Speech Input
Introduction

❖ The Android SDK 2.1 includes a voice enabled keyboard. Users can use this voice enabled keyboard for dictating a message instead of typing it.

❖ The user just need to tap the new microphone button on the keyboard and speak.

❖ The Android SDK allows us to integrate speech input directly into our own application.
The RecognizerIntent Class

- This class includes various constants that describe various action that we can use when creating intents for starting and checking the speech recognition activity.

```
PackageManager pm = getPackageManager();
List<ResolveInfo> activities = pm.queryIntentActivities(
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
if (activities.size() != 0)
{
    ...
    ...
}
```

Using this intent it is possible to know whether the speech recognition is supported on our handset.

© 2008 Haim Michael
Start Speech Recognition Activity

We can easily start a speech recognition activity and use the text, it creates out of the speech it processes, in our code.

```java
private void startVoiceRecognitionActivity()
{
    Intent intent = newIntent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(
        RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    startActivityForResult(intent, OUR_REQUEST_CODE);
}
```

This intent will start the speech recognition activity.

When we start an activity and a result is expected, we should associate that start with a specific code in order to allow us identify its result in our implementation for onActivityResult method.

© 2008 Haim Michael
Start Speech Recognition Activity

- The result returned back from the speech recognition activity will be retrieved within our definition for the `onActivityResult()` method.

```java
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode==OUR_REQUEST_CODE && resultCode==RESULT_OK) {
        ArrayList<String> matches = data.getStringArrayListExtra(
            RecognizerIntent.EXTRA_RESULTS);
        ...
    }
}
```
Google Server Side

- When the user taps the microphone button his speech is sent to google servers that perform that speech recognition.
- The google voice search application that is already installed on your device uses the same servers.
- Google's servers currently support English, Mandarin Chinese, and Japanese.
The Language Model

We should specify the language model we want to use when using the speech recognition activity.

```java
Intent intent = newIntent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
                   RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
startActivityForResult(intent, OUR_REQUEST_CODE);
...```
The Free Form Language Model

- The free form language model is appropriate for dictation scenarios.
- The `RecognizerIntent.LANGUAGE_MODEL_FREE_FORM` constant represents this model.
The Web Search Language Model

- The web search language model is appropriate for short search like phrases.
- The RecognizerIntent.LANGUAGE_MODEL_WEB_SEARCH constant represents this model.
Speech Input
Introduction

- The Android SDK 2.1 includes a voice enabled keyboard. Users can use this voice enabled keyboard for dictating a message instead of typing it.
- The user just need to tap the new microphone button on the keyboard and speak.
- The Android SDK allows us to integrate speech input directly into our own application.
The **RecognizerIntent** Class

- This class includes various constants that describe various action that we can use when creating intents for starting and checking the speech recognition activity.

```java
PackageManager pm = getPackageManager();
List<ResolveInfo> activities = pm.queryIntentActivities(
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
if (activities.size() != 0)
{
    ...
    Using this intent it is possible to know whether the speech recognition is supported on our handset.
}
```

04/20/10 © 2008 Haim Michael
We can easily start a speech recognition activity and use the text, it creates out of the speech it processes, in our code.

```java
private void startVoiceRecognitionActivity()
{
    This intent will start the speech recognition activity.
    Intent intent = newIntent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
    intent.putExtra(
        RecognizerIntent.EXTRA_LANGUAGE_MODEL,
        RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
    startActivityForResult(intent, OUR_REQUEST_CODE);
}
```

When we start an activity and a result is expected, we should associate that start with a specific code in order to allow us identify its result in our implementation for onActivityResult method.
Start Speech Recognition Activity

- The result returned back from the speech recognition activity will be retrieved within our definition for the onActivityResult() method.

protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if(requestCode==OUR_REQUEST_CODE && resultCode==RESULT_OK) {
        ArrayList<String> matches = data.getStringArrayListExtra(
            RecognizerIntent.EXTRA_RESULTS);
        ...
    }
}

04/20/10 © 2008 Haim Michael
Google Server Side

- When the user taps the microphone button his speech is sent to google servers that perform that speech recognition.
- The google voice search application that is already installed on your device uses the same servers.
- Google's servers currently support English, Mandarin Chinese, and Japanese.
The Language Model

- When using the speech recognition activity we should specify the language model we want to use.

```
... Intent intent = newIntent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);
intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,
    RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);
startActivityForResult(intent, OUR_REQUEST_CODE);
...```

04/20/10 © 2008 Haim Michael
The Free Form Language Model

- The free form language model is appropriate for dictation scenarios.
- The RecognizerIntent.LANGUAGE_MODEL_FREE_FORM constant represents this model.
The Web Search Language Model

- The web search language model is appropriate for short search like phrases.
- The `RecognizerIntent.LANGUAGE_MODEL_WEB_SEARCH` constant represents this model.