

Escher Voice Command Developer Guide

Introduction

Welcome to the Escher Innovation's Voice Command Software Developer Guide. By the end of this guide, you will be able to integrate your own custom voice commands into your app! Since the Escher Innovation's Voice Command API utilizes standard Android framework functionality, no external libraries are required. The Escher Innovation's Voice Command API is designed to be as simple as possible, allowing you, the developer, to integrate voice command functionality into your app as quickly as possible.

Architecture Overview

The Escher Innovation's Voice Command API utilizes an Android service-based architecture that allows multiple apps to utilize a single instance of the built-in speech recognition engine (powered by Think-A-Move's SPEAR SDK). Apps communicate with the Speech Recognizer service by broadcasting predefined Android intents and receive information from the Recognizer Service by implementing an Android Broadcast Receiver and intent filter. For more information on using Android intents and Broadcast Receivers, read here:

<https://developer.android.com/reference/android/content/Intent> ,
<https://developer.android.com/reference/android/content/BroadcastReceiver>

Defining Custom Voice Commands

Voice User Interfaces (VUI) provide a unique experience for software users, allowing hands-free interaction with their systems. A well-designed VUI can also enhance the user experience by providing a more intuitive way of accessing app functions and information.

To define your list of Voice Commands, create a string array containing the commands. Here is an example Voice Command List:

```
private String[] voiceCommands = {  
    "THIRD PARTY APP", "FOURTH PARTY APP", "HOME BASE"  
};
```

When creating voice commands for your app, be sure to utilize intuitive language that is clear in its intended function (i.e. "New Message" not "New"). Also, be sure to follow these guidelines to maximize accuracy and responsiveness:

- Before adding voice commands to your app, study your application to determine which actions or workflows would benefit from voice control. Simply voice-enabling all functions is almost never an effective strategy.
- Once the appropriate software actions have been identified, try vocalizing your intent the way that you would speak if another person were operating the system for you. Focus on what the result of the software action is, not on the individual steps needed to get there. What is the user asking for? Use this approach to guide your word choices.
- The Speech Recognizer Service can recognize voice commands that are surrounded by additional words (i.e. the command **Open Window** can be triggered by saying "Open Window" or

“Open Window Now” or “Please Open Window”). Keep your voice commands to 2-3 words if possible

to maximize accuracy and responsiveness. Short commands (< 3 syllables) can lead to false-positive

recognition and should be avoided.

- The Escher Speech Service implements a set of Global Commands that are used for system level functions. You may not use any commands from this set in your Voice Command List. If the Speech Recognizer service finds one of these commands in your Voice Command List, it will refuse to listen for any of your commands. The list of Global Commands is defined in the following string array:

```
private static String[] DEMO_COMMANDS = new String[] {  
    "SHOW MENU",  
    "HIDE MENU",  
    "FILES APP",  
    "HOME SCREEN",  
    "APP STORE",  
    "REMOTE HELP",  
    "GALLERY",  
    "EXPLORER",  
    "VOLUME UP",  
    "LEVEL UP",  
    "VOLUME DOWN",  
    "LEVEL DOWN",  
    "FLASHLIGHT ON",  
    "FLASHLIGHT OFF",  
    "LEVEL ONE",  
    "LEVEL TWO",  
    "LEVEL THREE",  
    "LEVEL FOUR",  
    "RECOGNITION",  
    "CENTER MAP",  
    "OPEN CAMERA",  
    "SCROLL UP",  
    "SCROLL DOWN",  
    "SCROLL LEFT",  
    "SCROLL RIGHT",  
    "START RECORDING",  
    "STOP RECORDING",  
    "TAKE PHOTO",  
    "TAKE PICTURE",  
    "SCREEN SHOT",  
    "SLEEP MODE",  
    "WAKE UP",  
    "MUTE SOUND",  
    "MUTE OFF",  
    "INTERNET",  
    "BROWSER",  
    "BRIGHTNESS DOWN",  
    "BRIGHTNESS UP",  
    "BRIGHTNESS MAX",  
    "BRIGHTNESS HIGH",  
    "BRIGHTNESS MEDIUM",  
    "BRIGHTNESS LOW",  
    "WAKE UP",  
    "SHUT DOWN THIRD EYE",  
    "SYSTEM SETTINGS",  
    "BLUE TOOTH SETTINGS",  
}
```

```

"WIRELESS SETTINGS",
"VOICE TYPING",
"VOICE COMMANDS OFF",
"SHOW COMMANDS LIST",
"CURSOR UP",
"CURSOR DOWN",
"CURSOR LEFT",
"CURSOR RIGHT",
"CURSOR SELECT",
"BACK BUTTON",
"RECENT APPS",
"CLOSE APP",
"CLEAR APPS",
"CAMERA ZOOM IN",
"CAMERA ZOOM OUT",
"OPEN THERMAL",
"UNLOCK THIRD EYE",
"ROTATION ON",
"ROTATION OFF",
"HEAD MOTION ON",
"HEAD MOTION OFF",
"GAZE CLICK ON",
"GAZE CLICK OFF",
"MOTION SELECT",
"VOICE DIAL",
"ACCEPT CALL",
"REJECT CALL",
"ANDROID DEBUGGING ON",
"ANDROID DEBUGGING OFF",
"OPEN STORE",
"OPEN WORK SPACE",
"LAUNCHER LIGHT MODE",
"LAUNCHER DARK MODE",
"ABOUT THIRD EYE DEVICE",
"OPEN APP DRAWER",
"SHOW ALL APPS",
"MY APPLICATIONS",
"CLOSE APP DRAWER",
"SHOW DESKTOP",
"CHECK THIRD EYE UPDATE",
"DOWNLOAD THIRD EYE UPDATE NOW",
"DOWNLOAD THIRD EYE UPDATE LATER",
"INSTALL THIRD EYE UPDATE NOW",
"INSTALL THIRD EYE UPDATE LATER",
"COMPLETE THIRD EYE UPDATE",
1;

```

Starting Voice Commands

To start listening for the commands in your Voice Command List, you can create an intent with the action defined as “com.escherinnovation.api.voicecommand”, with extras containing your list and an instruction for the Speech Recognizer Service. Here is an example:

```

Intent intent = new Intent("com.escherinnovations.api.voicecommand");
intent.putExtra("instructions", "start");
intent.putExtra("commands", voiceCommands);
sendBroadcast(intent);

```

The Speech Recognizer Service will receive your intent, and check for any Global Commands in your list. If this check is passed, your Voice Command List is added to the list of available

commands (*Note: When sending your voice command list for the first time, there will be a slight delay. For all subsequent times, until the list changes, the recognizer will respond immediately.*) The Speech Recognizer will display a toast, “LISTENING FOR APP COMMANDS”, when it is ready.

Receiving Voice Command Results

The Speech Recognizer Service utilizes a wakeup word workflow, similar to other VUIs (Amazon Echo, Google Home, Siri). To activate voice commands, speak the wakeup phrase, “Okay Escher.” The system will display a toast, “WAKEUP PHRASE RECOGNIZED”, and play a tone to let you know that it is listening. Speak one of the commands in your Voice Command List. When the Speech Recognizer Service recognizes your command, it broadcasts an intent containing the recognized command. To receive this intent, your app can implement a Broadcast Receiver. Implement the Broadcast Receiver in the on-Resume method in your app. Here is an example Broadcast Receiver implementation:

```
IntentFilter filter = new
IntentFilter("com.escherinnovations.recognizerservice.commandresult");
recognitionResultReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
        String listName = intent.getExtras().getString("listName");
        String result = intent.getExtras().getString("result");
        if (listName.equals("appCommandsList")){
            switch (result){
                case "THIRD PARTY APP":
                    // Do something
                    break;
                case "FOURTH PARTY APP":
                    // Do something
                    break;
            }
        }
    }
};
registerReceiver(recognitionResultReceiver, filter);
```

Note: All voice command results are broadcast in all caps, regardless of how they appear in your original commands list.

In your Broadcast Receiver’s onReceive method, you can extract the information contained in the received intent. Based on this information, you now know that one of your commands has been recognized by the Speech Recognizer Service. Based on the command that has been recognized, you can now perform whatever action you would like to associate with the recognized command.

Life-cycle Management

Since the Speech Recognizer Service only needs to listen for your commands while your app is open, it is important to make sure that your app communicates when the commands are no longer needed. For example, when your app closes, the Speech Recognizer Service should no longer listen for your commands. You also need to unregister your Broadcast Receiver when

your app is no longer active to conserve system memory. In your app's onPause and onDestroy methods, implement the following code:

```
Intent intent = new Intent("com.escherinnovations.api.voicecommand");
intent.putExtra("instructions", "stop");
sendBroadcast(intent);
unregisterReceiver(recognitionResultReceiver);
```

Using the Audio Input Device

Due to Android OS limitations, the audio input device (ie microphone) can only be utilized by one app/ process at a time. Since the Speech Recognizer service is always running, it is always using the audio input device while listening for voice commands. If your app requires the use of the audio input device, in addition to requesting the device from the Android framework (See Android SDK), you will also need to stop, and then restart, the Speech Recognizer. Before requesting the audio input device, use the following code to temporarily stop the Speech Recognizer:

```
Intent intent = new Intent("com.escherinnovations.api.voicecommand");
intent.putExtra("instructions", "interrupt");
sendBroadcast(intent);
```

When finished using the audio input device, use the following code to restart the Speech Recognizer:

```
Intent intent = new Intent("com.escherinnovations.api.voicecommand");
intent.putExtra("instructions", "resume");
sendBroadcast(intent);
```

The Speech Recognizer will display a toast after receiving each intent. The speech recognizer will remember its previous state and will return to the same mode/command list before it was stopped. It is also recommended to place a brief pause (1-2 seconds) in between stopping the recognizer and accessing the audio input device. This will allow the audio device to be fully released and will avoid issues where the audio device could malfunction.

Additional Notes

- If your app starts while the Speech Recognizer Service is in wakeup mode, your commands will be added to the Current Command List, but the user will still be required to speak the wakeup phrase, "Okay Escher", before your commands will be available to them.
- The Speech Recognizer Service can understand single letters and numbers (A, B, C, 1, 2, 3), but not most pronounceable acronyms (i.e. NHTSA, USARPAC, etc.) and some obscure jargon. For additional support, contact Escher Support at developers@escher-innovations.com