {
        // Create a recognizer that supports English.
        rec = Central.createRecognizer(new EngineModeDesc(Locale.ENGLISH));

        // Start up the recognizer
        rec.allocate();

        // Load the grammar from a file, and enable it
        //(order_search.gram in this case).
        FileReader reader = new FileReader(args[0]);
        RuleGrammar gram = rec.loadJSGF(reader);
        gram.setEnabled(true);

        // Add the listener to get results
        rec.addResultListener(new AllSpeechApp());

        // Commit the grammar
        rec.commitChanges();

        // Request focus and start listening
        rec.requestFocus();
        rec.resume();
    }
    catch (Exception e3)
    {
        System.out.println("Exception in MAIN method : " + e3);
    }
}

/**
 * Displays the frame first time
 */
frame.setSize(600,275);
frame.setVisible(true);
frame.setResizable(false);
}

```

Testing the sample application

To test the "AllSpeechApp" application, follow these steps:

1. Insert the code in each block of the framework by simply copying and pasting the code into the appropriate block of the framework.
2. Save the program code as `AllSpeechApp.java`.
3. Create a grammar file that has all the valid *shopper names* and *order numbers*. Also add `Yes`, `No`, `Submit`, and `Cancel` in the grammar file so the Speech Recognizer can recognize these commands given by shoppers. Your grammar file should look as follows:

```

grammar javax.speech.AllSpeechApp;
public <sentence> = Meg Carrol | John Pike | Bob Smith |
                  Satish Swaroop | Mark Jones | Joe Jacobson |
                  11 | 12 | 13 | 14 | 15 |
                  Yes | No |
                  Submit | Cancel;

```

Note that line 2 and line 3 contain all the valid shopper names and line 4 contains all the order numbers. Modify the order numbers as they appear in your ORDERS table.

4. Save your grammar file as `order_search.gram`
5. Compile your program by typing `javac AllSpeechApp.java` on the command line.
6. Finally, type `'java AllSpeechApp order_search.gram'` on the command line to start the application.

Summary

I hope my sample application and code showed how easy it can be to implement speech recognition and synthesis using Java Speech API and IBM's Speech for Java. As speech becomes a more common way to interact with computers, the possibilities for speech interaction go way beyond "smalltalk".

Download

Name	Size	Download method
AllSpeechApp.jar		HTTP

→ [Information about download methods](#)

Resources

- [Participate in the discussion forum.](#)
- Download a [jar file](#) that contains the code used in this article.
- Get more details on IBM ViaVoice from [Developer's Corner](#) and [IBM Voice Systems](#).
- Read about [Speech for Java](#) from alphaWorks and then [download the code](#).
- Learn more about the [Sun Java Speech API \(JSAPI\)](#) or get [Java Speech API specifications and publications](#) from Sun's site.
- Download an evaluation copy of the IBM [ViaVoice SDK for Windows](#).

About the author



Satish Swaroop is an IBM-certified e-Business Solution Technologist specializing in implementing IBM WebSphere Commerce Suite and other WebSphere-based solutions. Through his work at SBI, a leading e-business professional services firm specializing in manufacturing, retail, and financial markets, Satish has been successfully enabling customer solutions with wireless and voice-enabled capabilities using Java and IBM technologies. When not working, he spends his time with his wife and daughter Ishita. You can contact Satish at sswaroop@sbionline.com or satishswaroop@email.com.