

HyperStudio[®] Clip-Sounds Volume 1

by Dave Klimas

Contains 41 sound files, including:

- Musical Transitions
- Button Noises
- Sound Effects

Also Includes Two New XCMDs:

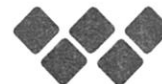
ButtonPlay

Allows HyperStudio 2.1 owners to play a sound attached to another button!

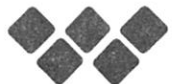
and

DiskPlay

Plays a sound directly from disk, allowing sounds of virtually unlimited length!



copyright © 1990-91, Roger Wagner Publishing, Inc.
1050 Pioneer Way, Suite P, El Cajon, California 92020
(619) 442-0522 voice • (619) 442-0525 facsimile



HyperStudio Clip-Sounds, Volume 1
by Dave Klimas

Copyright © 1990–91 Roger Wagner Publishing, Inc.
All Rights Reserved.

The software and documentation in this package are copyrighted products of Roger Wagner Publishing, Inc., and may not be duplicated other than as provided by the following License.

Sound Use License:

This product is licensed to the purchaser (Licensee) under the following conditions:

1. Individual sounds may be used by the Licensee for private or commercial use, provided the sounds are not resold for their own intrinsic value.
2. The sounds may not be sold or otherwise distributed by itself.
3. Roger Wagner Publishing, Inc. may revoke this license at its sole discretion, upon written notice to the Licensee and refund of the purchase price. Under such conditions, Licensee shall destroy all copies of licensed sounds in its possession, and cease further distribution of the sounds or any product containing the sounds.

Hey, what we're trying to say here is that you can use the sounds in any legitimate personal or commercial project, as long as you're not just trying to resell our sounds as your own, in the "Acme Clip-Sounds Disk," for example, or giving copies of our disk away to your friends or favorite modem on-line service.

Thanks!

HyperStudio Clip-Sounds Volume 1 by Dave Klimas

Use of the Xcmds

This package contains not only two new disks of Clip-Sounds for your HyperStudio stacks, but two new Xcmds as well.

ButtonPlay is an Xcmd that will let one button play an embedded sound that is attached to another button. For example, if you've added one button in your stack called "Try Again" that says those words when clicked, you can create as many other buttons in your stack as you wish that will use the sound from the first button.

DiskPlay is an Xcmd that will let you play sounds directly from disk, no matter how large the files are. This is because DiskPlay only uses 16K of computer memory to play a sound directly from disk!

The Xcmds in this package are provided for use in your own stacks. If you wish to distribute these Xcmds with stacks that you'll sell, an additional Commercial License Agreement is required, and is available at a nominal one-time additional charge.

Public domain and non-profit distribution is not currently provided for, although this may change in the future.

ButtonPlay

This Xcmd routine will play embedded HyperStudio digitized sounds that exist within the current stack, so the sound will not have to be embedded in the stack more than once.

Note that, if you're using HyperStudio 3.0, you don't need to use ButtonPlay because it is already built into the software. (And, addi-

tionally, ButtonPlay is incompatible with HS 3.0.) To play a sound already attached to another button with HyperStudio 3.0, simply select the sound from the list in the upper-right corner of the HyperStudio Tape deck.

To use this routine with HyperStudio 2.1, place the file `HS.Xcmd` from the `Button.Play` directory of the `/HS.Snds.Vol1b` disk in the same directory as your stack.

The second step is to add a new button that will play the sound of another button that is already in your stack. Add the button as you normally would, and when the Button Actions menu appears, check "Trigger Xcmd." When the dialog box asking for the text to pass to the Xcmd appears, enter the name of the other button. That's it! The new button will now play whatever sound is attached to the first button.

If you re-create the first button with a new sound (be sure to give the replacement button the same name!), all the ButtonPlay Xcmd buttons that use the first button will now play the new sound!

You can also add other options to the command line for the Xcmd:

```
[+Td] [+Ln] [+Vn] button.name
```

The `[` and `]` symbols represent the optional values you can include in the Button.Play command. They allow you to add controls for the volume of the sound, and the ability to loop the sound with a time delay between each loop:

```
+Vn    volume, where n=0...15  
       (default=0, use file value)
```

```
+Tn    time delay where n=0...255  
       (1 = 0.1 seconds; default=0 none)
```

```
+Ln    loop back until escape pressed; n=0...63  
       (default= no loop, 0= infinite loop)
```

The button name is the name assigned to the button that has the sound attached to it. If 'button.name' is not given (i.e., the name is blank), then all sounds in the stack will be played.

DiskPlay

This Xcmd routine will play HyperStudio digitized sound files found in the same directory as the current stack.

The sounds will be played in 16K blocks directly from disk.

To use DiskPlay, you must put the file `HS.Xcmd` (from the main directory of the `/HS.Snds.Vol1` disk) in the same directory as your stack. (If you're using HyperStudio 3.0, rename the file to "XCMD." — that is, the word "XCMD" followed by a period.)

When you define a new button and see the Button Actions menu, choose "Trigger Xcmd," and enter the name of the file you wish to play. (If you're using HyperStudio 3.0, select "New Button Actions" from the Button Actions screen, then "MiniScript", then click "Use NBA..." and then enter the examples below in the text item above the "Use NBA..." button.)

For example, entering:

```
BEAM.DOWN
```

would play the sound file BEAM.DOWN if that sound file were located in the same directory as the stack. Entering:

```
/HARD1/SNDS/BELL
```

would go directly to the specified pathname for the sound.

NOTE: DiskPlay does not do any error-checking, and will not display a dialog box the way HyperStudio itself does if the given sound file cannot be found. It is a good idea, therefore, to make sure your stack properly prompts the user to make sure the required disk is in the drive.

You can also add optional characters to control the volume, and to set up a looping sound:

```
[+Tn] [+Ln] [+Vn] [!][pathname]
```

Where:

+Tn time delay where n=0...63 seconds
 (default=0, none)

+Ln loop back until *escape* pressed
 n=0...63
 (default=no loop, 0=infinite loop)

+Vn volume, where n=0...15
 (default=0, use file value)

pathname sound file to play. If a directory name is given, all sounds in that directory will be played. If no pathname is given, all the sounds found in the stack directory will be played. Press the *escape* key to terminate a directory playback. Click to the mouse to advance to the next sound at any time.

Note: If the pathname is the '!' character, then sounds in the most-recently accessed directory will be played.

MASTER Xcmd Notes:

If you have the HyperStudio Xcmd Library Disk, Volume 1, then you can use the MASTER Xcmd to use multiple Xcmds in a stack, and also call them by name.

To use ButtonPlay and DiskPlay, rename the file HS.Xcmd in the main directory of the /HS.Snds.Vol1 disk to XCMD.DiskPlay. Rename the file HS.Xcmd in the Button.Play folder to XCMD.ButtonPlay. Then, use these files with the other Xcmds from the Xcmd Vol. 1 library disk.

For example, you can use the command:

```
ButtonPlay +V10 Lion
```

to play a sound attached to the button named "Lion" at a volume of 10.

ButtonPlay will accept input from the 'It' variable, including any of the option switches. The syntax is exactly the same as noted above. The 'It' variable switch inputs will over-ride any defined in the button.

Sound Library

This Clip-Sounds package includes the following sound files:

Musical Selections

Country Dance
Harp Intro
King's Court
Morning
Rock Intro
Guitar
Banjo
Dulcimer
Lute

Rubber Band
Spaceship
Harmonica
Bounce 1
Bounce 2

Button Noises

Boing1
Boing2
Klank
Click

Click Stuff

Clock Tower Bell
Grandfather Chime
Cuckoo Clock
Ticking

Sound Effects

Fire Alarm
Siren
Buzzer1
Buzzer2

Animals

Cat
Chicks
Horse

Knock at Door
Doorbell
Door Opening
Door Closing

Strange Sounds

Alien Radio
Slow Whine
Power Source

Phone Dialing
Phone Ringing
Phone Busy
Phone Pickup
Phone Hanging Up