APPLICATION NOTE

4D SYSTEMS

TURNING TECHNOLOGY INTO ART

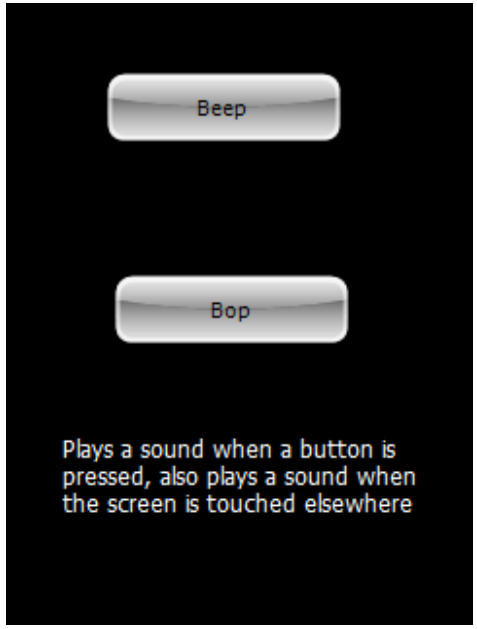
ViSi-Genie Magic Beep Bop

Document Date: 15th May 2015

Document Revision: 1.01

Description

This application note primarily shows how Magic Touch and Magic Release are used to implement a ViSi-Genie project that allows to play different sounds when touching different objects in the display. The Magic Touch and Magic Release objects are under the Genie Magic pane in Workshop 4 Pro. They contains a 4DGL code and they can be used when the detection of touch in an application is needed.



Note 1: The ViSi-Genie project for this application note is the demo "**BeepBop**", which is found in Workshop. Go to the File menu -> Samples ->ViSi Genie Magic (Picaso/Diablo16) ->**BeepBop.4DGenie**.

Note 2: Workshop Pro is needed for this application.
Before getting started, the following are required:

- Any of the following 4D Picaso **touch** display modules:

  uLCD-24PTU uLCD-32PTU uLCD-43(PT/PCT)

  uLCD-28PTU uLCD-32WPTU

  andother superseded modules which support the ViSi Genie environment
- The target module can also be a Diablo16 **touch** display

  uLCD-35DT uLCD-70DT

  uLCD-43DT uLCD-43DCT

  Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor.
- 4D Programming Cable or µUSB-PA5
- micro-SD (µSD) memory card
- Workshop 4 IDE with Pro License (installed according to the installation document)
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

In the past it was not possible to detect touch pressed or released in ViSi Genie and also to be able to control a sound object and detect it's status if playing or not. The application shown in this document checks if a sound is not playing. If there is no sound playing then it will check what button is touched and then plays the corresponding sound. Three sounds are used here, first one is when 'BEEP' button is pressed, second one is when 'BOP' button is pressed and lastly when anywhere on the screen is pressed besides the buttons. If the touch is released then the sound will stop playing.

Setup Procedure

For instructions on how to launch Workshop 4, how to open a ViSi-Genie project, and how to change the target display, kindly refer to the section "**Setup Procedure**" of the application note:

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

Create a New Project

Create a New Project

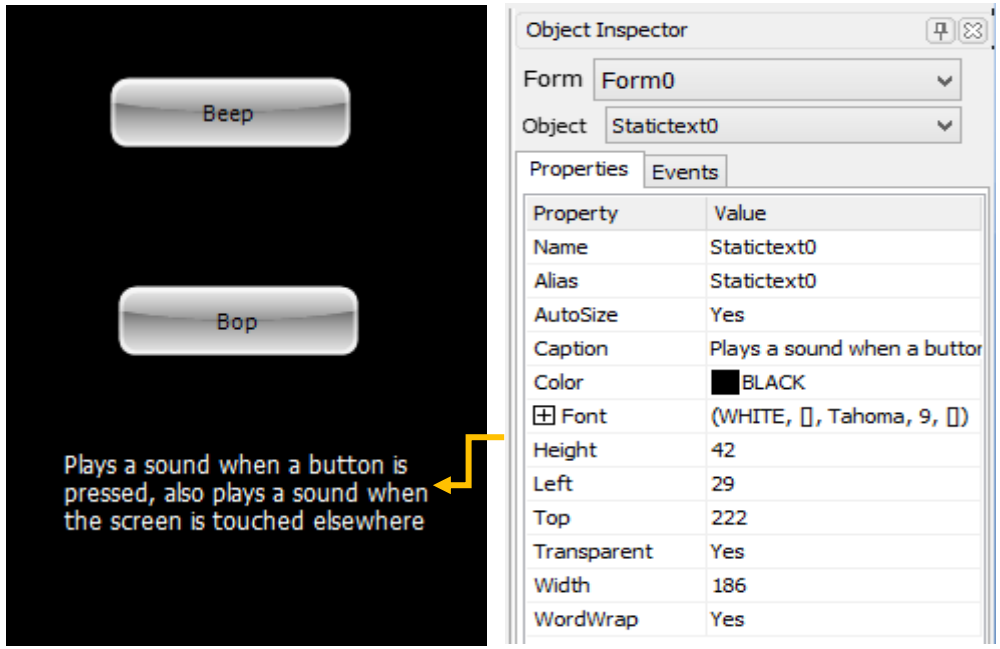
For instructions on how to create a new ViSi-Genie project, please refer to the section "**Create a New Project**" of the application note

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

Design the Project

Add Static Text Objects

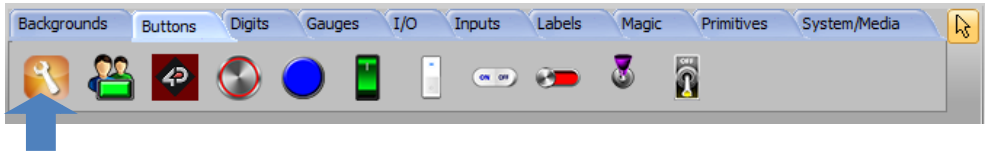
A static text object is added to Form0. This is **Statictext0**.



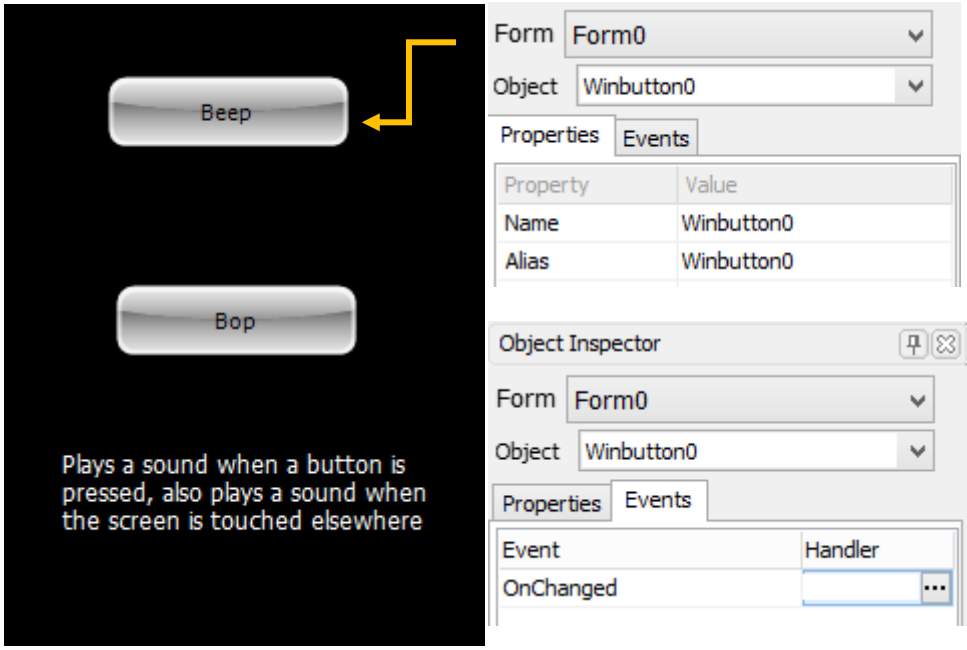
To know more about static text objects, their properties, and how they are added to a project, refer to the application note ViSi-Genie Labels, Text, and Strings

Add Two Fancy Buttons

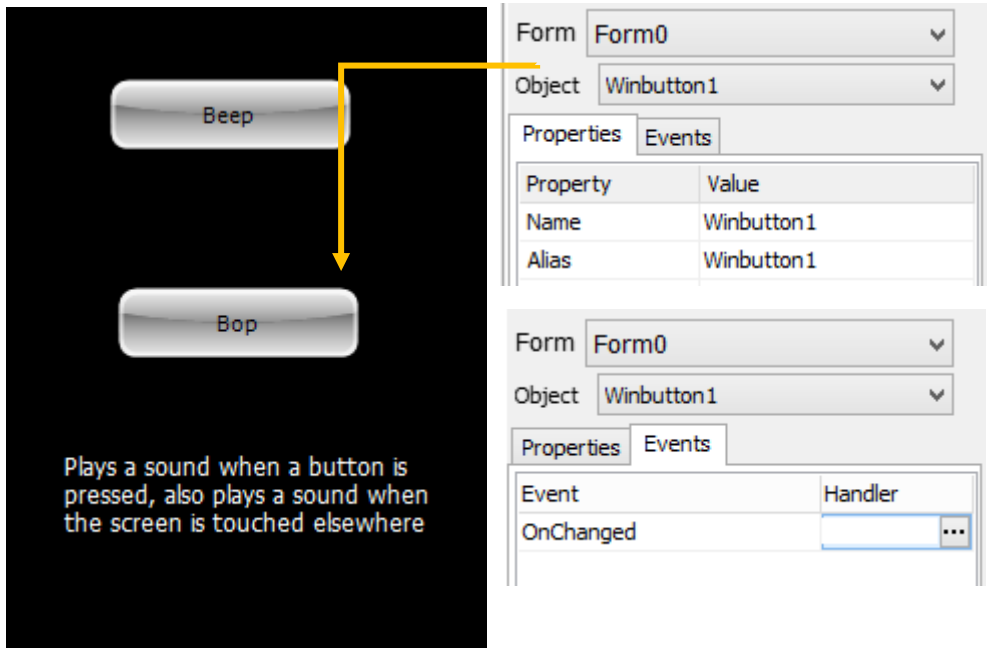
Two fancy button objects are added to Form 0. These are **Winbutton0** and **Winbutton1**.



Winbutton0



Winbutton1

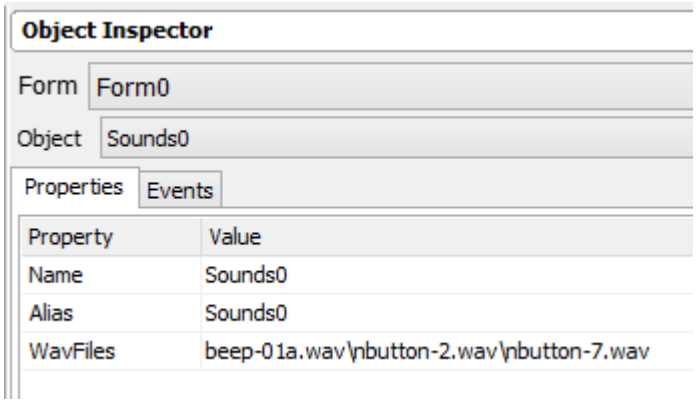


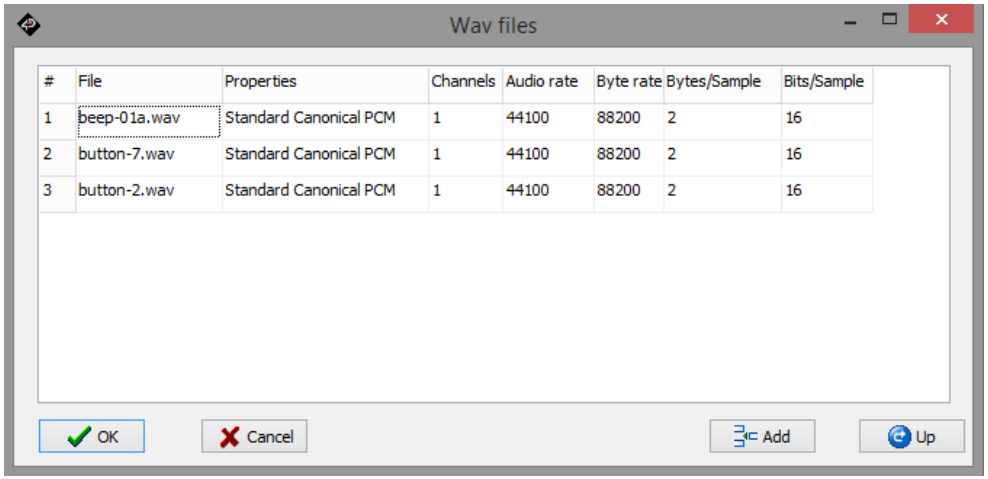
Both Winbutton0 and Winbutton1 have their OnChanged Event set to NONE.

To know more about Winbuttons, their properties, and how they are added to a project, refer to the application note ViSi-Genie Advanced Buttons

Add Sound Object

A sound object is added to Form0. This is **Sounds0**.



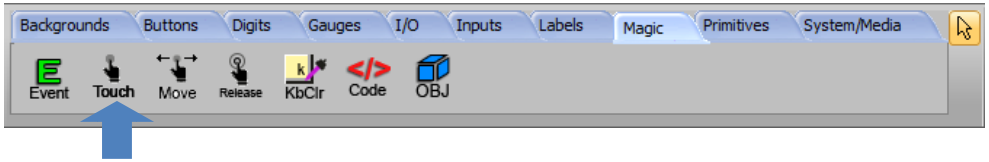


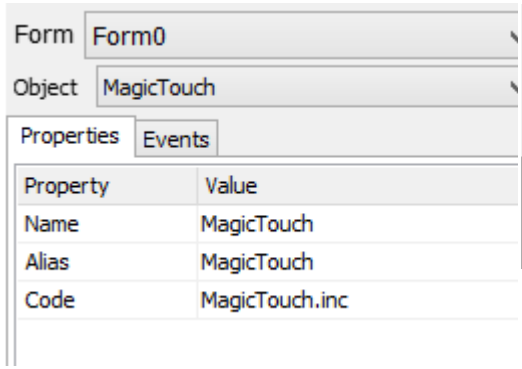
The sound Object contains 3 WAV files. As shown in the image above. How the wav files will be played will be explained later on.

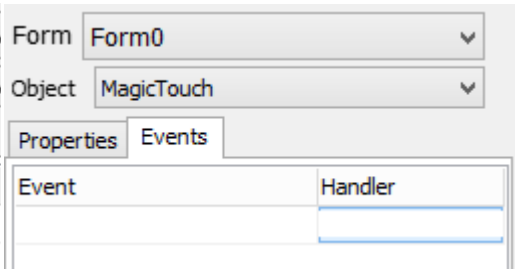
To know more about Sound Object, it's properties, and how it is added to a project, refer to the application note ViSi-Genie Play Sound

Add Magic Touch Object

A Magic Touch Object is added to Form0. This is **MagicTouch**.







A Magic Touch object contains a 4DGL routine that is executed the moment that a “TOUCH_PRESSED” action is detected on the display.

```
8     if (! snd_Playing())
9         if (ImageTouched == iWinbutton0)
10            file_PlayWAV(iSounds[0]) ;
11        else if (ImageTouched == iWinbutton1)
12            file_PlayWAV(iSounds[1]) ;
13        else
14            file_PlayWAV(iSounds[2]) ;
15        endif
16    endif
```

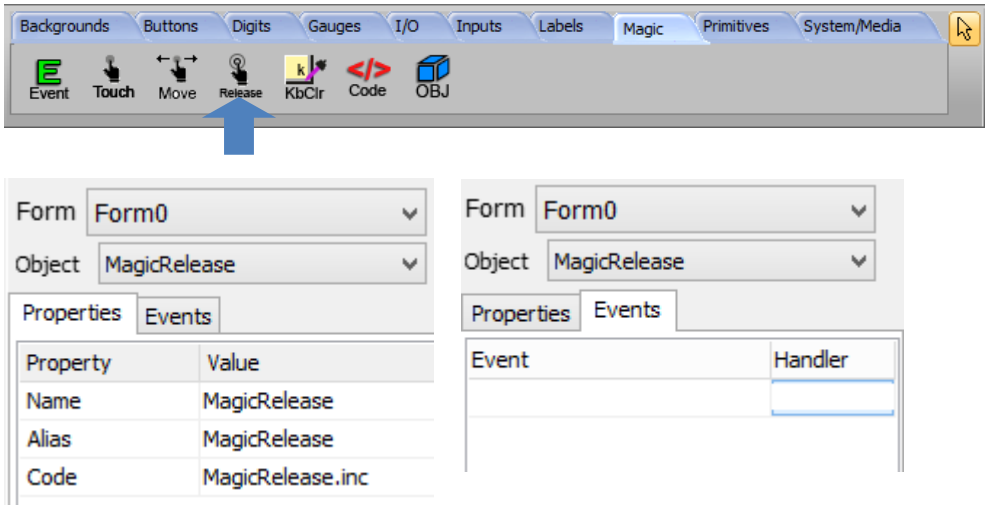
The code shown above is the content of the MagicTouch.inc. The function 'snd_Playing()' is used to check if the sound object is playing or not. It returns 0 if sound has finished playing, else returns number of 512 byte blocks to go. If sound is not playing then proceed to touch detection.

The variable 'ImageTouched' is the object currently being touched. It is compared to values such as iWinbutton0 to determine the object for which a touch has just been detected. If Winbutton0 is touched then the sound object will play the first wav file 'beep-01a.wav'. If Winbutton1 is touched then the sound object will play the second wav file 'button-7.wav'. Else if neither Winbutton0 nor Winbutton1 is pressed but the display is still touched then the program will play 'button-2.wav'.

The 'file_PlayWav(fname)' Open the wav file, decode the header to set the appropriate wave player parameters and set off the playing of the file as a background process. See “Sound Control Functions” in PICASO or DIABLO16 Internal functions for additional play control functions.

Add Magic Release

A Magic Release Object is added to Form0. This is **MagicRelease**.



A Magic Release object contains a 4DGL routine that is executed the moment that a “TOUCH_RELEASED” action is detected on the display.

```
8     snd_Stop() ;
```

The content of MagicRelease.inc is used to determine the object for which touch has just been released. If so then the sound playing will be stopped using 'snd_Stop()' function.

To know more about adding Magic Objects refer to the application note ViSi-Genie How to Add Magic Objects

Build and Upload the Project

For instructions on how to build and upload a ViSi-Genie project to the target display, please refer to the section "**Build and Upload the Project**" of the application note

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.
4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems.
All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expresses or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.
Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.
In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.
4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage (‘High Risk Activities’). 4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.
Use of 4D Systems’ products and devices in 'High Risk Activities' and in any other application is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.