

iTUNES CONTROL INTERFACE*

Blake Brogdon
Thomas Deitch
Kyle Barnhart
Britt Antley

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 2.0[†]

1 iTunes Control Interface

To accomplish anything useful with our signal processing, we need to drive an application. We chose iTunes, a popular multiplatform program for playing digital music files, as our example application for several reasons. First, the ability to control iTunes through external means is well documented and fairly simple to achieve. More important, however, is the aspect of music or audio playback in a room while simultaneously listening for whistles.

1.1 Java-COM

iTunes is both the system to control as well as the dominant source of noise to combat. Based on the profile of the whistle detected, whether it be increasing or decreasing in frequency, we will skip to the next or previous track in iTunes. To accomplish this, LabVIEW makes a call to a Java program that interacts with iTunes by accessing the Component Object Model (COM) interface of Windows. The JACOB¹ library serves as an intermediary to complete this task. JACOB² is a native Java-COM bridge, allowing Java programs to simultaneously run and interact with system applications. Once the COM interface receives instructions from LabVIEW, it then interacts with iTunes and changes tracks appropriately.

*Version 1.2: Jan 23, 2008 8:36 am -0600

[†]<http://creativecommons.org/licenses/by/2.0/>

¹<http://danadler.com/jacob/>

²<http://danadler.com/jacob/>