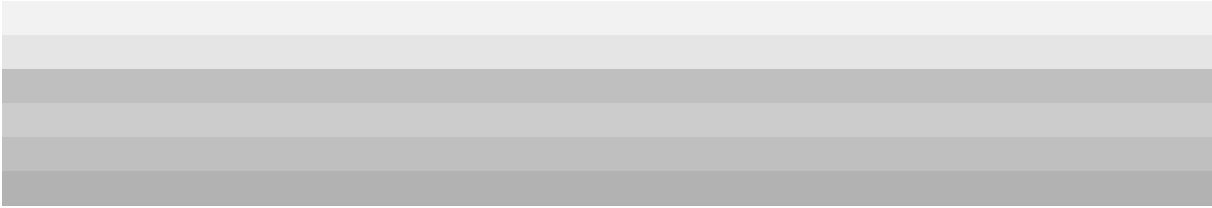




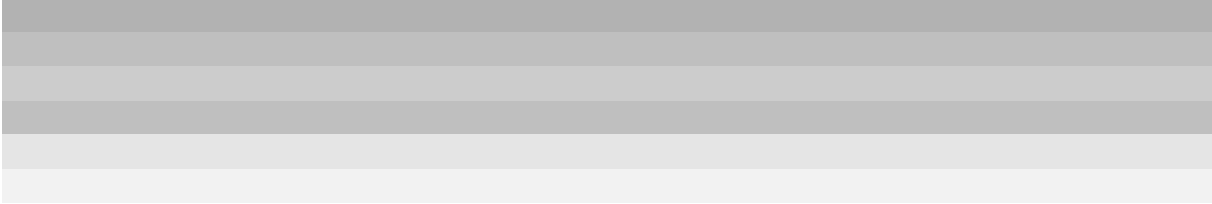
**B**UILD **D**OCUMENTATION



. 3D . REALMS . ENTERTAINMENT . INC .  
. 1997 . . . . .  
. . . . .  
. . . . .



Shadow Warrior Reference Guide



Copyright & Legal Stuff

This document is Copyright 1997 by Steffen Itterheim and Keith Schuler on behalf of 3D Realms Entertainment. This documentation may not be sold or distributed other than on the Shadow Warrior Registered Version CD. Printing, publishing, distributing or otherwise using excerpts of this document in any

media (books, CDs) is prohibited. It may only be printed for personal use. If you wish to use this document or excerpts from it for commercial use please contact the authors.

## **Table of Contents**

<b>1)</b>	<b>Introduction</b>	<b>3</b>
<b>2)</b>	<b>Build Key Command Reference</b>	<b>3</b>
	<b>2.I) Reading the Key Command Reference</b>	4
	<b>2.II) 2D View Key Commands</b>	4
	<b>2.III) 3D View Key Commands</b>	6
	<b>2.IV) 2D and 3D View Key Commands</b>	8
	<b>2.V) Texture View Key Commands</b>	8
<b>3)</b>	<b>Palette Color Reference</b>	<b>9</b>
<b>4)</b>	<b>Digital &amp; Ambient Sound Reference</b>	<b>10</b>
	<b>4.I) Digital Sound Reference</b>	10
	<b>4.II) Ambient Sound Reference</b>	21
<b>5)</b>	<b>Sector Tag Reference</b>	<b>22</b>
	<b>5.I) Sector Triggers</b>	22
	<b>5.II) Sine Wave on Floor and/or Ceiling</b>	24
	<b>5.III) Sector Objects Sector Tags</b>	25
<b>6)</b>	<b>Wall Tag Reference</b>	<b>26</b>
	<b>6.I) Sine Wave Wall Effect</b>	26
	<b>6.II) Misc Wall Tags</b>	27
	<b>6.III) Sector Object Wall Tags</b>	28
<b>7)</b>	<b>Sector Effector (ST1) Sprite Reference</b>	<b>29</b>
	<b>7.I) Meet the ST1 Sprite</b>	31
	<b>7.II) The Shadow Warrior Tag System</b>	31
	<b>7.III) Notes Regarding the ST1 Sprite</b>	33
	<b>7.IV) The Shadow Warrior ST1 Function Reference</b>	34
<b>8)</b>	<b>Track (T#) Sprite Reference</b>	<b>69</b>
	<b>8.I) How to use the Track Sprite</b>	70
	<b>8.II) Track Tag Function Reference</b>	71
<b>9)</b>	<b>Misc Sprites Reference</b>	<b>76</b>
	<b>9.I) Actors</b>	77
	<b>9.II) Cracks</b>	79
	<b>9.III) Switches</b>	79
	<b>9.IV) Keys and Keyholes</b>	81
	<b>9.V) TV Monitors</b>	82
	<b>9.VI) QuickJump and QuickJumpDown</b>	82
	<b>9.VII) Gratings</b>	82
	<b>9.VIII) DeathFlag(tm) Capture the Flag</b>	83
<b>10)</b>	<b>EXAMPLE.MAP Reference</b>	<b>84</b>
	<b>10.I) Explanation</b>	85
	<b>10.II) Column A</b>	85
	<b>10.III) Column B</b>	86
	<b>10.IV) Column C</b>	87
	<b>10.V) Column D</b>	87
	<b>10.VI) Column E</b>	88
	<b>10.VII) Column F</b>	90

## 1) Introduction

This Reference Guide was written by Steffen „Duke Addict“ Itterheim and Keith Schuler. Many thanks to Keith Schuler for providing me with the Sector, Wall and Sprite Tag references as well as a complete list of sound effects used in Shadow Warrior. I have slightly revised his texts, mainly to make the original ASCII text files look better in a Word document and to make them look the same way as the rest of the document. The Shadow Warrior Reference Guide contains complete lists of key commands available in the Build editor, Palette and Sound numbers as well as all the tag numbers used in the game to create all the special effects available in Shadow Warrior. Because of this I recommend that you print this document whether you're an advanced Build user or not and have it at hand while designing your maps. This will help you to quickly look up how to set up a certain special effect and helps you memorizing the key commands.

## 2) Build Key Command Reference

by Steffen Itterheim with help from Keith Schuler

Here's a summary of all the keys that are available in Shadow Warrior's Build editor. Most of these keys also work in other Build games, like Duke Nukem 3D. Note that for most keys to work the proper object or surface must be pointed at, or even selected before you can use the key. Also many keys only work in either 2D or 3D mode.

In case you're using a non-US keyboard you should know that the key layout is different. For example german keyboards have the Z and Y keys swapped compared to US keyboards. That means that you have to press Y to actually activate the function on the Z key (which is zoom out). On the other hand if you would press the Z key on the german keyboard Build will recognize it as the Y key and do nothing. To find out where the US keys are located on your keyboard either set your keyboard to use US keyboard layout and write down where the keys are as you press them, or use a keyboard layout chart that you might find in some documentation, for example the DOS User Manual. It can be a hassle finding keys like [ or / on a non-US standard keyboard but there's no working around this other than memorizing where these keys are located on your keyboard.

### ***Build Key Command Reference Contents***

2.I) Reading the Key Command Reference	4
2.II) 2D View Key Commands	4
2.III) 3D View Key Commands	6
2.IV) 2D and 3D View Key Commands	8
2.V) Texture View Key Commands	8

## 2.I) Reading the Key Command Reference

**S+Alt:** means the keys have to be pressed together at the same time, press both the Alt key and the S key at the same time.

**, or .:** means both keys perform basically the same function, in this case one is for clockwise rotation while the other is used for counter clockwise rotation.

**2, 4, 6 or 8 + Shift:** means press either one of the keys to the left of the + together with the key on the right, in this example press either 2, 4, 6 or 8 together with the Shift key.

**V; V:** means the keys have to be pressed in order, hit V and then hit V a second time

**Del+Ctrl (right):** means you have to press the right Ctrl key and not the left Ctrl key.

**(numeric keypad):** braces provide additional information about the keys

## 2.II) 2D View Key Commands

Key Command	What it does	Works in
, or . (comma or period)	Moves viewpoint left or right (strafing)	2D View
, or . (comma or period) while sector(s) is/are selected	Rotates selected sector(s) by 90 degrees	2D View
,+Shift or .+Shift (comma or period + Shift) while sector(s) or sprite(s) is/are selected	Rotates selected sector(s) or sprite(s) by 1 degree each time	2D View
;+1,2,3,4,5,6,7,8,9 or 0 (semicolon + ...)	Edits respective boolean tag (1-10) of currently selected sprite	2D View
;+1+ Shift (semicolon + ...)	Edits respective boolean tag (11) of currently selected sprite	2D View
‘+1,2,3,4,5 or 0 + Shift (apostrophe + ...)	Edits respective tag value (11-15) of currently selected sprite, wall or sector	2D View
‘+1,2,3,4,5,6,7,8,9 or 0 (apostrophe + ...)	Edits respective tag value (1-10) of currently selected sprite, wall or sector	2D View
‘+K (apostrophe + K)	Toggles skill level (0-4) of selected sprite(s)	2D View
‘+M (apostrophe + M)	Displays menu to adjust global settings for parallaxed floors/ceilings	2D View
‘+V (apostrophe + V)	Toggles through 5 levels of verbosity information shown on sprites	2D View
+ or - (plus or minus - numeric keypad)	Increases or decreases number of vertices when in circle wall drawing mode	2D View
A	Zoom in	2D View
Alt (Right Alt)	Selects one or more sectors together with all sprites in them. Used for mass editing sector attributes and to copy sectors.	2D View
Alt (Right Alt); Left Mouse Button held down; Ins (Insert); Esc; L; Alt (Right Alt)	Copies selected sector(s) from the current level to another	2D View
Backspace	Delete last placed vertex, quits sector drawing mode if used on first drawn vertex	2D View
C	Edits an existing wall and adds vertices in order to create a circular wall	2D View
Del+Ctrl (Delete+Right Ctrl)	Deletes a sector	2D View
Drag vertex onto neighbor vertex	Deletes vertex	2D View
Drawing a wall between two vertices of a sector	Splits sector	2D View
F5	Displays sprite information and next free tag number	2D View
F6	Displays information about current object, if a	2D View

	ST1 sprite is highlighted when pressing F8 context sensitive help will be displayed	
F7	Browse F8 search results	2D View
F8	Shows a menu for sprites to toggle skill level, multiplayer and to search for sprites	2D View
G	Changes grid size	2D View
H	Edits current sector's hitag	2D View
H+Alt	Edits hitag of either the currently selected wall or sprite	2D View
H+Ctrl	Toggles 2-sided wall's or sprite's hitscan bit	2D View
Ins (Insert)	Splits a wall and inserts a vertex	2D View
Ins (Insert) when sector(s) or sprite(s) is/are selected	Copies sector(s) or sprite(s); for sectors hold down left mouse button before copying	2D View
J; J (pressed once on each sector)	Joins two sectors into one, deleting the second	2D View
L	Toggles grid locking on and off	2D View
Left Mouse Button (dragging)	Move a vertex	2D View
P	Prompts for palette values for current sector's floor and ceiling	2D View
Return	Checks highlighted wall's line pointers, checks for misreferenced walls	2D View
Return+Shift+Ctrl	Checks all walls and sectors for misreferences, fixes found misreferences - useful for large maps	2D View
Right Mouse Button	Scrolls map around, repositions white arrow	2D View
S+Alt (while cursor points in carved out space)	Creates a child sector	2D View
Scroll Lock	Changes player start position	2D View
Shift+Ctrl+moving the mouse (Right Shift)	Selects all vertices in all sectors where at least one vertex is inside the selection rectangle	2D View
Shift+moving the mouse (Right Shift)	Selects multiple vertices	2D View
Spacebar	Starts sector drawing mode, or if already in drawing mode Spacebar places a vertex. Places vertices drawn with circle draw mode.	2D View
T	Edits current sector's lotag	2D View
T+Alt	Edits lotag of either the currently selected wall or sprite	2D View
T+Ctrl	Toggles sprite descriptions on and off	2D View
Tab (Tabulator)	Displays information about the current sector	2D View
Tab+Alt (Tabulator + Alt)	Displays information about either the currently selected sprite or wall; NOTE: windows users may want to disable the Alt+Tab windows key while using Build via shortcut properties (last page).	2D View

## 2.III) 3D View Key Commands

Key Command	What it does	Works in
. (period)	Aligns wall textures automatically, going from the current wall to the right	3D View
/ (slash)	Resets pan, scale, slope, etc. values of the selected object to default values	3D View
[ or ]	Slopes floor or ceiling of a sector	3D View
[+Alt or ]+Alt	Aligns slope to adjacent sector	3D View
[+Shift or ]+Shift	Slopes floor or ceiling of a sector in smaller increments	3D View
‘+A (apostrophe + A)	Toggles sprite autosizing, will make weapons, etc. appear in Build the same size as in the game	3D View
‘+C (apostrophe + C)	Pastes texture in clipboard on all surfaces in the selected sector(s) only that match the selected surface’s texture and are of the same type (wall, floor/ceiling or sprite).	3D View
‘+H (apostrophe + H)	Edits hitag value of sector, wall or sprite	3D View
‘+M (apostrophe + M)	Toggles sprite display: all sprites shown, only monsters and ST1 sprites or no sprites	3D View
‘+P (apostrophe + P)	Pastes texture in clipboard on all surfaces in the selected sectors that match the selected surface’s texture and are of the same type (wall, floor/ceiling or sprite). If no sectors are selected it works the same way as C+Alt	3D View
‘+Return (apostrophe + Return)	Copies texture from the clipboard but leaves surface’s current shading value as it is	3D View
‘+S (apostrophe + S)	Allows to directly input a shade value for the selected texture	3D View
‘+T (apostrophe + T)	Edits lotag value of sector, wall or sprite	3D View
‘+V (apostrophe + V)	Adjusts visibility of a sector or all selected sectors	3D View
‘+X (apostrophe + X)	Toggles between bitmap and voxel sprites	3D View
‘+Z (apostrophe + Z)	Toggles voxel rotation on and off	3D View
+ or - (plus or minus - numeric keypad)	Shades textures; can shade all selected sectors	3D View
+ plus Alt or - plus Alt (plus or minus + Alt) (numeric keypad)	Adjusts visibility of a sector or all selected sectors	3D View
+ plus Alt plus Ctrl or - plus Alt plus Ctrl (plus or minus + Alt + Ctrl) (numeric keypad)	Adjusts visibility of all sectors in the level	3D View
1	Turns a 2-sided wall into a 1-sided wall which can be walked through	3D View
2	Toggles whether upper and lower textures can be edited independently	3D View
2, 4, 6 or 8 (numeric keypad)	Scales Wall texture; pans floor/ceiling texture	3D View
2, 4, 6 or 8 + Shift (numeric keypad)	Pans Wall texture	3D View
3	Toggles room over room draw on and off. Use this rarely as certain functions will be disabled as long as it is on. Toggle it off before you switch to 2D mode.	3D View
4	Moves view position through room over room mirror divider when room over room drawing is turned on	3D View

5 (numeric keypad)	speeds up texture scaling and panning	3D View
A	Raise viewpoint	3D View
A+Ctrl	Look down	3D View
C+Alt	Pastes texture in clipboard on all surfaces in the level that match the selected surface's texture and are of the same type (wall/floor/ceil./sprite)	3D View
Caps Lock	Changes height movement mode	3D View
Ctrl+Left or Right Arrow (Right Ctrl + ...)	Moves viewpoint sideways	3D View
D+Alt	Adjusts a sprite's clipdist	3D View
E	Toggles floor/ceiling texture scale value	3D View
F	Rotates texture by 90 degrees	3D View
F11	Gamma Correction	3D View
H	Toggles 2-sided wall's/sprite's hitscan bit	3D View
Left Mouse Button held down	Locks the currently selected object	3D View
M	Creates a masked wall	3D View
M+Shift	Creates a masked wall only on one side	3D View
O (if used on a wall texture)	Change drawing orientation of a wall texture	3D View
P	Parallaxes floor/ceiling	3D View
P+Alt	Changes texture palette	3D View
P+Ctrl	Changes parallax mode of selected floor/ceiling	3D View
PgDn (Page Down)	Lowers floor/ceiling or a sprite	3D View
PgDn+Ctrl (Page Down + Ctrl)	Lowers a sprite onto the floor	3D View
PgUp (Page Up)	Raises floor/ceiling or a sprite	3D View
PgUp+Ctrl (Page Up + Ctrl)	Raises a sprite onto the ceiling	3D View
R (if used on a floor or ceiling)	Align floor/ceiling texture along sector's first wall (relative alignment)	3D View
R (if used on a sprite)	Toggles a sprite's display mode: normal (rotating), flat on wall or flat on floor/ceiling	3D View
Return	Pastes a texture from clipboard	3D View
Return+Ctrl	Pastes a texture from clipboard to all walls in a loop	3D View
Return+Shift	Pastes only the shade value of the texture in the clipboard	3D View
Return+Shift+Ctrl	Shades all walls in sector automatically according to their angle. Brightest wall in sector will be the wall pointed at, walls facing the opposite direction will be shaded darkest	3D View
T	Makes masked wall or sprite translucent	3D View
Tab (Tabulator)	Copies a wall or sprite texture into clipboard	3D View
V;Return	Displays list of currently used textures and applies the selection to a wall, floor/ceil./sprite	3D View
V;V;Return	Displays list of all textures and applies the selection to a wall, floor/ceiling or sprite	3D View
Z+Ctrl	Look up	3D View

## 2.IV) 2D and 3D View Key Commands

Key Command	What it does	Works in
, or . (comma or period)	Rotates selected sprite(s) by 22.5 degrees	2D and 3D
,+Shift or .+Shift (comma or period + Shift)	Rotates selected sprites by 1 degree each time	2D and 3D
Arrow keys	Moves viewpoint (white arrow)	2D and 3D
B	Makes a 2-sided wall or sprite blocking	2D and 3D
B+Shift	Makes a 2-sided wall or sprite blocking on only one side	2D and 3D
Del	Deletes a sprite	2D and 3D
Enter (numeric keypad)	Toggles between 2D and 3D mode	2D and 3D
Esc	Displays menu to save, quit etc.	2D and 3D
F+Alt	Changes sector's first wall	2D and 3D
F1	Moves viewpoint 1 unit north each time	2D and 3D
F10	Rotates viewpoint angle clockwise by 1 degree	2D and 3D
F12	Dumps a screenshot to CAPTxxxx.PCX	2D and 3D
F2	Moves viewpoint 1 unit south each time	2D and 3D
F3	Moves viewpoint 1 unit west each time	2D and 3D
F4	Moves viewpoint 1 unit east each time	2D and 3D
F9	Rotates viewpoint angle counter clockwise by 1 degree	2D and 3D
O (if used on a sprite)	Places a sprite onto a wall, making it flat	2D and 3D
S	Inserts a sprite, pastes the current texture in clipboard as sprite	2D and 3D
Shift	Speeds up movements when held down	2D and 3D
Z	Zoom out (2D); Lower viewpoint (3D)	2D and 3D

## 2.V) Texture View Key Commands

Key Command	What it does	Works in
Arrow keys, PgDn or PgUp	Moves texture selector	Texture View
G	Prompts for a tile number to go to	Texture View
Return	Applies selected texture to wall	Texture View
V	Displays a list of all textures in the game if not already viewing the full texture list	Texture View



### 3) Palette Color Reference

by Steffen Itterheim and Keith Schuler

Shadow Warrior can display one and the same texture with different color palette's applied to it, making it look blue, red, green and so on. Some palette numbers define attributes of certain objects, for example different palette numbers are used on the Evil Ninja monster to turn it into slightly different versions of the same monster in the game. Although the Ninja's use the same sprite frames they can shoot different weapons or become nearly invisible when given the proper palette numbers.

Keith Schuler, one of 3D Realms' Level Designers, uses a trick to make it easier to spot certain ST1 sprites. He applies ST1 sprites of the same type a certain palette number to make them blue, red, etc. so they're easier to spot out in places where there are many ST1 sprites used. This works without problems because none of the ST1 sprite's functions will be affected by their palette number, with the exception of the lighting types.

Palette No	Palette Color
0	Default, normal color
1	Fog (White) Haze
4	Dark Gray
5	Heat (Red) Haze
6	Red sprinkle
7	Gray swapped to brown/bronze
8	Green sprinkle
9	Water (Blue) Haze
11	Green
13	Blue
14	Red
16	Blue swapped to tan
17	Blue swapped to gray

Palette No	Palette Color
18	Blue swapped to purple
19	Blue swapped to cool red
20	Blue swapped to yellow
21	Blue swapped to olive
22	Blue swapped to green
24	Blue swapped to orange/brown
25	Blue swapped to warm red
26	Blue swapped to dark grey
27	Blue swapped to avocado
28	Blue swapped to deep blue
29	Blue swapped to bright orange
30	Blue swapped to dark orange
32	Night Vision colors

## 4) Digital & Ambient Sound Reference

by Keith Schuler, revised by Steffen Itterheim

This reference lists all the sound effects used in Shadow Warrior. You can use this as a reference for digital sound entry numbers to use in order to generate different sounds in your levels. This reference contains two parts. The first part is the digital sound list, which provides entry numbers to be used with the SOUND\_SPOT (hitag 134) ST1. The second part provides entry numbers to be used with the AMBIENT\_SOUND (hitag 1002) ST1.

### Digital & Ambient Sound Reference Contents

4.I) Digital Sound Reference **10**

4.II) Ambient Sound Reference **21**

#### 4.I) Digital Sound Reference

This section is arranged in four columns. The first column is the actual filename of the sound .VOC as it appears in the .GRP file. The second column lists an identifying label that the programmers used, but it was left in the list to help you identify the sound more easily. The third column is the actual sound entry number. This is the number you use as TAG4 (Angle), TAG5 or TAG6 value of the SOUND\_SPOT ST1 (hitag 134). You can also use this number in the lotag of a wall that is hitagged with TAG\_WALL\_PLAY\_SOUND (walltag 900). Do **NOT** use these numbers with AMBIENT\_SOUND ST1's (hitag 1002)! It won't work properly. Sound numbers for use with AMBIENT\_SOUND are listed in the second part of this reference. The fourth and final column might contain a descriptive comment.

**NOTE:** A lot of these sounds are not used in the game. Some of which I am not sure were used are marked with a „?“ (question mark). If you can't find a .VOC file listed here in the .GRP file then feel free to use your own custom sound by copying it into the game directory using the same filename. You can then use that entry's sound number to hear your custom sound in the game.

Filename	Identity	No.	Comments
NULL.VOC	DIGI_NULL	0	Don't change this
SWRDSTR1.VOC	DIGI_SWORDSWOOSH	1	Sword
THROW.VOC	DIGI_STAR	2	Shuriken sounds
STRCLNK.VOC	DIGI_STARCLINK	3	
STRWIZ.VOC	DIGI_STARWIZ	4	
UZIFIRE1.VOC	DIGI_UZIFIRE	5	Uzi sounds

RICH1.VOC	DIGI_RICHOCHET1	6	
RICH2.VOC	DIGI_RICHOCHET2	7	
RMVCLIP.VOC	DIGI_REMOVECLIP	8	
RPLCLIP.VOC	DIGI_REPLACECLIP	9	
SHELL.VOC	DIGI_SHELL	10	Spent shell
RIOTFIR1.VOC	DIGI_RIOTFIRE	11	Riot shotgun sounds
SHOTGUN.VOC	DIGI_RIOTFIRE2	12	
RIOTRLD.VOC	DIGI_RIOTRELOAD	13	
EXPMED.VOC	DIGI_BOLTEXPLODE	14	
RIOTWIZ.VOC	DIGI_BOLTWIZ	15	
40MMFIR2.VOC	DIGI_30MMFIRE	16	Grenade launcher
RIOTRLD.VOC	DIGI_30MMRELOAD	17	
40MMEXP2.VOC	DIGI_30MMEXPLODE	18	
RIOTWIZ.VOC	DIGI_30MMWIZ	19	
GHFIR1.VOC	DIGI_HEADFIRE	20	Guardian head
GHWIZ.VOC	DIGI_HEADSHOTWIZ	21	
EXPSML.VOC	DIGI_HEADSHOTHIT	22	
THROW.VOC	DIGI_MINETHROW	23	Sticky mines
PHITGRND.VOC	DIGI_MINEBOUNCE	24	
EXPLRG.VOC	DIGI_MINEBLOW	25	
STSCAN2.VOC	DIGI_MINEBEEP	26	
HBLOOP1.VOC	DIGI_HEARTBEAT	27	Heart attack
HSQUEEZ1.VOC	DIGI_HEARTFIRE	28	
HRTWIZ.VOC	DIGI_HEARTWIZ	29	
RIOTFIR1.VOC	DIGI_MISSLFIRE	30	Missile launcher
EXPMED.VOC	DIGI_MISSLEXP	31	
RFWIZ.VOC	DIGI_RFWIZ	32	Ring of Fire
NAPFIRE.VOC	DIGI_NAPFIRE	33	Napalm
NAPTWIZ.VOC	DIGI_NAPWIZ	34	
NAPPUFF.VOC	DIGI_NAPPUFF	35	
MMFIRE.VOC	DIGI_MIRVFIRE	36	Magic mirv spell
MMWIZ.VOC	DIGI_MIRVWIZ	37	
SPRLFIRE.VOC	DIGI_SPIRALFIRE	38	Spiral spell (?)
SPRLWIZ.VOC	DIGI_SPIRALWIZ	39	
MAGIC1.VOC	DIGI_MAGIC1	40	General magic sounds
MAGIC2.VOC	DIGI_MAGIC2	41	
MAGIC3.VOC	DIGI_MAGIC3	42	
MAGIC4.VOC	DIGI_MAGIC4	43	
MAGIC5.VOC	DIGI_MAGIC5	44	
MAGIC6.VOC	DIGI_MAGIC6	45	
MAGIC7.VOC	DIGI_MAGIC7	46	
SWSPELL.VOC	DIGI_SWCLOAKUNCLOAK	47	Cloaking bomb
DHVOMIT.VOC	DIGI_DHVOMIT	48	Vomiting dead head
DHCLUNK.VOC	DIGI_DHCLUNK	49	
DHSQSH.VOC	DIGI_DHSQUISH	50	
LAVAHIT.VOC	DIGI_PROJECTILELAVAHIT	51	Weapon fire hits lava
STSPL01.VOC	DIGI_PROJECTILEWATERHIT	52	or water
KEY.VOC	DIGI_KEY	53	Pick up item sounds
ITEM5A.VOC	DIGI_ITEM	54	
ITEMBIG2.VOC	DIGI_BIGITEM	55	
BODY9.VOC	DIGI_BODYFALL1	56	Pain and death sounds
PHITGRND.VOC	DIGI_HITGROUND	57	
BODY2.VOC	DIGI_BODYSQUISH1	58	
BURN1.VOC	DIGI_BODYBURN	59	
BURNSCRM.VOC	DIGI_BODYBURNSCREAM	60	
BODY3.VOC	DIGI_BODYCRUSHED1	61	
BODY4.VOC	DIGI_BODYHACKED1	62	
BURN2.VOC	DIGI_BODYSINGED	63	
DROWN1.VOC	DIGI_DROWN	64	
SCREAM1.VOC	DIGI_SCREAM1	65	
SCREAM2.VOC	DIGI_SCREAM2	66	

SCREAM3.VOC	DIGI_SCREAM3	67	
HIT1.VOC	DIGI_HIT1	68	
ELECTRC1.VOC	DIGI_ELECTRICUTE1	69	
SWDIE02.VOC	DIGI_REMOVEME	70	
IMPALE1.VOC	DIGI_IMPALED	71	
OOF1.VOC	DIGI_OOF1	72	
BODY1.VOC	DIGI_ACTORBODYFALL1	73	Player sounds that actors
HITGRND.VOC	DIGI_ACTORHITGROUND	74	also use
COLEXP.VOC	DIGI_COOLIEEXPLODE	75	Coolie
COLSCRM.VOC	DIGI_COOLIESCREAM	76	
COLALRT.VOC	DIGI_COOLIEALERT	77	
COLAMB.VOC	DIGI_COOLIEAMBIENT	78	
COLPAIN.VOC	DIGI_COOLIEPAIN	79	
CGMAT.VOC	DIGI_CGMATERIALIZE	80	Coolie Ghost
CGALRT.VOC	DIGI_CGALERT	81	
CGWHACK.VOC	DIGI_CGTHIGHBONE	82	
CGAMB.VOC	DIGI_CGAMBIENT	83	
CGPAIN.VOC	DIGI_CGPAIN	84	
CGSHOOT.VOC	DIGI_CGMAGIC	85	
CGHIT.VOC	DIGI_CGMAGICHIT	86	
CGSCRM.VOC	DIGI_CGSCREAM	87	
NINAMB.VOC	DIGI_NINJAAMBIENT	88	Evil Ninja
NINSTAR.VOC	DIGI_NINJASTAR	89	
NINPAIN.VOC	DIGI_NINJAPAIN	90	
NINSCRM.VOC	DIGI_NINJASCREAM	91	
NINALRT.VOC	DIGI_NINJAALERT	92	
NINSHOOT.VOC	DIGI_NINJAUZIATTACK	93	
RIOTFIR1.VOC	DIGI_NINJARIOTATTACK	94	
RIPAMB.VOC	DIGI_RIPPERAMBIENT	95	Ripper
RIPALRT.VOC	DIGI_RIPPERALERT	96	
RIPATCK.VOC	DIGI_RIPPERATTACK	97	
RIPPAIN.VOC	DIGI_RIPPERPAIN	98	
RIPSCRM.VOC	DIGI_RIPPERSCREAM	99	
RIPHRT.VOC	DIGI_RIPPERHEARTOUT	100	
GRDAMB.VOC	DIGI_GRDAMBIENT	101	Guardian
GRDALRT.VOC	DIGI_GRDALERT	102	
GRDPAIN.VOC	DIGI_GRDPAIN	103	
GRDSCRM.VOC	DIGI_GRDSCREAM	104	
GRDFIR.VOC	DIGI_GRDFIREBALL	105	
GRDAXE.VOC	DIGI_GRDSWINGAXE	106	
GRDAXHT.VOC	DIGI_GRDAXEHIT	107	
SPAMB.VOC	DIGI_SPAMBIENT	108	Skeletor Priest (?)
SPALRT.VOC	DIGI_SPALERT	109	
SPPAIN.VOC	DIGI_SPPAIN	110	
SPSCRM.VOC	DIGI_SPSCREAM	111	
SPBLADE.VOC	DIGI_SPBLADE	112	
SPELEC.VOC	DIGI_SPELEC	113	
SPTLPRT.VOC	DIGI_SPTLEPORT	114	
AHAMB.VOC	DIGI_AHAMBIENT	115	Accursed Head
AHSCRM.VOC	DIGI_AHSCREAM	116	
AHEXP.VOC	DIGI_AHEXPLODE	117	
AHSHWSH.VOC	DIGI_AHSWOOSH	118	
HBUZZ.VOC	DIGI_HORNETBUZZ	119	Hornet
HSTING.VOC	DIGI_HORNETSTING	120	
HPAIN.VOC	DIGI_HORNETPAIN	121	
HDEATH.VOC	DIGI_HORNETDEATH	122	
SGAMB.VOC	DIGI_SERPAMBIENT	123	Serpent Boss
SGALRT.VOC	DIGI_SERPALERT	124	
SGPAIN.VOC	DIGI_SERPPAIN	125	
SGSCRM.VOC	DIGI_SERPSCREAM	126	
SGDETH.VOC	DIGI_SERPDEATHEXPLODE	127	

SGSWORD.VOC	DIGI_SERPSWORDATTACK	128	
SGMAGIC.VOC	DIGI_SERPMAGICLAUNCH	129	
SGHEADS.VOC	DIGI_SERPSUMMONHEADS	130	
SGTAUNT.VOC	DIGI_SERPTAUNTYOU	131	
LVAMB.VOC	DIGI_LAVABOSSAMBIENT	132	Lava Boss (?)
LVSWIM.VOC	DIGI_LAVABOSSSWIM	133	
LVRISE.VOC	DIGI_LAVABOSSRISE	134	
LVALRT.VOC	DIGI_LAVABOSSALERT	135	
LVFLAME.VOC	DIGI_LAVABOSSFLAME	136	
LVMETEOR.VOC	DIGI_LAVABOSSMETEOR	137	
LVMETEXP.VOC	DIGI_LAVABOSSMETEXP	138	
LVPAIN.VOC	DIGI_LAVABOSSPAIN	139	
LVSIZZLE.VOC	DIGI_LAVABOSSSIZZLE	140	
LVEXPL.VOC	DIGI_LAVABOSSEXPLODE	141	
BTSTRT.VOC	DIGI_BOATSTART	142	Motor Boat
BTRUN01.VOC	DIGI_BOATRUN	143	
BTSTOP.VOC	DIGI_BOATSTOP	144	
BTFIRE.VOC	DIGI_BOATFIRE	145	
TNKSTRT.VOC	DIGI_TANKSTART	146	Army Tank
TNKRUN.VOC	DIGI_TANKRUN	147	
TNKSTOP.VOC	DIGI_TANKSTOP	148	
TNKIDLE.VOC	DIGI_TANKIDLE	149	
TNKFIRE.VOC	DIGI_TANKFIRE	150	
TRUKRUN.VOC	DIGI_TRUKRUN	151	Army Truck (?)
TRUKIDLE.VOC	DIGI_TRUKIDLE	152	
SUBRUN.VOC	DIGI_SUBRUN	153	Submarine (?)
SUBIDLE.VOC	DIGI_SUBIDLE	154	
SUBDOOR.VOC	DIGI_SUBDOOR	155	
BMBFLY.VOC	DIGI_BOMBRFLYING	156	Bomber Plane (?)
BMBDROP.VOC	DIGI_BOMBRDROPBOMB	157	
BUBBLE.VOC	DIGI_BUBBLES	158	Bubbles
CHAIN.VOC	DIGI_CHAIN	159	(?)
CHNDOOR.VOC	DIGI_CHAINDOOR	160	Chain door
CRCKT2.VOC	DIGI_CRICKETS	161	Crickets
DRWOODO.VOC	DIGI_WOODDOOROPEN	162	Door sounds
DRMETO.VOC	DIGI_METALDOOROPEN	164	
DRMETC.VOC	DIGI_METALDOORCLOSE	165	
DRSLDO.VOC	DIGI_SLIDEDOOROPEN	166	
DRSLDC.VOC	DIGI_SLIDEDOORCLOSE	167	
DRSTNO.VOC	DIGI_STONEDOOROPEN	168	
DRSTNC.VOC	DIGI_STONEDOORCLOSE	169	
DRSQKO.VOC	DIGI_SQUEAKYDOOROPEN	170	
DRSQKC.VOC	DIGI_SQUEAKYDOORCLOSE	171	
DRILL.VOC	DIGI_DRILL	172	Giant drill
CAVE1.VOC	DIGI_CAVEDRIP1	173	Water dripping in cave
CAVE2.VOC	DIGI_CAVEDRIP2	174	
DRIP.VOC	DIGI_DRIP	175	
WTRFAL1.VOC	DIGI_WATERFALL1	176	Waterfall sound
WTRFAL2.VOC	DIGI_WATERFALL2	177	(?)
WTRFLW1.VOC	DIGI_WATERFLOW1	178	Water flowing
WTRFLW2.VOC	DIGI_WATERFLOW2	179	(?)
ELEV1.VOC	DIGI_ELEVATOR	180	Elevator sound
EXPSML.VOC	DIGI_SMALLEXP	181	Explosion sounds
EXPMED.VOC	DIGI_MEDIUMEXP	182	
EXPLRG.VOC	DIGI_LARGEEXP	183	
BIGEXP.VOC	DIGI_HUGEEXP	184	
FIRE1.VOC	DIGI_FIRE1	185	Crackling fire
FIRE2.VOC	DIGI_FIRE2	186	Powerful fire
FBALL1.VOC	DIGI_FIREBALL1	187	Fireball trap
FIREBALL1.VOC	DIGI_FIREBALL2	188	
GEAR1.VOC	DIGI_GEAR1	189	Gear turning

GONG.VOC	DIGI_GONG	190	Gong
LAVAFLW1.VOC	DIGI_LAVAFLW1	191	Lava flowing ambience
MACHN1.VOC	DIGI_MACHINE1	192	Generic machine (?)
MUD1.VOC	DIGI_MUBBUBBLES1	193	Mud or lava bubbles
QUAKE1.VOC	DIGI_EARTHQUAKE	194	Earthquake sound
SEWER1.VOC	DIGI_SEWERFLOW1	195	Sewer noise (?)
SPLASH1.VOC	DIGI_SPLASH1	196	Generic splash
STEAM1.VOC	DIGI_STEAM1	197	Steam
VOLSTM1.VOC	DIGI_VOLCANOSTEAM1	198	Volcanic Steam (?)
STMPR.VOC	DIGI_STOMPER	199	(?)
SWAMP1.VOC	DIGI_SWAMP	200	Swamp ambience
SWITCH1.VOC	DIGI_REGULARSWITCH	201	Switch sounds
SWITCH2.VOC	DIGI_BIGSWITCH	202	
SWITCH3.VOC	DIGI_STONESWITCH	203	
SWITCH4.VOC	DIGI_GLASSSWITCH	204	
SWITCH5.VOC	DIGI_HUGESWITCH	205	
THUNDR.VOC	DIGI_THUNDER	206	Thunder ambience
TELPORT.VOC	DIGI_TELEPORT	207	Teleporter
UNDRWTR.VOC	DIGI_UNDERWATER	208	Underwater ambience
UNLOCK.VOC	DIGI_UNLOCK	209	Key unlock
VALVE.VOC	DIGI_SQUEAKYVALVE	210	
VOID1.VOC	DIGI_VOID1	211	Various weird ambience
VOID2.VOC	DIGI_VOID2	212	
VOID3.VOC	DIGI_VOID3	213	
VOID4.VOC	DIGI_VOID4	214	
VOID5.VOC	DIGI_VOID5	215	
ERUPT.VOC	DIGI_ERUPTION	216	Eruption ambience
VOLPRJCT.VOC	DIGI_VOLCANOPROJECTILE	217	(?)
WIND1.VOC	DIGI_LIGHTWIND	218	Wind ambience
WIND2.VOC	DIGI_STRONGWIND	219	
WOODBK.VOC	DIGI_BREAKINGWOOD	220	Breakage sounds (?)
STONEBRK.VOC	DIGI_BREAKSTONES	221	
ENGROOM1.VOC	DIGI_ENGROOM1	222	Various engine ambience
ENGROOM2.VOC	DIGI_ENGROOM2	223	
ENGROOM3.VOC	DIGI_ENGROOM3	224	
ENGROOM4.VOC	DIGI_ENGROOM4	225	
ENGROOM5.VOC	DIGI_ENGROOM5	226	
GLASS3.VOC	DIGI_BREAKGLASS	227	Breaking glass
MUSSTING.VOC	DIGI_MUSSTING	228	Musical stinger (?)
HELI.VOC	DIGI_HELI	229	Helicopter (?)
BIGHART.VOC	DIGI_BIGHART	230	Heart ambience (?)
WIND4.VOC	DIGI_WIND4	231	More wind
SPOOKY1.VOC	DIGI_SPOOKY1	232	Spooky sound (?)
DRILL1.VOC	DIGI_DRILL1	233	Another drill (?)
JET.VOC	DIGI_JET	234	Jet ambience
DRUMCHNT.VOC	DIGI_DRUMCHANT	235	Drum chanting
FLY.VOC	DIGI_BUZZZ	236	Chopstick sounds
UZICLK.VOC	DIGI_CHOP_CLICK	237	
STICKY2R.VOC	DIGI_SWORD_UP	238	Weapon up sounds
UZI1R.VOC	DIGI_UZI_UP	239	
SHOTG1R.VOC	DIGI_SHOTGUN_UP	240	
BOLT1R.VOC	DIGI_ROCKET_UP	241	
BOLT1R.VOC	DIGI_GRENADE_UP	242	
BOLT1R.VOC	DIGI_RAIL_UP	243	
STICKY1R.VOC	DIGI_MINE_UP	244	
NULL	DIGI_FIRSTPLAYERVOICE	245	Don't change this
BADMAN04.VOC	DIGI_TAUNTAI1	246	Various player
COMEGE2.VOC	DIGI_TAUNTAI2	247	taunts when killing
GHOP07.VOC	DIGI_TAUNTAI3	248	bad guys
GOODDAY4.VOC	DIGI_TAUNTAI4	249	

KILLU05.VOC	DIGI_TAUNTAI5	250	
NATURAL4.VOC	DIGI_TAUNTAI6	251	
NOHONOR6.VOC	DIGI_TAUNTAI7	252	
SAYON09.VOC	DIGI_TAUNTAI8	253	
TAKSAN1.VOC	DIGI_TAUNTAI9	254	
SNATCH01.VOC	DIGI_TAUNTAI10	255	
CHOTO7.VOC	DIGI_PLAYERPAIN1	256	Player pain sounds
SWPAIN05.VOC	DIGI_PLAYERPAIN2	257	
SWPAIN03.VOC	DIGI_PLAYERPAIN3	258	
SWPAIN07.VOC	DIGI_PLAYERPAIN4	259	
SWPAIN22.VOC	DIGI_PLAYERPAIN5	260	
SWYELL03.VOC	DIGI_PLAYERYELL1	261	Player yelling
SWYELL05.VOC	DIGI_PLAYERYELL2	262	
SWYELL06.VOC	DIGI_PLAYERYELL3	263	
GRUNT06.VOC	DIGI_SEARCHWALL	264	Grunt searching wall
NOWAY1.VOC	DIGI_NOURINAL	265	No way water cold
SWDIE02.VOC	DIGI_FALLSCREAM	266	Player scream
AHSO5.VOC	DIGI_GOTITEM1	267	Ahh so
NULL	DIGI_LASTPLAYERVOICE	268	Don't change this
HSHOT1.VOC	DIGI_RAILFIRE	269	Rail gun sounds
RAIL2.VOC	DIGI_RAILREADY	270	
RAILUP09.VOC	DIGI_RAILPWRUP	271	
HBOMB2.VOC	DIGI_NUCLEAREXP	272	Nuke sounds
STANDBY.VOC	DIGI_NUKESTDBY	273	
CDOWN.VOC	DIGI_NUKECDOWN	274	
SYSREAD.VOC	DIGI_NUKEREADY	275	
HISS1.VOC	DIGI_CHEMGAS	276	ChemBomb
CHBNCE1.VOC	DIGI_CHEMBOUNCE	277	
THROW.VOC	DIGI_THROW	278	
PULL.VOC	DIGI_PULL	279	
STSCAN2.VOC	DIGI_MINEARM	280	Other weapon/item sounds
HBDOWN1.VOC	DIGI_HEARTDOWN	281	
TOOLUSE1.VOC	DIGI_TOOLBOX	282	
GASPOP.VOC	DIGI_GASPOP	283	
40MMBNCE.VOC	DIGI_40MMBNCE	284	
BURGALRM.VOC	DIGI_BURGLARALARM	285	Various
CARALRM2.VOC	DIGI_CARALARM	286	
CAOFF1.VOC	DIGI_CARALARMOFF	287	
TACK1.VOC	DIGI_CALTROPS	288	
NVON3.VOC	DIGI_NIGHTON	289	
NVOFF2.VOC	DIGI_NIGHTOFF	290	
SGSH01.VOC	DIGI_SHOTSHELLSPENT	291	
SKID3.VOC	DIGI_BUSSKID	292	
CRASH4.VOC	DIGI_BUSCRASH	293	
BUS1.VOC	DIGI_BUSENGINE	294	
BIMP01.VOC	DIGI_ARMORHIT	295	
ASIREN1.VOC	DIGI_ASIREN1	296	Traffic ambience
FIRETRK1.VOC	DIGI_FIRETRK1	297	
TRAFFIC1.VOC	DIGI_TRAFFIC1	298	
TRAFFIC2.VOC	DIGI_TRAFFIC2	299	
TRAFFIC3.VOC	DIGI_TRAFFIC3	300	
TRAFFIC4.VOC	DIGI_TRAFFIC4	301	
TRAFFIC5.VOC	DIGI_TRAFFIC5	302	
TRAFFIC6.VOC	DIGI_TRAFFIC6	303	
HELI1.VOC	DIGI_HELI1	304	
JET1.VOC	DIGI_JET1	305	
MOTO1.VOC	DIGI_MOTO1	306	
MOTO2.VOC	DIGI_MOTO2	307	
NEON1.VOC	DIGI_NEON1	308	
SUBWAY1.VOC	DIGI_SUBWAY	309	
TRAINS01.VOC	DIGI_TRAIN1	310	

COIN.VOC	DIGI_COINS	311	
SWRDSMK1.VOC	DIGI_SWORDCLANK	312	
RIP2AMB.VOC	DIGI_RIPPER2AMBIENT	313	Big Ripper
RIP2ALRT.VOC	DIGI_RIPPER2ALERT	314	
RIP2ATCK.VOC	DIGI_RIPPER2ATTACK	315	
RIP2PAIN.VOC	DIGI_RIPPER2PAIN	316	
RIP2SCRM.VOC	DIGI_RIPPER2SCREAM	317	
RIP2HRT.VOC	DIGI_RIPPER2HEARTOUT	318	
M60.VOC	DIGI_M60	319	Enemy M60 fire
SUMSCRM.VOC	DIGI_SUMOSCREAM	320	Sumo Boss sounds
SUMALRT.VOC	DIGI_SUMOALERT	321	
SUMAMB.VOC	DIGI_SUMOAMBIENT	322	
SUMPAIN.VOC	DIGI_SUMOPAIN	323	
RAMLOCK.VOC	DIGI_RAMUNLOCK	324	Unlocking sounds
CARDLOCK.VOC	DIGI_CARDUNLOCK	325	
ACS10.VOC	DIGI_ANCIENTSECRET	326	More player voices
AMDRIV01.VOC	DIGI_AMERICANDRIVER	327	
BABOON03.VOC	DIGI_DRIVELIKEBABOON	328	
BBURN04.VOC	DIGI_BURNBABY	329	
BIGWPN01.VOC	DIGI_LIKEBIGWEAPONS	330	
CBUNG01.VOC	DIGI_COWABUNGA	331	
CHARAD09.VOC	DIGI_NOCHARADE	332	
DTIME.VOC	DIGI_TIMETODIE	333	
EAT02.VOC	DIGI_EATTHIS	334	
FCRACK01.VOC	DIGI_FIRECRACKERUPASS	335	
HCOW03.VOC	DIGI_HOLYCOW	336	
HCOW06.VOC	DIGI_HOLYPEICESOFCOW	337	
HSHIT03.VOC	DIGI_HOLYSHIT	338	
HSHIT04.VOC	DIGI_HOLYPEICESOFSHIT	339	
IHOPE01.VOC	DIGI_PAYINGATTENTION	340	
ILIKE01.VOC	DIGI_EVERYBODYDEAD	341	
KUNGFU06.VOC	DIGI_KUNGFU	342	
LMOVE01.VOC	DIGI_HOWYOULIKEMOVE	343	
LWANG05.VOC	DIGI_NOMESSWITHWANG	344	
RAW01.VOC	DIGI_RAWREVENGE	345	
STUPID01.VOC	DIGI_YOULOOKSTUPID	346	
TDICK02.VOC	DIGI_TINYDICK	347	
TOURN01.VOC	DIGI_NOTOURNAMENT	348	
WWANG11.VOC	DIGI_WHOWANTSWANG	349	
YAK02.VOC	DIGI_MOVELIKEYAK	350	
REFLEX08.VOC	DIGI_ALLINREFLEXES	351	
EVADE01.VOC	DIGI_EVADEFORVER	352	
MFLY03.VOC	DIGI_MRFLY	353	
SHISEI03.VOC	DIGI_SHISEISI	354	
FWORKS01.VOC	DIGI_LIKEFIREWORKS	355	
HIRO03.VOC	DIGI_LIKEHIROSHIMA	356	
NAGA06.VOC	DIGI_LIKENAGASAKI	357	
PEARL03.VOC	DIGI_LIKEPEARL	358	
IAM01.VOC	DIGI_IAMSHADOW	359	
LIKNUK01.VOC	DIGI_ILIKENUKES	360	
LIKSRD01.VOC	DIGI_ILIKESWORD	361	
LIKSHK02.VOC	DIGI_ILIKESHURIKEN	362	
LUCK06.VOC	DIGI_BADLUCK	363	
MCHAN01.VOC	DIGI_NOMOVIEMRCHAN	364	
RCHAN13.VOC	DIGI_REALLIFEMRCHAN	365	
MUSIC03.VOC	DIGI_NOLIKEMUSIC	366	
NODIFF07.VOC	DIGI_NODIFFERENCE	367	
NOFEAR01.VOC	DIGI_NOFEAR	368	
NOPAIN.VOC	DIGI_NOPAIN	369	
REPMAN15.VOC	DIGI_NOREPAIRMAN	370	
SOB15.VOC	DIGI_SONOFABITCH	371	



WEAK03.VOC	DIGI_PAINFORWEAK	372	
SPEED04.VOC	DIGI_GOSPEEDY	373	
STIFF01.VOC	DIGI_GETTINGSTIFF	374	
TOMB05.VOC	DIGI_TOMBRAIDER	375	
TSTICK01.VOC	DIGI_STICKYGOTU1	376	
TSTICK05.VOC	DIGI_STICKYGOTU2	377	
TSTICK07.VOC	DIGI_STICKYGOTU3	378	
TSTICK10.VOC	DIGI_STICKYGOTU4	379	
TSWORD05.VOC	DIGI_SWORDGOTU1	380	
TSWORD08.VOC	DIGI_SWORDGOTU2	381	
TSWORD01.VOC	DIGI_SWORDGOTU3	382	
SWYELL22.VOC	DIGI_HURTBAD1	383	
SWYELL14.VOC	DIGI_HURTBAD2	384	
SWYELL23.VOC	DIGI_HURTBAD3	385	
SWYELL16.VOC	DIGI_HURTBAD4	386	
SWYELL13.VOC	DIGI_HURTBAD5	387	
TGSCRM.VOC	DIGI_TOILETGIRLSCREAM	388	Girl on the toilet
TGALRT.VOC	DIGI_TOILETGIRLALERT	389	
TGAMB.VOC	DIGI_TOILETGIRLAMBIENT	390	
TGPAIN.VOC	DIGI_TOILETGIRLPAIN	391	
TGTNT1.VOC	DIGI_TOILETGIRLTAUNT1	392	
TGTNT2.VOC	DIGI_TOILETGIRLTAUNT2	393	
SUMOFART.VOC	DIGI_SUMOFART	394	Sumo fart attack
GIBS04.VOC	DIGI_GIBS1	395	Gib splats
GIBS05.VOC	DIGI_GIBS2	396	
BIRDS01.VOC	DIGI_BIRDS1	397	Bird ambience
BIRDS02.VOC	DIGI_BIRDS2	398	
TOILET01.VOC	DIGI_TOILET	399	Toilet flush
FLIDLE.VOC	DIGI_FORKLIFTIDLE	400	Forklift driving
FLRUN01.VOC	DIGI_FORKLIFTRUN	401	
TOYCAR03.VOC	DIGI_TOYCAR	402	
PRESS03.VOC	DIGI_UZIMATIC	403	
COMPON01.VOC	DIGI_COMPUTERPOWER	404	
TURBON01.VOC	DIGI_GENERATORON	405	
TURBRN01.VOC	DIGI_GENERATORRUN	406	
BIGDRL03.VOC	DIGI_BIGDRILL	407	
FLUOR01.VOC	DIGI_FLUORLIGHT	408	
AMOEB03.VOC	DIGI_AMOEB	409	
BODY6.VOC	DIGI_BODYFALL2	410	
GIBS01.VOC	DIGI_GIBS3	411	
CHOK01.VOC	DIGI_NINJACHOKE	412	
TRAIN3.VOC	DIGI_TRAIN3	413	
TRAINR02.VOC	DIGI_TRAINR02	414	
TRAIN8.VOC	DIGI_TRAIN8	415	
TCLID01.VOC	DIGI_TRASHLID	416	
ACCU01.VOC	DIGI_GETMEDKIT	417	Medkit player voices
AHH03.VOC	DIGI_AHH	418	
PALARM1.VOC	DIGI_PALARM	419	Pachinko sounds
PFLIP4.VOC	DIGI_PFLIP	420	
PROLL1.VOC	DIGI_PROLL1	421	
PROLL2.VOC	DIGI_PROLL2	422	
PROLL3.VOC	DIGI_PROLL3	423	
RABATK1.VOC	DIGI_BUNNYATTACK	424	Bunny sounds
RABDIE1.VOC	DIGI_BUNNYDIE1	425	
RABDIE2.VOC	DIGI_BUNNYDIE2	426	
RABDIE3.VOC	DIGI_BUNNYDIE3	427	
RABAMB.VOC	DIGI_BUNNYAMBIENT	428	
STONE2.VOC	DIGI_STONESLIDE	429	
NINCUT3.VOC	DIGI_NINJAINHALF	430	
RIPCHST1.VOC	DIGI_RIPPER2CHEST	431	
WHIPIN2.VOC	DIGI_WHIPME	432	Whipping TV show

ENDLEV3.VOC	DIGI_ENDLEV	433	
MDALARM1.VOC	DIGI_MDALARM	434	Metal detector
METALBRK.VOC	DIGI_BREAKMETAL	435	Breakage sounds
DEBRIBRK.VOC	DIGI_BREAKDEBRIS	436	
MARBELS.VOC	DIGI_BREAKMARBELS	437	
BANZAI1.VOC	DIGI_BANZAI	438	Player voices
HAHA19.VOC	DIGI_HAHA1	439	
HAHA11.VOC	DIGI_HAHA2	440	
HAHA15.VOC	DIGI_HAHA3	441	
TELEPT02.VOC	DIGI_ITEM_SPAWN	442	Item respawns
JG1075.VOC	DIGI_NOREPAIRMAN2	443	Player voices
JG1082.VOC	DIGI_NOPOWER	444	
JG1087.VOC	DIGI_DOUBLEUZI	445	
JG1088.VOC	DIGI_NOTORDBUNNY	446	
JG1103.VOC	DIGI_CANBEONLYONE	447	
JG2000.VOC	DIGI_MIRROR1	448	
JG2005.VOC	DIGI_MIRROR2	449	
JG2020.VOC	DIGI_HITTINGWALLS	450	
JG2032.VOC	DIGI_GOTRAILGUN	451	
JG2053.VOC	DIGI_RABBITHUMP1	452	
JG2054.VOC	DIGI_RABBITHUMP2	453	
JG2045.VOC	DIGI_RABBITHUMP3	454	
JG2087.VOC	DIGI_RABBITHUMP4	455	
JG2074.VOC	DIGI_FAGRABBIT1	456	
JG2075.VOC	DIGI_FAGRABBIT2	457	
JG2078.VOC	DIGI_FAGRABBIT3	458	
JG3005.VOC	DIGI_STINKLIKEBABBOON	459	
JG3017.VOC	DIGI_WHATYOUPEATBABY	460	
JG3047.VOC	DIGI_WHATDIEDUPHERE	461	
JG3022.VOC	DIGI_YOUGOPOOPOO	462	
JG6053.VOC	DIGI_PULLMYFINGER	463	
JG3059.VOC	DIGI_SOAPYOUGOOD	464	
JG4012.VOC	DIGI_WASHWANG	465	
JG3070.VOC	DIGI_DROPSOAP	466	
JG6051.VOC	DIGI_REALTITS	467	
JG4002.VOC	DIGI_MSTRLEEP	468	
JG4024.VOC	DIGI_SEEKLEEPADVICE	469	
JG5042.VOC	DIGI_AVENGELEEPDEATH	470	
JG5049A.VOC	DIGI_LEEPGHOST	471	
WDOOR02.VOC	DIGI_DOOR1	472	More door sounds
MDOOR03.VOC	DIGI_DOOR2	473	
603981_1.VOC	DIGI_DOOR3	474	
FLAG03.VOC	DIGI_FLAGWAVE	475	Flag ambience
JG7009.VOC	DIGI_SURFACE	476	
JG7001.VOC	DIGI_GASHURT	477	
JG2001.VOC	DIGI_BONUS_GRAB	478	
ACHCRY04.VOC	DIGI_ANIMECRY	479	Girl sounds
ACHS010.VOC	DIGI_ANIMESING1	480	
ACHT1006.VOC	DIGI_ANIMEMAD1	481	
ACHS016.VOC	DIGI_ANIMESING2	482	
ACHT120A.VOC	DIGI_ANIMEMAD2	483	
TELEPT02.VOC	DIGI_PLAYER_TELEPORT	484	Player teleporting
SLASH1.VOC	DIGI_INTRO_SLASH	485	
WARNING.VOC	DIGI_WARNING	486	
WHIRL1.VOC	DIGI_INTRO_WHIRL	487	
ACHF1003.VOC	DIGI_TOILETGIRLFART1	488	Toilet girl sounds
ACHF1002.VOC	DIGI_TOILETGIRLFART2	489	
ACHF1016.VOC	DIGI_TOILETGIRLFART3	490	
CHIMES4.VOC	DIGI_WINDCHIMES	491	Wind Chime ambience
JGB023.VOC	DIGI_MADATCARPET	492	Player voices

JGB020.VOC	DIGI_JUMPONCARPET	493	
JGB080.VOC	DIGI_USEBROKENVEHICLE	494	
JGB106.VOC	DIGI_STEPONCALTROP	495	
JGB130.VOC	DIGI_WANGSEESERP	496	
JGSB4.VOC	DIGI_SERPTAUNTWANG	497	
JGB166.VOC	DIGI_WANGTAUNTSERP1	498	
JGB156.VOC	DIGI_WANGTAUNTSERP2	499	
JGB193.VOC	DIGI_WANGORDER1	500	
JGB202.VOC	DIGI_WANGORDER2	501	
JGB340A.VOC	DIGI_WANGDROWNING	502	
JGEN06.VOC	DIGI_ZILLAREGARDS	503	
MSG9.VOC	DIGI_PMESSAGE	504	
UGLY1A.VOC	DIGI_SHAREND_UGLY1	505	First episode ending
UGLY1B.VOC	DIGI_SHAREND_UGLY2	506	
TELEPT07.VOC	DIGI_SHAREND_TELEPORT	507	
GOROSW1.VOC	DIGI_HOTHEADSWITCH	508	Change head weapon mode
BTCREAK2.VOC	DIGI_BOATCREAK	509	New boat sounds
BTRUN05.VOC	DIGI_BOATRUN2	510	
BTIDLE4.VOC	DIGI_BOATIDLE	511	
SHIPBELL.VOC	DIGI_SHIPBELL	512	
SHIPHRN1.VOC	DIGI_FOGHORN	513	
EXP3.VOC	DIGI_CANNON	514	Cannon firing
JG41001.VOC	DIGI_JG41001	515	Player voices
JG41012.VOC	DIGI_JG41012	516	Most of these will be
JG41018.VOC	DIGI_JG41018	517	the map specific stuff
JG41028.VOC	DIGI_JG41028	518	
JG41048.VOC	DIGI_JG41048	519	
JG41052.VOC	DIGI_JG41052	520	
JG41058.VOC	DIGI_JG41058	521	
JG41060.VOC	DIGI_JG41060	522	
JG41075.VOC	DIGI_JG41075	523	
JG42004.VOC	DIGI_JG42004	524	
JG42019.VOC	DIGI_JG42019	525	
JG42021.VOC	DIGI_JG42021	526	
JG42028.VOC	DIGI_JG42028	527	
JG42033.VOC	DIGI_JG42033	528	
JG42034.VOC	DIGI_JG42034	529	
JG42050.VOC	DIGI_JG42050	530	
JG42056.VOC	DIGI_JG42056	531	
JG42061.VOC	DIGI_JG42061	532	
JG43004.VOC	DIGI_JG43004	533	
JG43015.VOC	DIGI_JG43015	534	
JG43019.VOC	DIGI_JG43019	535	
JG43021.VOC	DIGI_JG43021	536	
JG44011.VOC	DIGI_JG44011	537	
JG44014.VOC	DIGI_JG44014	538	
JG44027.VOC	DIGI_JG44027	539	
JG44038.VOC	DIGI_JG44038	540	
JG44039.VOC	DIGI_JG44039	541	
JG44048.VOC	DIGI_JG44048	542	
JG44052.VOC	DIGI_JG44052	543	
JG45014.VOC	DIGI_JG45014	544	
JG44068.VOC	DIGI_JG44068	545	
JG45010.VOC	DIGI_JG45010	546	
JG45018.VOC	DIGI_JG45018	547	
JG45030.VOC	DIGI_JG45030	548	
JG45033.VOC	DIGI_JG45033	549	
JG45043.VOC	DIGI_JG45043	550	
JG45053.VOC	DIGI_JG45053	551	
JG45067.VOC	DIGI_JG45067	552	
JG46005.VOC	DIGI_JG46005	553	

JG46010.VOC	DIGI_JG46010	554	
LANI049.VOC	DIGI_LANI049	555	Race Queen talks back
LANI051.VOC	DIGI_LANI051	556	
LANI052.VOC	DIGI_LANI052	557	
LANI054.VOC	DIGI_LANI054	558	
LANI060.VOC	DIGI_LANI060	559	Sailor Girl talks back
LANI063.VOC	DIGI_LANI063	560	
LANI065.VOC	DIGI_LANI065	561	
LANI066.VOC	DIGI_LANI066	562	
LANI073.VOC	DIGI_LANI073	563	Mechanic girl talks back
LANI075.VOC	DIGI_LANI075	564	
LANI077.VOC	DIGI_LANI077	565	
LANI079.VOC	DIGI_LANI079	566	
LANI089.VOC	DIGI_LANI089	567	Old prune talks back
LANI091.VOC	DIGI_LANI091	568	
LANI093.VOC	DIGI_LANI093	569	
LANI095.VOC	DIGI_LANI095	570	
AD5.VOC	DIGI_VENTWALK	571	Climb in a vent
AD6.VOC	DIGI_CARWALK	572	Walk on a car
JET05.VOC	DIGI_JETSOAR	573	Crashing jet
VC04.VOC	DIGI_VACUUM	574	Driving a vacuum cleaner
LANI017.VOC	DIGI_GIRLNINJAALERTT	575	Female Warrior sounds
LANI033.VOC	DIGI_GIRLNINJASCREAM	576	
LANI001.VOC	DIGI_GIRLNINJAALERT	577	
CAKLE.VOC	DIGI_PRUNECACKLE	578	Old prune laughing
CAKLE2.VOC	DIGI_PRUNECACKLE2	579	
CAKLE3.VOC	DIGI_PRUNECACKLE3	580	
SUMO058.VOC	DIGI_SUMOSTOMP	581	Sumo stomping
ELEV01.VOC	DIGI_VATOR	582	Long elevator sound
JG9009.VOC	DIGI_JG9009	583	Player voice
Z16004.VOC	DIGI_Z16004	584	Zilla voices
Z16012.VOC	DIGI_Z16012	585	
Z16022.VOC	DIGI_Z16022	586	
Z16027.VOC	DIGI_Z16027	587	
JG93030.VOC	DIGI_JG93030	588	Player voice
JG94002.VOC	DIGI_JG94002	589	
Z17010.VOC	DIGI_Z17010	590	Zilla voices
Z17052.VOC	DIGI_Z17052	591	
Z17025.VOC	DIGI_Z17025	592	
ML25014.VOC	DIGI_ML25014	593	Master Leep and
ML250101.VOC	DIGI_ML250101	594	more player voices
JG9022.VOC	DIGI_JG9022	595	
JG9032.VOC	DIGI_JG9032	596	
JG9038.VOC	DIGI_JG9038	597	
JG9055.VOC	DIGI_JG9055	598	
JG9060.VOC	DIGI_JG9060	599	
JG92055.VOC	DIGI_JG92055	600	
ML25032.VOC	DIGI_ML25032	601	
JG92036.VOC	DIGI_JG92036	602	
JG92042.VOC	DIGI_JG92042	603	
ML26001.VOC	DIGI_ML26001	604	
JG93000.VOC	DIGI_JG93000	605	
JG93011.VOC	DIGI_JG93011	606	
JG93018.VOC	DIGI_JG93018	607	
JG93023.VOC	DIGI_JG93023	608	
ML26008.VOC	DIGI_ML26008	609	
ML26011.VOC	DIGI_ML26011	610	
JG94007.VOC	DIGI_JG94007	611	
JG94024.VOC	DIGI_JG94024	612	
JG94039.VOC	DIGI_JG94039	613	
JG95012.VOC	DIGI_JG95012	614	

## 4.II) Ambient Sound Reference

Ambient sounds do not work the same as digital sounds. Rather than triggering them with a door or something, the game either constantly or intermittently plays the ambient sound in an area of the map. They're used for waterfalls, wind, or other environmental sound effects.

This section is divided into two columns. The first column is an identifier that the programmers used to name the ambient sound. The second column is the sound entry number for use with the Lotag (TAG2) value of the AMBIENT\_SOUND ST1 (hitag 1002). Do **NOT** use these sound numbers with SOUND\_SPOT ST1's (hitag 134), or the ambient sound won't work properly.

All of these ambient sounds have an entry with a matching identifier in the digital sound index above.

**NOTE:** Some of these ambient sounds were not used in the game. Feel free to place your own custom ambient sounds in these slots.

<i>Identity</i>	<i>No.</i>
DIGI_BUBBLES	0
DIGI_CRICKETS	1
DIGI_CAVEDRIP1	2
DIGI_CAVEDRIP2	3
DIGI_DRIP	4
DIGI_WATERFALL1	5
DIGI_WATERFALL2	6
DIGI_WATERFLOW1	7
DIGI_WATERFLOW2	8
DIGI_FIRE1	9
DIGI_FIRE2	10
DIGI_GONG	11
DIGI_LAVAFLOW1	12
DIGI_MUBBUBBLES1	13
DIGI_EARTHQUAKE	14
DIGI_SEWERFLOW1	15
DIGI_STEAM1	16
DIGI_VOLCANOSTEAM1	17
DIGI_SWAMP	18
DIGI_THUNDER	19
DIGI_UNDERWATER	20
DIGI_VOID1	21
DIGI_VOID2	22
DIGI_VOID3	23
DIGI_VOID4	24
DIGI_VOID5	25
DIGI_ERUPTION	26
DIGI_VOLCANOPROJECTILE	27
DIGI_LIGHTWIND	28
DIGI_STRONGWIND	29
DIGI_BREAKINGWOOD	30
DIGI_BREAKSTONES	31
DIGI_BOMBRFLYING	40
DIGI_BOMBRDROPBOMB	41
DIGI_DRILL	42
DIGI_GEAR1	43
DIGI_MACHINE1	44
DIGI_ENGROOM1	45

<i>Identity</i>	<i>No.</i>
DIGI_ENGROOM2	46
DIGI_ENGROOM3	47
DIGI_ENGROOM4	48
DIGI_ENGROOM5	49
DIGI_HELI	50
DIGI_BIGHART	51
DIGI_WIND4	52
DIGI_SPOOKY1	53
DIGI_JET	54
DIGI_DRUMCHANT	55
DIGI_ASIREN1	56
DIGI_FIRETRK1	57
DIGI_TRAFFIC1*	58*
DIGI_TRAFFIC2	59
DIGI_TRAFFIC3	60
DIGI_TRAFFIC4	61
DIGI_TRAFFIC5	62
DIGI_TRAFFIC6	63
DIGI_HELI1	64
DIGI_JET1	65
DIGI_MOTO1	66
DIGI_MOTO2	67
DIGI_NEON1	68
DIGI_SUBWAY	69
DIGI_TRAIN1	70
DIGI_BIRDS1	71
DIGI_BIRDS2	72
DIGI_AMOEBA	73
DIGI_TRAIN3	74
DIGI_TRAIN8	75
DIGI_WHIPME	76
DIGI_FLAGWAVE	77
DIGI_ANIMECRY	78
DIGI_WINDCHIMES	79
DIGI_BOATCREAK	80
DIGI_SHIPBELL	81
DIGI_FOGHORN	82

(\*) No. 58: DIGI\_TRAFFIC1 has special functionality, it randomly picks a traffic ambient sound.

## 5) Sector Tag Reference

by Keith Schuler, revised by Steffen Itterheim

Many things can be done in Shadow Warrior using sector tags. In general, the sector's lotag is used to identify the effect the sector will produce, while the hitag is used to supply supplemental information like a match tag or a track to follow. Some examples of these tags can be found in EXAMPLE.MAP, so be sure to check that map out.

### Sector Tag Reference Contents

<b>5.I) Sector Triggers</b>	<b>22</b>
43 TAG_TRIGGER_MISSILE_TRAP	
116 TAG_LEVEL_EXIT_SWITCH	
206 TAG_SECTOR_TRIGGER_VATOR	
210 TAG_LIGHT_TRIGGER	
211 TAG_TRIGGER_EVERYTHING	
212 TAG_TRIGGER_EVERYTHING_ONCE	
216 TAG_SPAWN_ACTOR_TRIGGER	
217 TAG_SECRET_AREA_TRIGGER	
240 TAG_SPRING_BOARD	
<b>5.II) Sine Wave on Floor and/or Ceiling</b>	<b>24</b>
400 TAG_SINE_WAVE_FLOOR	
420 TAG_SINE_WAVE_CEILING	
440 TAG_SINE_WAVE_BOTH	
<b>5.III) Sector Object Sector Tags</b>	<b>25</b>
501 TAG_OBJECT_CENTER	

### 5.I) Sector Triggers

A „trigger“ is the term in Shadow Warrior for a sector that triggers a tag event with the same match tag when the player steps into it, much like a switch does. In Duke Nukem 3D, these triggers were called „touchplates“, but the general idea is the same. The sector triggers available in Shadow Warrior are as follows.

<b>Lotag 43</b>	<b>TAG_TRIGGER_MISSILE_TRAP</b>
	Hitag = match tag
	Shoots a fireball from the FIREBALL_TRAP (hitag 43), BOLT_TRAP (hitag 44) or the SPEAR_TRAP (hitag 63) ST1 sprite that has a lotag matching the sector's hitag.
<b>Lotag 116</b>	<b>TAG_LEVEL_EXIT_SWITCH</b>

Hitag = number of next level

When the player steps into this sector, the level will end. The game will then proceed to the level listed in the sector's hitag, provided the current level has the same filename as one of the levels that came with Shadow Warrior. For most external maps, a hitag of 0 should be sufficient.

**Lotag 206 TAG\_SECTOR\_TRIGGER\_VATOR**

Hitag = match tag

Triggers a vator, spike, slider or rotator with a Lotag (TAG2) that matches the hitag of this sector.

**Lotag 210 TAG\_LIGHT\_TRIGGER**

Hitag = match tag

Triggers all LIGHTING ST1's (hitag 108) with a Lotag (TAG2) equal to the hitag of this sector when the player steps into it.

**Lotag 211 TAG\_TRIGGER\_EVERYTHING**

Hitag = match tag

Stepping into this sector triggers any event with a matching tag.

**Lotag 212 TAG\_TRIGGER\_EVERYTHING\_ONCE**

Hitag = match tag

Stepping into this sector triggers any event with a matching tag once, and only once.

**Lotag 216 TAG\_SPAWN\_ACTOR\_TRIGGER**

Hitag = match tag

Give an actor a lotag of 203, and a matching hitag, and he will not appear normally at the map startup. Instead, he will teleport in when the player steps on this sector.

**Lotag 217 TAG\_SECRET\_AREA\_TRIGGER**

Marks a sector as a secret area. The player will be rewarded with the „found secret“ message when he steps on this sector.

**Lotag 240 TAG\_SPRING\_BOARD**

Hitag = height of the jump

Step on the sector, press the SPACEBAR, and it shoots you high into the air. Hitag designates how high the player will jump. One example of this is the ventilation shaft at the end of \$NEWMINE.MAP.

## 5.II) Sine Wave on Floor and/or Ceiling

You can have a maximum of five of these per level. See room GA in EXAMPLE.MAP to see how this is done. Lotag the first sector with 400, the next with 401, and so on - up to a lotag of 419 (meaning you can have up to 20 wave sectors in a row). The wave will actually flow from the opposite end. The wave direction is what is referred to when BEGINNING and END are referred to below.

### Required Tags for all Sine Waves

<b>1<sup>st</sup> sector's hitag</b>	The range in pixels the floor will undulate in the Z (up/down) direction
<b>2<sup>nd</sup> sector's hitag</b>	Range decrement - takes the range from the 1st sector's high tag and adjusts all the rest by this decrement - makes the wave start off small and get larger toward the end - THIS IS IN Z COORDINATES NOT PIXELS!
<b>3<sup>rd</sup> sector's hitag</b>	Speed of the wave motion - defaults to 3
<b>4th sector's hitag</b>	Number of sectors from one peak (top of the curve) to the next. The default is the total number of sectors in the entire sine wave (max of 20). In other words this defines the amount of waves the wave sectors form in each oscillation.
<b>Last sector's hitag</b>	Special tag to make a realistic ocean wave - modifies the range toward the end (greatest range is usually here) of the wave so that it dissapates quickly. Without this it would just continue to get bigger toward the end of the wave.

### Lotag 400 TAG\_SINE\_WAVE\_FLOOR

Hitag = range in Z pixels floor will undulate

First sector's lotag must be set to 400, next sector's lotag must be 401, then 402, all the way up to a maximum number of 419, meaning you can have up to 20 adjacent wave sectors. Creates a sine wave on the floor. Slope the sectors in one direction to make a smooth, sloping sine wave.

### Lotag 420 TAG\_SINE\_WAVE\_CEILING

Hitag = range in Z pixels floor will undulate

First sector's lotag must be set to 420, next sector's lotag must be 421, then 422, all the way up to a maximum number of 439, meaning you can have up to 20 adjacent wave sectors. Creates a sine wave effect on the ceiling. Slope the sectors in one direction to make a smooth, sloping sine wave.

### Lotag 440 TAG\_SINE\_WAVE\_BOTH



Hitag = range in Z pixels floor will undulate

First sector's lotag must be set to 440, next sector's lotag must be 441, then 442, all the way up to a maximum number of 459, meaning you can have up to 20 adjacent wave sectors. Creates a sine wave effect on both the floor and ceiling. Slope the sectors in one direction to make a smooth, sloping sine wave.

### 5.III) Sector Object Sector Tags

Sector objects (SO's) are groups of sectors that can be moved around together as a unit. For example the remote controlled mini-cars in the first Shareware level \$BULLET.MAP or the crashing Minivan and the gun turret in the same level are Sector Objects.

#### Required Global Setup for all Sector Objects

<b>Bounding Sprites</b>	Bounding sprites (upper left and lower right) that form an imaginary rectangle to contain all sectors and sprites to be moved. SECT_SO_BOUNDING ST1 sprites (hitags 500-600) are used for this.
<b>Center Sector</b>	A sector marked as the center of the Sector Object (lotags 501/506/511/etc.).
<b>Enclosing Sector</b>	An outer wall loop which forms an unbroken single sector that entirely encircles the Sector Object. One of the walls in this loop must have a lotag of 504.

Sector lotags 500-600 are set reserved for Sector Objects. There is a maximum of 20 Sector Objects per level, each getting 5 reserved tags (500-504, 505-509, 510-514, etc.). Lotags 500-504 correspond to object 0, 505-509 to object 1, 510-514 to object 2, and so on but only the first two tags are used (500 and 501, 505 and 506, etc.). The ST1 sprites BOUND\_SO\_UPPER and BOUND\_SO\_LOWER always use hitags 500 and 501 for object 0, hitags 505 and 506 for object 1, and so on. The hitag of the BOUND\_SO\_LOWER ST1 sprite corresponds to the object's center sector lotag, as seen in the table below.

<b>Lotag 501</b>	<b>TAG_OBJECT_CENTER</b> (for Sector Object 0)
<b>Lotag 506</b>	<b>TAG_OBJECT_CENTER</b> (for Sector Object 1)
<b>Lotag 511</b>	<b>TAG_OBJECT_CENTER</b> (for Sector Object 2)
<b>Lotag 516</b>	<b>TAG_OBJECT_CENTER</b> (for Sector Object 3)
<b>Lotag 521</b>	<b>TAG_OBJECT_CENTER</b> (for Sector Object 4)
<b>Lotag 526</b>	<b>TAG_OBJECT_CENTER</b> (for Sector Object 5)
.	.
.	.
.	.
<b>etc.</b>	<b>etc.</b>

Hitag	if this is set to: <ul style="list-style-type: none"> <li>-1 = Sector Object doesn't follow a track</li> <li>0-94 = track number SO follows</li> <li>95 = Sector Object is killable</li> <li>96 = auto turret (always aims at player)</li> <li>98 = operational (for player controlled SOs)</li> </ul>
-------	---

Lotag 501 tags this sector as the center sector of SO 0, while lotag 506 would tag the sector as center sector of SO 1, and so on. The center sector lotag must always be the same as the BOUND\_SO\_LOWER ST1 sprite hitag (hitags 501/506/511/516/etc.). The BOUND\_SO\_UPPER ST1 sprite will always have the BOUND\_SO\_LOWER hitag minus 1 (hitags 500/505/510/515/etc.). The sector hitag defines how the Sector Object will behave.

## 6) Wall Tag Reference

by Keith Schuler

Several things can be done in Shadow Warrior using wall tags. In general, the wall's lotag is used to identify the effect the wall will produce, while the hitag is used to supply supplemental information like a match tag to link with another effect. Some examples of these tags can be found in EXAMPLE.MAP, so be sure to check that map out.

### Wall Tag Reference Contents

<b>6.I) Sine Wave Wall Effect</b>	<b>26</b>
300 TAG_WALL_SINE_Y_BEGIN	
301 TAG_WALL_SINE_X_BEGIN	
302 TAG_WALL_SINE_Y_END	
303 TAG_WALL_SINE_X_END	
<b>6.II) Misc Wall Tags</b>	<b>27</b>
305 TAG_WALL_CLIMB	
306 TV_CAMERA_SCREEN	
307 TAG_WALL_BREAK	
900 TAG_WALL_PLAY_SOUND	
<b>6.III) Sector Object Wall Tags</b>	<b>28</b>
500 TAG_WALL_LOOP_DONT_SPIN	
501 TAG_WALL_LOOP_REVERSE_SPIN	
502 TAG_WALL_LOOP_SPIN_2X	
503 TAG_WALL_LOOP_SPIN_4X	
504 TAG_WALL_LOOP_OUTER	
507 TAG_WALL_LOOP_DONT_SCALE	
508 TAG_WALL_LOOP_OUTER_SECONDARY	
550 TAG_WALL_ALIGN_SLOPE_TO_POINT	

### 6.I) Sine Wave Wall Effect

You can have a maximum of 5 of these per level. Lotag the first wall to sine wave with 300, the last one with 302. Look in room GB in EXAMPLE.MAP to see how this is done.

## Required Tags for all Sine Waves

<b>1<sup>st</sup> wall's hitag</b>	The range in pixels the wall will undulate.
<b>2<sup>nd</sup> wall's hitag</b>	Speed of the wave motion - defaults to 3
<b>3<sup>rd</sup> wall's hitag</b>	The distance from one peak (top of the curve) to the next. The default is the total number of walls in the entire sine wave (max of 20). In other words this defines the amount of waves the wave walls form in each oscillation.

<b>Lotag 300</b>	<b>TAG_WALL_SINE_Y_BEGIN</b>
------------------	------------------------------

Hitag = range wall will undulate
Use this to start a sine wave effect on a horizontal (east-west) wall. Sine wave will travel along the wall until it encounters a wall with a lotag of 302.

<b>Lotag 301</b>	<b>TAG_WALL_SINE_X_BEGIN</b>
------------------	------------------------------

Hitag = range wall will undulate
Use this to start a sine wave effect on a vertical (north-south) wall. Sine wave will travel along the wall until it encounters a wall with a lotag of 303.

<b>Lotag 302</b>	<b>TAG_WALL_SINE_Y_END</b>
------------------	----------------------------

End a horizontal (east-west) sine wave wall effect.
---

<b>Lotag 303</b>	<b>TAG_WALL_SINE_X_END</b>
------------------	----------------------------

End a vertical (north-south) sine wave wall effect.
---

## 6.II) Misc Wall Tags

<b>Lotag 305</b>	<b>TAG_WALL_CLIMB</b>
------------------	-----------------------

Climbable wall, used with SPRI_CLIMB_MARKER ST1 (hitag 30). You may only use this walltag on a red (2-sided) wall. Ladders aren't allowed on white (1-sided) walls.
---

<b>Lotag 306</b>	<b>TV_CAMERA_SCREEN</b>
------------------	-------------------------

Hitag = match tag
Use this walltag only on a mirror. The hitag is a match that equals the Lotag (TAG2) value of a MIRROR_CAM ST1 (hitag 1000). Allows a mirror to display the view of the camera instead of the mirror reflection.

<b>Lotag 307</b>	<b>TAG_WALL_BREAK</b>
------------------	-----------------------

Hitag = match tag	
Used in conjunction with BREAKABLE ST1 (hitag 132). Marks a wall as being breakable. The wall's hitag must equal the Lotag (TAG2) value of the BREAKABLE ST1 sprite.	

<b>Hitag 900</b>	<b>TAG_WALL_PLAY_SOUND</b>
------------------	----------------------------

Hitag	900 (see description)
Lotag	number of sound effect to play (see Digital Sound Reference)
<i>Notice that the hitag is 900, not the lotag!</i> When the player presses SPACEBAR against this wall, the sound effect designated by the lotag will be played.	

### 6.III) Sector Object Wall Tags

These are mainly for Sector Objects that rotate. Tag **ONLY ONE** wall of the loop (sector) to obtain the desired effect. See room FD in EXAMPLE.MAP for an example of how to use this kind of walltag.

<b>Lotag 500</b>	<b>TAG_WALL_LOOP_DONT_SPIN</b>
This wall loop (sector) will not spin along with the rest of the Sector Object.	

<b>Lotag 501</b>	<b>TAG_WALL_LOOP_REVERSE_SPIN</b>
This wall loop (sector) will spin in the opposite direction of the rest of the Sector Object.	

<b>Lotag 502</b>	<b>TAG_WALL_LOOP_SPIN_2X</b>
This wall loop (sector) will spin two times as fast as the rest of the Sector Object.	

<b>Lotag 503</b>	<b>TAG_WALL_LOOP_SPIN_4X</b>
This wall loop (sector) will spin four times as fast as the rest of the Sector Object.	

<b>Lotag 504</b>	<b>TAG_WALL_LOOP_OUTER</b>
All Sector Objects use this. Tags the outermost bounding wall loop (sector) of the Sector Object.	

<b>Lotag 507</b>	<b>TAG_WALL_LOOP_DONT_SCALE</b>
------------------	---------------------------------

Tag one wall in a loop (sector) you don't want to scale with the rest of the Sector Object

<b>Lotag 508</b>	<b>TAG_WALL_LOOP_OUTER_SECONDARY</b>
	Rectangular clipping doesn't work too well if you've got an operational Sector Object in a stacked sector (room over room) area. Use lotag 508 on the secondary Sector Object instead of the usual 504 to make rectangular clipping behave. See the driveable boat in \$YAMATO.MAP for an example of how to use this.

<b>Lotag 550</b>	<b>TAG_WALL_ALIGN_SLOPE_TO_POINT</b>
	See room ED in EXAMPLE.MAP for an example of this. Used to make all slopes in a Sector Object align to a single point, usually as the point moves up and down.

## 7) Sector Effector (ST1) Sprite Reference

by Keith Schuler, revised by Steffen Itterheim

### *Sector Effector Sprite Reference Contents*

<b>7.I) Meet the ST1 Sprite</b>	<b>31</b>
7.I.1 Using the ST1 Sprite	
7.I.2 ST1 Functions	
7.I.3 ST1 Context Sensitive Help	
<b>7.II) The Shadow Warrior Tag System</b>	<b>31</b>
7.II.1 The 15 Standard Tags	
7.II.2 The 11 Boolean Flags	
7.II.3 The Hitag (TAG1)	
7.II.4 The Lotag (TAG2) and Match Tags	
7.II.5 The TAG4 Tag (Sprite Angle)	
7.II.6 The BOOL1 Tag	
7.II.7 The BOOL11 Tag	
<b>7.III) Notes Regarding the ST1 Sprite</b>	<b>33</b>
7.III.1 Z-Height	
7.III.2 Color Coding the ST1's	
7.III.3 Brighten the ST1's	
<b>7.IV) The Shadow Warrior ST1 Function Reference</b>	<b>34</b>
0 SECT_SINK	
1 SECT_OPERATIONAL	
3 SECT_CURRENT	
5 SECT_NO_RIDE	
7 SECT_DIVE_AREA	
8 SECT_UNDERWATER	
9 SECT_UNDERWATER2	
16 SO_ANGLE	
19 SECT_FLOOR_PAN	
21 SECT_CEILING_PAN	
23 WALL_PAN_SPEED	
29 SECT_LOCK_DOOR	
30 SPRI_CLIMB_MARKER	
31 SECT_SO_SPRITE_OBJ	
32 SECT_SO_DONT_BOB	

33 SECT\_SO\_SINK\_DEST  
34 SECT\_SO\_DONT\_SINK  
37 SECT\_SO\_FORM\_WHIRLPOOL  
38 SECT\_ACTOR\_BLOCK  
42 MULTI\_PLAYER\_START  
43 FIREBALL\_TRAP  
44 BOLT\_TRAP  
45 SECT\_SO\_DONT\_ROTATE  
46 PARALLAX\_LEVEL  
47 SECT\_DONT\_COPY\_PALETTE  
48 MULTI\_COOPERATIVE\_START  
49 SO\_SET\_SPEED  
50 SO\_SPIN  
51 SO\_SPIN\_REVERSE  
52 SO\_BOB\_START  
53 SO\_BOB\_SPEED  
55 SO\_TURN\_SPEED  
56 LAVA\_ERUPT  
57 SECT\_EXPLODING\_CEIL\_FLOOR  
58 SECT\_COPY\_DEST  
59 SECT\_COPY\_SOURCE  
62 SHOOT\_POINT  
63 SPEAR\_TRAP  
64 SO\_SYNC1  
65 SO\_SYNC2  
66 DEMO\_CAMERA  
68 SO\_LIMIT\_TURN  
69 SPAWN\_SPOT  
70 SO\_MATCH\_EVENT  
71 SO\_SLOPE\_FLOOR\_TO\_POINT  
72 SO\_SLOPE\_CEILING\_TO\_POINT  
73 SO\_TORNADO  
75 SO\_AMOEBE  
76 SO\_MAX\_DAMAGE  
77 SO\_RAM\_DAMAGE  
80 SO\_KILLABLE  
81 SO\_AUTO\_TURRET  
82 SECT\_DAMAGE  
84 WARP\_TELEPORTER  
85 WARP\_CEILING\_PLANE  
86 WARP\_FLOOR\_PLANE  
87 WARP\_COPY\_SPRITE1  
88 WARP\_COPY\_SPRITE2  
90 PLAX\_GLOB\_Z\_ADJUST  
92 SECT\_VATOR  
97 CEILING\_Z\_ADJUST  
98 FLOOR\_Z\_ADJUST  
99 FLOOR\_SLOPE\_DONT\_DRAW  
100 SO\_SCALE\_INFO  
101 SO\_SCALE\_POINT\_INFO  
102 SO\_SCALE\_XY\_MULTIPLY  
103 SECT\_WALL\_MOVE  
104 SECT\_WALL\_MOVE\_CANSEE  
106 SECT\_SPIKE  
108 LIGHTING  
109 LIGHTING\_DIFFUSE  
110 VIEW\_LEVEL1  
111 VIEW\_LEVEL2  
120 VIEW\_THRU\_CEILING  
121 VIEW\_THRU\_FLOOR  
132 BREAKABLE  
132 BREAKABLE\_SPRITE  
133 QUAKE\_SPOT  
134 SOUND\_SPOT  
135 SLIDE\_SECTOR  
136 CEILING\_FLOOR\_PIC\_OVERRIDE  
140 TRIGGER\_SECTOR  
141 DELETE\_SPRITE  
143 SECT\_ROTATOR

```

144 SECT_ROTATOR_PIVOT
145 SECT_SLIDOR
146 SECT_CHANGOR
147 SO_DRIVABLE_ATTRIB
148 WALL_DONT_STICK
149 SPAWN_ITEMS
150 STOP_SPOUND_SPOT
202 BOUND_FLOOR_BASE_OFFSET
203 BOUND_FLOOR_OFFSET
500-
-596 SECT_SO_BOUNDING
1000 MIRROR_CAM
1002 AMBIENT_SOUND
1005 ECHO_SPOT
1006 DRIP_GENERATOR

```

## 7.1) Meet the ST1 Sprite

Shadow Warrior has one sector effector, the ST1 sprite. Using it, you will be able to create all of the different special effects in the game. This is how you make working doors, dynamic slopes, explosions, moving sectors, room over room, water, and everything else. For this reason, it is by far the most commonly used sprite in Shadow Warrior. Examples of almost all uses of this sprite is given in EXAMPLE.MAP, so be sure to check that map out.

### 7.1.1) Using the ST1 Sprite

An ST1 is a sprite just like any other. You can find it in the tile set (the "V" menu in BUILD) at position 2307. The most common way to use an ST1 is to place it in the sector you wish to be affected. For example, if you want a sector to be a door, you would place the ST1 in that sector, tag it properly, and then that sector would behave like a door.

### 7.1.2) ST1 Functions

There are almost 100 different functions an ST1 can perform. When you you first place an ST1 sprite, its sign in 2D mode will probably read „SECT\_SINK“. This indicates the ST1's hitag is 0, which designates the SECT\_SINK function. By changing the hitag (Alt H in 2D mode, or '+H in 3D mode) of the sprite, you change the function it will perform, and its sign will change to reflect that. All of the different ST1 functions are listed below.

### 7.1.3) ST1 Context Sensitive Help

If, in 2D mode, you point at an ST1 and press F6, context sensitive help will be listed at the bottom of the screen. The information provided will be based on the hitag of the ST1. If, for instance, the ST1's hitag is 92, then F6 will give you help regarding vators. Most of this help will be the same material you find in this document.

## 7.II) The Shadow Warrior Tag System

The Shadow Warrior tagging system was developed by a team which was not in contact with the Duke Nukem 3D team, so if you are familiar with Duke Nukem 3D mapping, then forget everything you know, because it can only confuse you here. Shadow Warrior's tagging system is very powerful, and with this power comes complexity. Stick with it, though, and you'll soon be proficient enough to set up a shootable light or a locked door without even looking at this reference.

### 7.II.1) The 15 Standard Tags

Every sprite in Shadow Warrior makes use of 15 different TAGS. A tag is merely a variable that contains information regarding the sprite. To enter a value for a tag, point to the sprite, then press the QUOTE key plus the number of the tag. To access TAG11 through TAG15, press SHIFT+APOSTROPHE+the number of the tag. Note that TAG1 and TAG2 are the Hitag and Lotag, respectively, and so can also be altered with the Alt+H and the Alt+T keys. The following is a list of the 15 tags, their function, and the range of values they can hold.

Tag No.	Name/Function	Range of Values	Default Value
TAG 1	Hitag	-32768 to 32767	
TAG 2	Lotag	-32768 to 32767	
TAG 3	clipdist	-128 to 127	32
TAG 4	Angle	-32768 to 32767	
TAG 5	xvel	-32768 to 32767	
TAG 6	yvel	-32768 to 32767	
TAG 7	zvel 1	-128 to 127	
TAG 8	zvel 2	-128 to 127	
TAG 9	owner 1	-128 to 127	255 or -1
TAG 10	owner 2	-128 to 127	255 or -1
TAG 11	Shade	-128 to 127	
TAG 12	Palette	-128 to 127	
TAG 13	xoffset and yoffset	-32768 to 32767	
TAG 14	xrepeat and yrepeat	-32768 to 32767	
TAG 15	z	-32768 to 32767	

### 7.II.2) The 11 Boolean Flags

A boolean flag can hold only two possible values: 0 or 1. Every sprite in Shadow Warrior makes use of 11 boolean flags. To enter a value for a boolean flag, press the SEMICOLON key plus the number of the tag. To enter a value for BOOL11, press SHIFT, plus SEMICOLON, plus the „1“ key.

### 7.II.3) The Hitag (TAG1)

An ST1 sprite always uses TAG1 (the Hitag) to specify which function the sprite is to perform on the sector.



#### 7.II.4) The Lotag (TAG2) and Match Tags

An ST1 sprite often uses TAG2 (the Lotag) to specify a so-called „match“ tag. This means that the value in TAG2 (Lotag) is a unique and arbitrary value designated by you, the level designer. This unique value is used to match the ST1's function to a trigger. A trigger can be caused by such events as operating a switch or stepping into a trigger sector. BUILD will report the next available match tag when you press F5 in 2D mode.

#### 7.II.5) The TAG4 Tag (Sprite Angle)

TAG4, as you can see from the heading above, is a sprite's angle. By changing the TAG4 (Angle) value of a sprite, you also change the angle it's facing. Be very cautious of this, because a great many ST1 functions rely on having an exact value in TAG4 (Angle). If you use the „,“ or „.“ (comma or period) keys on that sprite, the value will shift by 128, which is probably a large enough change that the ST1 will no longer behave as you'd expect it to. In the case of ST1 134 (SOUND\_SPOT), the game will probably quit to DOS with an error. Another pitfall to beware is rotating Right+Alt selected groups of sectors. Since all of the sprites within the selected group are rotated with the sectors when you do this, their TAG4 (Angle) values will often become grossly incorrect.

#### 7.II.6) The BOOL1 Tag

If the BOOL1 flag is set to 1 on any sprite, BUILD will cause that individual sprite to behave as though it were an ST1 sprite. This is generally only used to create breakable sprites.

#### 7.II.7) The BOOL11 Tag

BOOL11 is a flag used by all sprites in the game, not just ST1s. If the BOOL11 flag on any sprite is set to 1, then that sprite will remain stationary rather than ride up and down on a vator.

### **7.III) Notes Regarding the ST1 Sprite**

#### 7.III.1) Z-Height

Many functions are affected by the height of the ST1 sprite. For instance, a door might open to the height of the sprite, or a teleporter might move the player to the same height as the sprite. When calculating a height, always use the lowest point of the sprite graphic as the reference, no matter which way the sprite is oriented. A sprite that is placed on the floor by means of Ctrl+PgDn is at the exact same z-coordinate as the floor, no matter how far the graphic sticks up. A sprite that is placed against the ceiling using Ctrl+PgUp, however, does not have the same z-height as the ceiling because the top of the graphic is resting against the ceiling, and not the bottom. You'll find that you

will occasionally need to sink an ST1 into the floor or into the ceiling to get a desired effect.

### 7.III.2) Color Coding the ST1's

The only ST1 functions affected by the sprite's palette value are LIGHTING (hitag 108) and LIGHTING\_DIFFUSE (hitag 109). Since in most cases the ST1 isn't affected by palette, you may find it useful to assign different palette values to certain ST1 functions, so as to more readily identify them in 3D mode. For instance, I sometimes like to color SPAWN\_SPOTS green (palette 11) and TAG\_SPRITE\_HIT\_MATCHES red (palette 14.) Make a habit of color coding your ST1 sprites, and you won't regret it.

### 7.III.3) Brighten the ST1's

No ST1 function is affected by the shade value of the sprite, so why not make them really bright? This will make them easier to spot in 3D editing mode.

## 7.IV) The Shadow Warrior ST1 Function Reference

<b>Hitag 0</b>	<b>SECT_SINK</b>
<p>Lotag = how deep player will sink into floor</p> <p>Place this in water, lava, or any sector with a floor surface you want Lo Wang to sink into. He will sink to the depth specified in Lotag (TAG2). When building dive sectors (see Hitag 7: SECT_DIVE_AREA) Lotag (TAG2) needs to be at least 35. You probably never want to use a value greater than 40, or the player's view may become distorted.</p> <p><i>NOTE:</i> If used with a lava texture, SECT_SINK will cause Lo Wang to take damage when stepping in the sector.</p> <p><i>EXAMPLES:</i> AA and BB in EXAMPLE.MAP</p> <p><i>SEE ALSO:</i> PLAX_GLOB_Z_ADJUST (Hitag 90), CEILING_Z_ADJUST (Hitag 97), FLOOR_Z_ADJUST (Hitag 98)</p>	
<b>Hitag 1</b>	<b>SECT_OPERATIONAL</b>
<p>Place this in a sector of a driveable Sector Object. This will be the sector the player must stand in and press SPACE to operate the driveable.</p> <p><i>EXAMPLES:</i> EE and EF in EXAMPLE.MAP</p>	
<b>Hitag 3</b>	<b>SECT_CURRENT</b>

Lotag	speed of current
Angle	direction of current
<p>If SECT_CURRENT is used in a sector along with either a SECT_SINK (Hitag 0), a SECT_UNDERWATER (Hitag 8), or a SECT_UNDERWATER2 (Hitag 9), the player will be moved in the direction the SECT_CURRENT is facing, at the speed designated in Lotag (TAG2).</p> <p><i>EXAMPLES:</i> AA and BB in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_SINK (Hitag 0), SECT_UNDERWATER (Hitag 8), SECT_FLOOR_PAN (Hitag 19)</p>	

<b>Hitag 5</b>	<b>SECT_NO_RIDE</b>
<p>Use SECT_NO_RIDE in any sector of a moving Sector Object to keep the player from moving with that sector.</p> <p><i>EXAMPLES:</i> EC, FC and FD in EXAMPLE.MAP</p>	

<b>Hitag 7</b>	<b>SECT_DIVE_AREA</b>
<p>Lotag = match tag</p> <p>Diving area - pressing the CROUCH key when on the sector with a SECT_DIVE_AREA ST1 will teleport the player to an UNDERWATER (Hitag 8) area with a matching UNIQUE match tag. You must also have a SECT_SINK (Hitag 0) sprite in the same sector with a Lotag (TAG2) of at least 35 for this to work.</p> <p><i>EXAMPLES:</i> BA in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_SINK (Hitag 0), SECT_UNDERWATER (Hitag 8)</p>	

<b>Hitag 8</b>	<b>SECT_UNDERWATER</b>
<p>Lotag = match tag</p> <p>Underwater area - floating to the ceiling of this area will teleport the player to the DIVE_AREA (Hitag 7) with a matching UNIQUE non zero match tag.</p> <p><i>IMPORTANT:</i> The over- and underwater sectors need to have congruent shapes, and the SECT_DIVE_AREA sprite and the SECT_UNDERWATER sprite need to be in the same relative position for both of their respective sectors.</p> <p><i>IMPORTANT:</i> A zero in Lotag (TAG2) denotes that this is a swim area with no matching SECT_DIVE_AREA. This can be used for areas where the player cannot surface.</p> <p><i>EXAMPLES:</i> BA in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_DIVE_AREA (Hitag 7), SECT_UNDERWATER2 (Hitag 9)</p>	

<b>Hitag 9</b>	<b>SECT_UNDERWATER2</b>
----------------	-------------------------

Lotag	match tag
TAG3	surface type of sector: <b>0/32</b> = can surface here <b>1</b> = can not surface here

Secondary Underwater area - floating to the surface of this area will teleport you to the SECT\_DIVE\_AREA (Hitag 7) with a matching unique match tag. SECT\_UNDERWATER2 allows you to split an underwater area into multiple sectors, while still having a single large DIVE\_AREA sector.

*EXAMPLES:* BA in EXAMPLE.MAP  
*SEE ALSO:* SECT\_DIVE\_AREA (Hitag 7), SECT\_UNDERWATER (Hitag 8)

<b>Hitag 16</b>	<b>SO_ANGLE</b>
-----------------	-----------------

Angle = angle of Sector Object
--------------------------------

Place this anywhere in a Sector Object to tell Shadow Warrior which end is the front. This will affect the orientation of a Sector Object as it follows a track or is operated by the player.

*EXAMPLES:* EE, EF, and FA in EXAMPLE.MAP

<b>Hitag 19</b>	<b>SECT_FLOOR_PAN</b>
-----------------	-----------------------

Lotag	pan speed (0 - 16384)
Angle	direction of panning
BOOL1	move with speed of Sector Object if set to „1“

This causes the floor of the sector it is in to pan in the direction indicated by Angle (TAG4) at the rate indicated by Lotag (TAG2). Set BOOL1 to „1“ if the sector is part of a Sector Object and you want the floor to pan at the same speed the Sector Object is moving. The BOOL1 feature is primarily used to simulate moving tank treads on driveable Sector Objects.

*EXAMPLES:* BB in EXAMPLE.MAP  
*SEE ALSO:* SECT\_CEILING\_PAN (Hitag 21), WALL\_PAN\_SPEED (Hitag 23)

<b>Hitag 21</b>	<b>SECT_CEILING_PAN</b>
-----------------	-------------------------

Lotag	pan speed (0 - 16384)
Angle	direction of panning
BOOL1	move with speed of Sector Object if set to „1“

This causes the ceiling of the sector it is in to pan in the direction indicated by Angle (TAG4) at the rate indicated by Lotag (TAG2). Set BOOL1 to „1“ if the sector is part of a Sector Object and you want the ceiling to pan at the same speed the Sector Object is moving.

*EXAMPLES:* BB in EXAMPLE.MAP  
*SEE ALSO:* SECT\_FLOOR\_PAN (Hitag 19), WALL\_PAN\_SPEED (Hitag 23)

<b>Hitag 23</b>	<b>WALL_PAN_SPEED</b>
-----------------	-----------------------

Lotag	pan speed (0 - 16384)
Angle	point sprite at desired wall
TAG6	direction of panning
BOOL1	move with speed of Sector Object

Point the sprite at the wall you want it to affect. That wall's texture will pan in the direction indicated by TAG6 at the rate indicated by Lotag (TAG2). Set BOOL1 to „1“ if the wall is part of a Sector Object and you want the wall texture to pan at the same speed the Sector Object is moving. This feature is usually used to simulate tank treads on driveable Sector Objects.

*EXAMPLES:* BB in EXAMPLE.MAP

*SEE ALSO:* SECT\_CEILING\_PAN (Hitag 21), SECT\_FLOOR\_PAN (Hitag 19)

<b>Hitag 29</b>	<b>SECT_LOCK_DOOR</b>
-----------------	-----------------------

Lotag	type of key required to unlock door: 1 = RED Keycard 2 = BLUE Keycard 3 = GREEN Keycard 4 = YELLOW Keycard 5 = GOLD Master Key 6 = SILVER Master Key 7 = BRONZE Master Key 8 = RED Master Key
-------	---

Locks a door. The player must have the correct key to open it. Works for rotating, sliding and all vator uses, including switches. Place it in the same sector with a SECT\_VATOR (Hitag 92), SECT\_ROTATOR (Hitag 143), or SECT\_SLIDOR (Hitag 145).

*EXAMPLES:* EB in EXAMPLE.MAP

<b>Hitag 30</b>	<b>SPRI_CLIMB_MARKER</b>
-----------------	--------------------------

Angle = point away from wall
------------------------------

Center this sprite on the ladder and set the angle pointing out away from the wall. The position of the sprite is important, because that is the (x,y) location the player sprite will be in while climbing the ladder. Used in conjunction with TAG\_WALL\_CLIMB (Walltag 305).

*EXAMPLES:* BC and DF in EXAMPLE.MAP

<b>Hitag 31</b>	<b>SECT_SO_SPRITE_OBJ</b>
	Place this ST1 anywhere in a Sector Object. Makes a Sector Object become a „Sprite Object“. The sprites will move but the sectors will not. Sprite Objects can cross sector boundaries, whereas regular Sector Objects cannot.  <i>EXAMPLES:</i> EC in EXAMPLE.MAP
<b>Hitag 32</b>	<b>SECT_SO_DONT_BOB</b>
	Mark a sector in a bobbing Sector Object so it will not bob with the rest of the sector.  <i>EXAMPLES:</i> Look at the boat in \$WHIRL.MAP for an example. <i>SEE ALSO:</i> SO_BOB_START (Hitag 52), SO_BOB_SPEED (Hitag 53)
<b>Hitag 33</b>	<b>SECT_SO_SINK_DEST</b>
	Lotag = offset for sinking Mark a sector in a Sector Object as the destination for sinking. It will not sink with other sectors. Lotag (TAG2) is the offset from the sector's current floor height to use as the sinking destination.  <i>EXAMPLES:</i> Look at the boat in \$WHIRL.MAP for an example. <i>SEE ALSO:</i> SECT_SO_DONT_SINK (Hitag 34)
<b>Hitag 34</b>	<b>SECT_SO_DONT_SINK</b>
	Mark a sector in a Sector Object so it will not sink with other sectors.  <i>EXAMPLES:</i> Look at the boat in \$WHIRL.MAP for an example. <i>SEE ALSO:</i> SECT_SO_SINK_DEST (Hitag 33)
<b>Hitag 37</b>	<b>SECT_SO_FORM_WHIRLPOOL</b>
	Lotag = depth of whirlpool Lower the whirlpool sectors to a 'depth' level when the appropriate track sprite is reached by the Sector Object (TRACK_SO_FORM_WHIRLPOOL - Tracktag 724).  <i>EXAMPLES:</i> Look at the boat in \$WHIRL.MAP for an example.
<b>Hitag 38</b>	<b>SECT_ACTOR_BLOCK</b>
	Mark a sector to keep enemy actors from entering, but the player can still enter. Both enemies and the player can still fire weapons through the sector.  <i>EXAMPLES:</i> BC in EXAMPLE.MAP
<b>Hitag 42</b>	<b>MULTI_PLAYER_START</b>

Lotag	player number (0 - 7)
Angle	direction the player is facing at start
<p>These mark the positions where players will spawn in WangBang games. The players' feet will be placed at the same height as the bottom of the ST1 sprite, so place them on the floor unless you want the player to fall. Always be sure to include start point number 0 (zero). Also, for multiplayer levels to work correctly you have to place all 8 starting positions or else the game might go out of sync.</p> <p><i>EXAMPLES:</i> Any game map  <i>SEE ALSO:</i> MULTI_COOPERATIVE_START (Hitag 48)</p>	

<b>Hitag 43</b>	<b>FIREBALL_TRAP</b>
-----------------	----------------------

Lotag	match tag
Angle	direction the fireball is going to fly
<p>Shoots a fireball from the location of this sprite in the direction the angle is pointing when a sector tagged with TAG_TRIGGER_MISSILE_TRAP (Sector Tag 43) is stepped on.</p> <p><i>EXAMPLES:</i> BD in EXAMPLE.MAP  <i>SEE ALSO:</i> BOLT_TRAP (Hitag 44), SPEAR_TRAP (Hitag 63)</p>	

<b>Hitag 44</b>	<b>BOLT_TRAP</b>
-----------------	------------------

Lotag	match tag
Angle	direction the rocket is going to fly
<p>Shoots a rocket from the location of this sprite in the direction the angle is pointing when a sector tagged with TAG_TRIGGER_MISSILE_TRAP (Sector Tag 43) is stepped on.</p> <p><i>EXAMPLES:</i> BD in EXAMPLE.MAP  <i>SEE ALSO:</i> FIREBALL_TRAP (Hitag 43), SPEAR_TRAP (Hitag 63)</p>	

<b>Hitag 45</b>	<b>SECT_SO_DONT_ROTATE</b>
-----------------	----------------------------

<p>Causes a Sector Object to move without rotating as it follows its track or is driven by the player. Place in the center sector.</p> <p><i>EXAMPLES:</i> FB in EXAMPLE.MAP  <i>SEE ALSO:</i> SO_TURN_SPEED (Hitag 55), SO_LIMIT_TURN (Hitag 68)</p>	
---	--

<b>Hitag 46</b>	<b>PARALLAX_LEVEL</b>
-----------------	-----------------------

<p>Lotag = type of the parallaxed sky (0, 1 or 2)</p> <p>Set parallax sky type to 0, 1, or 2. This value adjusts the xrepeat of the sky texture. You only need to place one per level in a single sector that uses a parallax sky texture, and it will affect the whole level. If more than one is placed in a map, the sky will usually behave strangely.</p> <p><i>EXAMPLES:</i> BE in EXAMPLE.MAP</p>	
--	--

<b>Hitag 47</b>	<b>SECT_SO_DONT_COPY_PALETTE</b>
<p>Normally any sprite or the player standing on a floor with a palette number higher than 0 (default) take on the palette of the floor. Use this ST1 on sectors where you don't want this to happen.</p> <p><i>EXAMPLES:</i> BE in EXAMPLE.MAP</p>	

<b>Hitag 48</b>	<b>MULTI_COOPERATIVE_START</b>
Lotag	player number (0 - 7)
Angle	direction the player is facing at start
<p>Spawn spots for cooperative multi-player games. The players' feet will be placed at the same height as the bottom of the ST1 sprite, so place them on the floor unless you want the player to fall. Always be sure to include start point number 0 (zero). Also, for multiplayer levels to work correctly you have to place all 8 starting positions or else the game might go out of sync.</p> <p><i>EXAMPLES:</i> Any game map  <i>SEE ALSO:</i> MULTI_PLAYER_START (Hitag 42)</p>	

<b>Hitag 49</b>	<b>SO_SET_SPEED</b>
<p>Lotag = speed of the Sector Object (2 - 128)</p> <p>Sets the speed at which a Sector Object follows a track. This is functionally the same as the TRACK_SET_SPEED track sprite (Tracktag 702).</p> <p><i>EXAMPLES:</i> FA in EXAMPLE.MAP</p>	

<b>Hitag 50</b>	<b>SO_SPIN</b>
<p>Lotag = spin speed of the Sector Object</p> <p>Causes a Sector Object to spin clockwise at the rate specified in Lotag (TAG2). This is functionally the same as the TRACK_SPIN track sprite (Tracktag 715).</p> <p><i>EXAMPLES:</i> EC in EXAMPLE.MAP  <i>SEE ALSO:</i> SO_SPIN_REVERSE (Hitag 51)</p>	

<b>Hitag 51</b>	<b>SO_SPIN_REVERSE</b>
<p>Lotag = spin speed of the Sector Object</p> <p>Causes a Sector Object to spin counter-clockwise at the rate specified in Lotag (TAG2). This is functionally the same as the TRACK_SPIN_REVERSE track sprite (Tracktag 720).</p> <p><i>EXAMPLES:</i> EC in EXAMPLE.MAP  <i>SEE ALSO:</i> SO_SPIN (Hitag 50)</p>	



<b>Hitag 52</b>	<b>SO_BOB_START</b>
-----------------	---------------------

<p>Lotag = amount of bobbing</p> <p>Causes a Sector Object to bob up and down continuously. Lotag (TAG2) specifies how high the object will bob. This is functionally the same as the TRACK_BOB_START track sprite (Tracktag 717).</p> <p><i>EXAMPLES:</i> \$WHIRL.MAP  <i>SEE ALSO:</i> SO_BOB_SPEED (Hitag 53), SECT_SO_DONT_BOB (Hitag 32)</p>
---

<b>Hitag 53</b>	<b>SO_BOB_SPEED</b>
-----------------	---------------------

<p>Lotag = speed of bobbing</p> <p>Place this in the same sector with a SO_BOB_START ST1 sprite (Hitag 52). Lotag (TAG2) specifies the speed at which the Sector Object will bob. This is functionally the same as the TRACK_BOB_SPEED track sprite (Tracktag 718).</p> <p><i>EXAMPLES:</i> \$WHIRL.MAP  <i>SEE ALSO:</i> SO_BOB_START (Hitag 52), SECT_SO_DONT_BOB (Hitag 32)</p>
--

<b>Hitag 55</b>	<b>SO_TURN_SPEED</b>
-----------------	----------------------

<p>Lotag = turn speed (0 is fastest)</p> <p>Determines how fast a Sector Object turns to meet the new angle as it follows a track. The default is 4. For some reason this has no track tag functional equivalent.</p> <p><i>EXAMPLES:</i> FA in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_SO_DONT_ROTATE (Hitag 45), SO_LIMIT_TURN (Hitag 68)</p>
--

<b>Hitag 56</b>	<b>LAVA_ERUPT</b>
-----------------	-------------------

Lotag	amount of erupting lava debris (1 - 1024)
TAG3	type of eruption: 0 = lava shards 1 = volcano boulders
Angle	direction of eruption
TAG5	angle delta from Angle
TAG6	maximum x/y velocity
TAG7	Z (up/down) velocity amount
TAG8	Z (up/down) velocity random amount - added to TAG5 value
TAG9	erupt for this many seconds - default 10
TAG10	wait for this many seconds till next eruption
BOOL1	if „1“ then don't spawn unless player is in TRIGGER_SECTOR
<p>The LAVA_ERUPT ST1 sprite will spawn lava debris at its location. The lava will constantly erupt in intervals (thus killing framerate) unless BOOL1 is set to „1“, in which case you will need to set up trigger sectors. See TRIGGER_SECTOR (Hitag 140). In that case the lava will only erupt while the player is standing in those sectors.</p> <p><i>EXAMPLES:</i> BF in EXAMPLE.MAP</p>	

<b>Hitag 57</b>	<b>SECT_EXPLODING_CEIL_FLOOR</b>
-----------------	----------------------------------

<p>Lotag = match tag</p> <p>Pulls the ceiling and floor together when the map is loaded. The ceiling and floor can then be explosively blown apart by shooting a TAG_SPRITE_HIT_MATCH sprite (Lotag 257), otherwise known as a crack sprite. The TAG_SPRITE_HIT_MATCH Hitag must match Lotag (TAG2) of the SECT_EXPLODING_CEIL_FLOOR ST1 sprite.</p> <p><i>EXAMPLES:</i> CA in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_COPY_DEST (Hitag 58), SECT_COPY_SOURCE (Hitag 59)</p>
---

<b>Hitag 58</b>	<b>SECT_COPY_DEST</b>
-----------------	-----------------------

Lotag	match tag for triggering event
TAG3	match tag for sector source-to-destination copying

Effectively copies all of the source sector attributes to the destination sector when triggered. This is probably the most powerful feature of Shadow Warrior, because it lets you do so many cool things to a map. Examples are scattered throughout almost every map. You can do things like blow a hole in the floor and reveal a hidden swimmable stream, lava pool, or gettable item. It will move sprites from the source sector to the destination also. It's very important that the source and destination sectors are congruent. Lotag (TAG2) is an event match (i.e. hit a switch and the copy occurs), while TAG3 serves to identify which source sector belongs to which destination sector when copying multiple sectors at the same time.

Copy sector 58/59 tags will activate in conjunction with matching SECT\_MOVE\_WALL (Hitag 103) Wall tags *AFTER* the copy is done.

Copy sector 58/59 tags should remove any visible sprites from the destination sector, although this doesn't apply to gettable items.

Copy sector 58/59 tags will activate in conjunction with matching SPAWN\_SPOT tags *AFTER* the copy is done.

*EXAMPLES:* CB in EXAMPLE.MAP  
*SEE ALSO:* SECT\_COPY\_SOURCE (Hitag 59), SECT\_EXPLODING\_CEIL\_FLOOR (Hitag 57)

<b>Hitag 59</b>	<b>SECT_COPY_SOURCE</b>
-----------------	-------------------------

Lotag	match tag for triggering event
TAG3	match tag for sector source-to-destination copying

This is the source from which a destination sector is copied. See SECT\_COPY\_DEST (Hitag 58) for more information.

*EXAMPLES:* CB in EXAMPLE.MAP  
*SEE ALSO:* SECT\_COPY\_DEST (Hitag 58), SECT\_EXPLODING\_CEIL\_FLOOR (Hitag 57)

<b>Hitag 62</b>	<b>SHOOT_POINT</b>
-----------------	--------------------

Lotag	match tag; if non-zero it will call DoMatchEverything every time Sector Object is fired; can be a one time operationg with BOOL1
TAG3	type of ammo: -1/255 = „do nothing“ shoot point, primarily to make the player's weapon go away when using the Sector Object <b>0</b> = Tank Shell <b>1</b> = Machine Gun <b>2</b> = Laser Gun <b>3</b> = Rail Gun <b>4</b> = Fireball <b>5</b> = Rocket <b>6</b> = Micro Missile Battery
Angle	shot direction
TAG5	delay until next fire in 120 <sup>th</sup> of a second tics (60 = half a second); defaults to 80 tics, except ammo type 1 which defaults to 6 tics
TAG6	SOUND_SPOT (Hitag 134) match tag; plays the sound defined by the matching SOUND_SPOT ST1 sprite
BOOL1	if „1“, Lotag will be reset to 0 (zero) after first use
<p>For operational Sector Objects and Auto Turrets. Sets the point from which the Sector Object shoots. Place on the end of a gun barrel and point the sprite in the desired shooting direction. When a shoot point exists on an operational Sector Object, the player's weapon is no longer used, otherwise the player's weapon stays up. Sometimes TAG6 is not needed because some ammo types have default sounds.</p> <p><i>EXAMPLES:</i> EE and EF in EXAMPLE.MAP</p>	

<b>Hitag 63</b>	<b>SPEAR_TRAP</b>
-----------------	-------------------

Lotag	match tag
Angle	direction the arrow is going to fly
<p>Shoots an arrow from the location of this sprite in the direction the angle is pointing when a sector tagged with TAG_TRIGGER_MISSILE_TRAP (Sector Tag 43) is stepped on.</p> <p><i>EXAMPLES:</i> BD in EXAMPLE.MAP  <i>SEE ALSO:</i> FIREBALL_TRAP (Hitag 43), BOLT_TRAP (Hitag 44)</p>	

<b>Hitag 64</b>	<b>SO_SYNC1</b>
-----------------	-----------------

<p>Place this in a Sector Object. Keeps Sector Objects running at a constant rate so they can stay in sync better, if needed. Possibly useful for meshing gears or something. If it's rideable use SYNC1. Only use this when you absolutely need to.</p> <p><i>NOTE:</i> This isn't used anywhere in Shadow Warrior, and probably isn't even necessary.  <i>SEE ALSO:</i> SO_SYNC2 (Hitag 65)</p>	
---	--

<b>Hitag 65</b>	<b>SO_SYNC2</b>
-----------------	-----------------

<p>Same as SO_SYNC1 (Hitag 64), only slightly faster. If it's rideable use SYNC1. Only use this when you absolutely need to.</p>	
--	--

*NOTE:* This isn't used anywhere in Shadow Warrior, and probably isn't even necessary.  
*SEE ALSO:* SO\_SYNC1 (Hitag 64)

**Hitag 66**    **DEMO\_CAMERA**

Lotag	delta angle for rotation
TAG3	camera type (0 or 1)
Angle	direction the camera will point at

A third person camera view which is used when playing back demos, or when playing with the „-cam“ command line option. Point it in the direction you want the camera to face. Lotag (TAG2) is the amount the camera will turn in either direction (512 = 90 degrees in both directions). If TAG3 is 1, then a special camera will be used which orbits around the player at a constant distance, speed, and height, until the player moves out of the camera's view.

*EXAMPLES:* FD in EXAMPLE.MAP  
*SEE ALSO:* MIRROR\_CAM (Hitag 1000)

**Hitag 68**    **SO\_LIMIT\_TURN**

Lotag = delta angle for turning

Limits the turning of a Sector Object. Especially useful with turrets. Point the SO\_LIMIT\_TURN sprite in the center direction. Lotag (TAG2) is the amount the Sector Object can pivot in either direction (512 = 90 degrees either way, for a 180 degree total range of motion).

*EXAMPLES:* EE in EXAMPLE.MAP  
*SEE ALSO:* SECT\_SO\_DONT\_ROTATE (Hitag 45), SO\_TURN\_SPEED (Hitag 55)

**Hitag 69**    **SPAWN\_SPOT**

Lotag	match tag
TAG3	type of spawned item/debris: <b>0</b> = nothing <b>1</b> = breaking glass <b>2</b> = smashed wood <b>3</b> = smoke, only used for damaged Sector Objects <b>4</b> = rice paper (silent) <b>5</b> = blood splat (silent) <b>6</b> = standard explosion <b>7</b> = explosion with phosphorous <b>8</b> = breaking metal <b>9</b> = crumbling stone <b>11</b> = gib splat <b>12</b> = smashed wood <b>13</b> = crumbling stone
Angle	direction debris flies out
TAG5	time delay before activating, in 1/8 second tics (8 tics = 1 sec.)
TAG6	always set this equal to TAG3!
TAG8	amount of shrap pieces (if 0 uses default amount)
TAG9	random Z (up/down) range in pixels from ST1 Z position
TAG10	increase size of shrap (0 - 255)
<p>Spawn some effect, triggered by just about anything. This is used a lot throughout the game for various special effects, including crumbling ceilings, large complex explosions and breakable sprites.</p> <p><i>EXAMPLES:</i> CC in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_EXPLODING_CEIL_FLOOR (Hitag 57), SPAWN_ITEMS (Hitag 149)</p>	

<b>Hitag 70</b>	<b>SO_MATCH_EVENT</b>
-----------------	-----------------------

Lotag	match tag
TAG3	spin speed adjustment, can be negative or positive
TAG7	velocity adjustment
BOOL1	velocity adjustment type: <b>0</b> = velocity adjustment is instant <b>1</b> = velocity adjustment is approached
BOOL2	direction flip: <b>0</b> = false <b>1</b> = true
BOOL3	toggle help, only used for triggers and things that don't have a definite ON/OFF <b>0</b> = off <b>1</b> = on

Certain Sector Object effects, like SO\_SCALE\_INFO (Hitag 100) and SO\_KILLABLE (Hitag 80) can be triggered by a match tag event. This is accomplished using the SO\_MATCH\_EVENT ST1 sprite. The adjustment values are added when the event is ON and they are subtracted when the event is OFF. You may need to experiment with this a bit, since we didn't use it very much in the game.

*EXAMPLES:* EE in EXAMPLE.MAP

<b>Hitag 71</b>	<b>SO_SLOPE_FLOOR_TO_POINT</b>
	Aligns a sloping floor sector in a Sector Object to a point tagged as TAG_WALL_ALIGN_SLOPE_TO_POINT (Walltag 550). Generally used with a spike or drill effect rising up from the floor.  <i>SEE ALSO:</i> SO_SLOPE_CEILING_TO_POINT (Hitag 72).

<b>Hitag 72</b>	<b>SO_SLOPE_CEILING_TO_POINT</b>
	Aligns a sloping floor sector in a Sector Object to a point tagged as TAG_WALL_ALIGN_SLOPE_TO_POINT (walltag 550). Generally used with a tornado or whirlpool effect coming down from the ceiling.  <i>EXAMPLES:</i> ED in EXAMPLE.MAP <i>SEE ALSO:</i> SO_SLOPE_FLOOR_TO_POINT (Hitag 71).

<b>Hitag 73</b>	<b>SO_TORNADO</b>
	This sets up a Sector Object to behave like a tornado.  <i>EXAMPLES:</i> ED in EXAMPLE.MAP <i>SEE ALSO:</i> SO_AMOEBA (Hitag 75), SO_AUTO_TURRET (Hitag 81)

<b>Hitag 75</b>	<b>SO_AMOEBA</b>
	This sets up a Sector Object to behave like an amoeba.  <i>EXAMPLES:</i> EC in EXAMPLE.MAP <i>SEE ALSO:</i> SO_TORNADO (Hitag 73), SO_AUTO_TURRET (Hitag 81), SO_SCALE_INFO (Hitag 100)

<b>Hitag 76</b>	<b>SO_MAX_DAMAGE</b>
-----------------	----------------------

Lotag	maximum amount of damage a Sector Object can take before breaking (= health of the SO)
TAG3	always set to 0
TAG5	initial damage setting (-1 is broken); if this value is 0 the SO health is set to the value in Lotag (= full health)

Damage for operational and auto turret Sector Objects. Sector Objects only take damage from explosive type weapons, not machine gun types. Be sure to add a SPAWN\_SPOT (Hitag 69) with TAG3 set to 3 for the smoke effect. When the driveable is broken, it will trigger the smoke, and when the object is repaired, the smoke will go away again.

*EXAMPLES:* EE in EXAMPLE.MAP  
*SEE ALSO:* SO\_KILLABLE (Hitag 80)

**Hitag 77**    **SO\_RAM\_DAMAGE**

Lotag = amount of damage

Damage amount done to objects by ramming with an operational Sector Object. Not needed for running over actors, since this is done by default. This is usually used for ramming killable Sector Objects or other operational Sector Objects.

**Hitag 80**    **SO\_KILLABLE**

Marks a Sector Object as killable. A Sector Object can be killed either when it receives enough damage, set by SO\_MAX\_DAMAGE (Hitag 76), or when a match event occurs, set by SO\_MATCH\_EVENT (Hitag 70). A killable Sector Object is actually completely deleted by becoming part of the sector surrounding the Sector Object.

*EXAMPLES:* EE in EXAMPLE.MAP

**Hitag 81**    **SO\_AUTO\_TURRET**

A Sector Object with a SO\_AUTO\_TURRET ST1 will always turn to face the player. Useful for a variety of effects, since it won't shoot the player unless you give it a SHOOT\_POINT (Hitag 62). A SHOOT\_POINT on an auto turret will only fire the machine gun ammo type. The Sector Object center must have a sector Hitag of 96 or 98 for this to work properly.

*EXAMPLES:* EE in EXAMPLE.MAP

*SEE ALSO:* SO\_AMOEBA (Hitag 75), SO\_TORNADO (Hitag 73)

**Hitag 82**    **SECT\_DAMAGE**

Lotag	damage done per second
BOOL1	set this to:
	<ul style="list-style-type: none"> <li>0 - to do damage when touching the floor of the sector</li> <li>1 - to do damage above the sector, like swimming in lava</li> </ul>

Causes a sector to do damage to players and actors.

*EXAMPLES:* CD in EXAMPLE.MAP

**Hitag 84**    **WARP\_TELEPORTER**

Lotag	match tag
TAG3	type of the teleporter: <ul style="list-style-type: none"> <li>0 = always teleports you to the matching sprite location at the angle the sprite is facing (standard teleport)</li> <li>1 = always teleports you by the offset, does not affect the player's angle (silent teleport)</li> </ul>
Angle	direction the player will face
TAG5	random match location in addition to Lotag
TAG6	random match location in addition to Lotag
TAG7	random match location in addition to Lotag
TAG8	random match location in addition to Lotag
TAG10	set this to: <ul style="list-style-type: none"> <li>0 = to teleport when player enters this sector (two way teleporter)</li> <li>1 = to not teleport when player enters this sector (one way teleporter)</li> </ul>

Magic teleporter warps just about anything from one sector to another. If TAG3 is to „0“, then a magic shimmering effect and sound will occur. It is very important that the destination sector be at least as large as the source sector, or the game may quit with an error. Set TAG3 to „1“ to achieve a silent teleporter, used to simulate falling through holes to another story. The player was actually teleported to another place in the map when he fell through the hole. Make sure to build the holes in the floor of the one room and the ceiling of the other room with congruent shapes.

*EXAMPLES:* CD in EXAMPLE.MAP

*SEE ALSO:* WARP\_CEILING\_PLANE (Hitag 85), WARP\_FLOOR\_PLANE (Hitag 86), WARP\_COPY\_SPRITE1 (Hitag 87), WARP\_COPY\_SPRITE2 (Hitag 88)

### Hitag 85 WARP\_CEILING\_PLANE

Lotag = match tag

If the player's view (or a NPC Z coordinate) goes *ABOVE* the Z position of this ST1 sprite then a warp to the Z coordinate of the matching WARP\_FLOOR\_PLANE (Hitag 86) will occur.

*EXAMPLES:* CE in EXAMPLE.MAP

*SEE ALSO:* WARP\_TELEPORTER (Hitag 84), WARP\_FLOOR\_PLANE (Hitag 86), WARP\_COPY\_SPRITE1 (Hitag 87), WARP\_COPY\_SPRITE2 (Hitag 88)

### Hitag 86 WARP\_FLOOR\_PLANE

Lotag = match tag

If the player's view (or a NPC Z coordinate) goes *BELOW* the Z position of this ST1 sprite then a warp to the Z coordinate of the matching WARP\_CEILING\_PLANE (Hitag 85) will occur.

*EXAMPLES:* CE in EXAMPLE.MAP

*SEE ALSO:* WARP\_CEILING\_PLANE (Hitag 85), WARP\_TELEPORTER (Hitag 84), WARP\_COPY\_SPRITE1 (Hitag 87), WARP\_COPY\_SPRITE2 (Hitag 88)



<b>Hitag 87</b>	<b>WARP_COPY_SPRITE1</b>
-----------------	--------------------------

Lotag = match with WARP_COPY_SPRITE2 tag
Any sprite that enters a sector marked with WARP_COPY_SPRITE1 will cause a duplicate image to appear in the sector marked with a WARP_COPY_SPRITE2 (Hitag 88) with a matching Lotag (TAG2).
<i>EXAMPLES:</i> CE in EXAMPLE.MAP
<i>SEE ALSO:</i> WARP_CEILING_PLANE (Hitag 85), WARP_FLOOR_PLANE (Hitag 86), WARP_TELEPORTER (Hitag 84), WARP_COPY_SPRITE2 (Hitag 88)

<b>Hitag 88</b>	<b>WARP_COPY_SPRITE2</b>
-----------------	--------------------------

Lotag = match with WARP_COPY_SPRITE1 tag
Any sprite that enters a sector marked with WARP_COPY_SPRITE1 will cause a duplicate image to appear in the sector marked with a WARP_COPY_SPRITE2 (Hitag 87) with a matching Lotag (TAG2).
<i>EXAMPLES:</i> CE in EXAMPLE.MAP
<i>SEE ALSO:</i> WARP_CEILING_PLANE (Hitag 85), WARP_FLOOR_PLANE (Hitag 86), WARP_COPY_SPRITE1 (Hitag 87), WARP_TELEPORTER (Hitag 84)

<b>Hitag 90</b>	<b>PLAX_GLOB_Z_ADJUST</b>
-----------------	---------------------------

Lotag	ceiling Z (up/down) movement in pixels
TAG3	floor Z (up/down) movement in pixels
Global parallax ceiling and floor adjustment. Default is 500 pixels. This affects how far past the actual surface height a sprite can move in sectors with parallax ceilings and floors.	
<i>SEE ALSO:</i> FLOOR_Z_ADJUST (Hitag 98), CEILING_Z_ADJUST (Hitag 97), SECT_SINK (Hitag 0)	

<b>Hitag 92</b>	<b>SECT_VATOR</b>
-----------------	-------------------

Lotag	match tag
TAG3	<p>type of the vator:</p> <p><i>ALL</i>      <b>0</b> = manual/switch/trigger - primarily used for a basic manually operated door. If a manual type has a match tag that is not 0 then vators with the same match type will be operated on.      Manual operation - works from either within the sector or alongside of the sector. Switch operation - switch resets when the vator goes to default position. Trigger operation - same as switch but it's a floor trigger.</p> <p><b>1</b> = Switch/Trigger only - cannot be operated manually. <i>ALL</i> vator types with the same matching tag get operated on</p> <p><b>2</b> = automatic - starts active</p> <p><b>3</b> = automatic - starts inactive, must be turned on with a switch/trigger</p>
Angle	speed of the vator (usually 100 - 200)
TAG5	acceleration at which door approaches maximum speed
TAG6	trigger this match tag at destination - when a vator reaches its destination it will trigger any event with this match tag
TAG9	auto return time (in 1/8 <sup>th</sup> of a second tics, 8 tics = 1 sec.); if non-zero the vator will return to default position after allotted time - if 0 (zero) it will stay in the on position until operated again
BOOL1	<p>if set to:</p> <p><b>0</b> = vator starts in the OFF position</p> <p><b>1</b> = vator starts in the ON position</p>
BOOL2	<p>if set to:</p> <p><b>0</b> = vator can be toggled</p> <p><b>1</b> = vator can only be operated once</p>
BOOL3	<p>if set to:</p> <p><b>0</b> = vator can crush player</p> <p><b>1</b> = vator can not crush player</p>
BOOL4	<p>if set to:</p> <p><b>0</b> = vator can be operated in WangBang games (Deathmatch)</p> <p><b>1</b> = vator can not be operated in WangBang games (Deathmatch)</p>
BOOL5	<p>if set to:</p> <p><b>0</b> = vator won't call TAG6 match when moving to OFF position</p> <p><b>1</b> = vator calls TAG6 match when moving to ON or OFF position</p>
BOOL6	<p>if set to:</p> <p><b>0</b> = vator waits for motion to complete before it allows retriggering</p> <p><b>1</b> = vator can be operated while it is moving</p>
BOOL7	<p>if set to:</p> <p><b>0</b> = vator can be operated by actors</p> <p><b>1</b> = vator can not be operated by actors</p>
BOOL11	<p>if set to:</p> <p><b>1</b> = displays „Single Play only“ message when operated in WangBang mode, use this together with BOOL4 = 1</p>
<p>Elevators, stompers, doors, lifts and other tricks are collectively called „vators“. The „off“ position is the position of the sector in BUILD, the „on“ position is the height of the ST1 sprite. Flip the ST1 sprite upside down (by pressing „F“ twice on it) to do a ceiling vator like a door or stomper, otherwise the floor will move. Any vator can be locked by placing a SECT_LOCK_DOOR (Hitag 29) in the same sector. Two SECT_VATOR sprites can be placed in a sector to operate the ceiling and floor separately.</p> <p><i>EXAMPLES:</i> DB in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_ROTATOR (Hitag 143), SECT_SLIDOR (Hitag 145), SECT_SPIKE (Hitag 106)</p>	
<b>Hitag 97</b>	<b>CEILING_Z_ADJUST</b>

**Lotag = adjustment amount**

Allows the player and actors to 'sink' into the ceiling of a sector by the amount in Lotag (TAG2). The player view will probably be strange unless the ceiling texture is a parallax sky.

*EXAMPLES:* \$VOLCANO.MAP  
*SEE ALSO:* PLAX\_GLOB\_Z\_ADJUST (Hitag 90), FLOOR\_Z\_ADJUST (Hitag 98), SECT\_SINK (Hitag 0)

**Hitag 98 FLOOR\_Z\_ADJUST**

**Lotag = adjustment amount**

Allows the player and actors to sink into the floor of a sector by the amount in Lotag (TAG2).

*EXAMPLES:* AB and CE in EXAMPLE.MAP  
*SEE ALSO:* PLAX\_GLOB\_Z\_ADJUST (Hitag 90), CEILING\_Z\_ADJUST (Hitag 97), SECT\_SINK (Hitag 0)

**Hitag 99 FLOOR\_SLOPE\_DONT\_DRAW**

Use this ST1 in any sector with a sloped floor and the slope angle will be set to 0 when the scene is drawn. The sector will still act like a slope for all movement purposes.

*EXAMPLES:* CE in EXAMPLE.MAP

**Hitag 100 SO\_SCALE\_INFO**

<b>Lotag</b>	<b>Sector Object scale speed</b>
<b>TAG3</b>	start position: <b>0</b> = start at minimum <b>1</b> = start at maximum
<b>Angle</b>	initial type: <b>0</b> = none <b>1</b> = hold <b>2</b> = destination <b>3</b> = random <b>4</b> = cycle <b>5</b> = random point (needs SO_SCALE_POINT_INFO - Hitag 101)
<b>TAG5</b>	minimum distance - distance from default size to scale inward
<b>TAG6</b>	maximum distance - distance from default size to scale outward
<b>TAG7</b>	activate type - same types as Angle (TAG4) but determines what happens after the Sector Object is activated with a SO_MATCH_EVENT (Hitag 70)
<b>TAG8</b>	random frequency (0 - 64) - the lower the number the slower the random direction change

This allows you to scale simple Sector Objects. „Hold“ type maintains the default scale. „Destination“ moves either to the minimum or maximum scaling size. „Random“ is just that - the Sector Object scales larger and smaller randomly. „Cycle“ continuously scales from min to max and back again. „Random point“ is just like a

SO\_AMOEBA (Hitag 75), except that SO\_SCALE\_INFO gives you greater control over its behavior.

*EXAMPLES:* EE in EXAMPLE.MAP, the amoeba in \$SHRINE.MAP

*SEE ALSO:* SO\_AMOEBA (Hitag 75), SO\_SCALE\_POINT\_INFO (Hitag 101)

**Hitag 101 SO\_SCALE\_POINT\_INFO**

Lotag	vertices scale speed
Angle	random frequency (0 - 1024) - the lower the number the slower the direction change
TAG5	minimum distance - distance from default size to scale inward
TAG6	maximum distance - distance from default size to scale outward

SO\_SCALE\_POINT\_INFO is used with a SO\_SCALE\_INFO ST1 (Hitag 100) to provide information about how the vertices should behave. TAG5 and TAG6 are used to designate how far in and out the individual vertices should move at the speed in Lotag (TAG2). Be careful; when wall points cross each other, strange and unpredictable sector behavior can occur.

*EXAMPLES:* The amoeba in \$SHRINE.MAP

*SEE ALSO:* SO\_AMOEBA (Hitag 75), SO\_SCALE\_INFO (Hitag 100)

**Hitag 102 SO\_SCALE\_XY\_MULTIPLY**

TAG5	x multiply value (0-255; 256 is equal to 1)
TAG6	y multiply value (0-255; 256 is equal to 1)

Use in conjunction with SO\_SCALE\_INFO (Hitag 100) and SO\_SCALE\_POINT\_INFO (Hitag 101). Try experimenting with this to see what kind of effects you can achieve.

**Hitag 103 SECT\_WALL\_MOVE**

Lotag	match tag - triggering one of these will trigger all SECT_WALL_MOVE ST1's with the same matching tags. This will also trigger SPAWN_SPOT ST1's.
TAG3	match with a SECT_WALL_MOVE_CANSEE (Hitag 104)
Angle	point in the direction you want the wall point to move
TAG5	tile number used as texture for the „next wall“ when shot
TAG6	tile number used as texture for the „previous wall“ when shot
TAG7	shade value applied to the „next wall“ when shot
TAG8	shade value applied to the „previous wall“ when shot
TAG9	number of times the point can be shot
TAG10	random angle difference (1 - 128) - the larger the number the greater the angle deviance
TAG13	distance to move point each time it is shot
BOOL1	if set to: <div style="padding-left: 20px;"> <b>0</b> = it can work independently  <b>1</b> = it can only be activated when part of a TAG3 group </div>

This creates a dentable wall. Every time the wall vertex is shot with an explosive weapon, it will move the distance in TAG13, up to TAG9 times. Place the ST1 sprite

on TOP of the vertex to move. One vertex will move. Every vertex is associated with two walls: the „next wall“ and the „previous wall“. If TAG5 or TAG6 are non-zero, then that wall will take on the specified texture when the vertex is shot. If TAG7 or TAG8 are non-zero, then that wall will take on the new shade value specified. This will work for Sector Object walls with one limitation: you cannot tag the outside wall loop tagged with 504 to be moveable.

*NOTE:* Changing the value of TAG13 will cause the ST1 sprite to move strangely in 3D editing mode. Don't worry. If 2D mode says it's on the vertex, then it's on the vertex.

*EXAMPLES:* DA in EXAMPLE.MAP

*SEE ALSO:* SECT\_WALL\_MOVE\_CANSEE (Hitag 104), SPAWN\_SPOT (Hitag 69), SECT\_EXPLODING\_CEIL\_FLOOR (Hitag 57)

<b>Hitag 104</b>	<b>SECT_WALL_MOVE_CANSEE</b>
------------------	------------------------------

Lotag = match with TAG3 of SECT\_MOVE\_WALL (Hitag 103)

Placing sprites on wall points renders them somewhat unreliable for certain engine routines. If there is a problem activating a SECT\_MOVE\_WALL, set up one of these off of the wall point for the engine to use for detection purposes. I have never needed to use this.

*EXAMPLES:* DA in EXAMPLE.MAP

<b>Hitag 106</b>	<b>SECT_SPIKE</b>
------------------	-------------------

Lotag	match tag
TAG3	<p>type of the dynamic slope:</p> <p>a</p> <p>0 = manual/switch/trigger - primarily used for manual activation. If manual type has a match tag that is not 0 then ALL spikes with the same match type will be operated on. A spike can be manually operated from either within the sector, or to the side. With switch operation, the switch resets when the spike goes to default position. With trigger operation, it's the same as a switch but it's a floor trigger.</p> <p>1 = Switch/Trigger Only - cannot be operated manually, ALL spike types with the same matching tag get operated on by the switch or trigger.</p> <p>2 = Automatic - Starts active</p> <p>3 = Automatic - Starts inactive, must be activated with a switch or trigger.</p>
Angle	speed of the dynamic slope
TAG5	acceleration at which Angle (TAG4) speed is approached
TAG6	match tag at destination - when a spike reaches its destination it will trigger any event with this match tag
TAG7	Sector Object number to operate on. Set this to -1 if the SECT_SPIKE is not in a Sector Object.
TAG9	auto return time (in 1/8 <sup>th</sup> second tics, 8 tics = 1 sec.) - if non zero then the spike will return to its default position after the allotted time. If 0 then it will stay in the ON position until the spike is operated again.
BOOL1	<p>if set to:</p> <p>0 = it starts in the OFF position</p> <p>1 = it starts in the ON position</p>
BOOL2	<p>if set to:</p> <p>0 = it can be toggled</p> <p>1 = it can only be operated once</p>
BOOL3	<p>if set to:</p> <p>0 = it can crush the player</p> <p>1 = it can not crush the player</p>
BOOL4	<p>if set to:</p> <p>0 = it can be operated in WangBang (Deathmatch) games</p> <p>1 = it can not be operated in WangBang (Deathmatch) games</p>
BOOL5	<p>if set to:</p> <p>0 = it doesn't call TAG6 match when moving to OFF position</p> <p>1 = it calls TAG6 match when moving to ON or OFF position</p>
BOOL6	<p>if set to:</p> <p>0 = it waits until the motion is complete before it allows retriggering</p> <p>1 = it can be operated while it is moving</p>
BOOL7	<p>if set to:</p> <p>0 = actors can operate this spike</p> <p>1 = actors can not operate this spike</p>
BOOL11	<p>if set to:</p> <p>1 = displays „Single Play only“ message when operated in WangBang (Deathmatch) mode, use together with BOOL4 = 1</p>

A „spike“ is the term for a dynamic slope in Shadow Warrior. Useful for tilting ramps and slides, or special ceiling effects like a drill. A spike behaves pretty much like a vator, except instead of the floor height moving to the ST1, the floor instead slopes to the height of the ST1. Remember that slopes always hinge on the „first wall“ of a sector (set with Alt+F). To do a spike from the ceiling, flip the ST1 sprite upside down (press „F“ twice on the sprite in 3D mode). „Off“ position is the position of the sector in Build, „On“ position is the height of the ST1 sprite.

*WARNING:* Don't forget to put that -1 in TAG7! Shadow Warrior will usually crash if you forget to do this.  
*EXAMPLES:* DC in EXAMPLE.MAP  
*SEE ALSO:* SECT\_ROTATOR (Hitag 143), SECT\_SLIDOR (Hitag 145), SECT\_VATOR (Hitag 92), SLIDE\_SECTOR (Hitag 135)

<b>Hitag 108</b>	<b>LIGHTING</b>
------------------	-----------------

Lotag	match tag
TAG3	type of the light: <ul style="list-style-type: none"> <li>0 = constant</li> <li>1 = flicker</li> <li>2 = fade</li> <li>3 = flicker ON</li> <li>4 = fade to ON/OFF</li> </ul>
Angle	timing - the smaller the number, the faster a light flickers or fades
TAG5	brighten amount - this is not an actual shade amount, instead, this is the value the ST1 will subtract from the sector's shade value (+S in 3D mode) when it is turned on.
TAG6	darken amount - This is not an actual shade amount, instead, this is the value the ST1 will add to the sector's shade value when it is turned on.
TAG7	shade increment/decrement for fading lights - default is 1
BOOL 1	if set to: <ul style="list-style-type: none"> <li>0 = light is off</li> <li>1 = light is on</li> </ul>
BOOL2	if set to: <ul style="list-style-type: none"> <li>1 = light doesn't affect floor</li> </ul>
BOOL3	if set to: <ul style="list-style-type: none"> <li>1 = light doesn't affect ceiling</li> </ul>
BOOL4	if set to: <ul style="list-style-type: none"> <li>1 = light doesn't affect walls</li> </ul>
BOOL5	if set to: <ul style="list-style-type: none"> <li>1 = light will shade outer walls - red (2-sided) walls have inner and outer walls, by default LIGHTING will not shade the outer walls</li> </ul>
BOOL6	if set to: <ul style="list-style-type: none"> <li>1 = light doesn't affect the palette of the sector</li> </ul>
BOOL8	if set to: <ul style="list-style-type: none"> <li>1 = light inverts intensity - this means that the sector will be brightened by TAG6 amount when the light is on, and darkened by TAG5 amount when the light is off. This is useful for making lights that flicker when they are broken.</li> </ul>
BOOL10	determines the direction to start the fade

The sector will have the same palette as the LIGHTING sprite, even if the light is turned off. All LIGHTING ST1's with the same match tags will operate together.

*WARNING:* Every LIGHTING sprite tends to have its own independent rate of fade or flicker. This means that if you have a room comprised of multiple sectors and you put a flickering LIGHTING ST1 in every sector, they will all flicker at different rates and look really bad even if their Lotag (TAG2)'s match. If you want a room of uniform flickering or fading, you need to place a single LIGHTING sprite in one sector and use LIGHTING\_DIFFUSE (Hitag 109) in all the sectors you want to share that lighting effect.

*EXAMPLES:* DD in EXAMPLE.MAP  
*SEE ALSO:* LIGHTING\_DIFFUSE (Hitag 109)

<b>Hitag 109</b>	<b>LIGHTING_DIFFUSE</b>
------------------	-------------------------

Lotag	match to a LIGHTING (Hitag 108) Lotag
TAG3	shade multiplier - the higher this number, the more quickly the light grows dark with distance from the light source
Angle	distance (1 - 20 for diffuse lighting) - the larger the number the more distant and therefore darker the shade from the source
BOOL2	if set to: 1 = light doesn't affect floor
BOOL3	if set to: 1 = light doesn't affect ceiling
BOOL4	if set to: 1 = light doesn't affect walls
BOOL5	if set to: 1 = light affects outer walls (both sides of 2-sided walls)
BOOL6	if set to: 1 = light doesn't change palette

Used with LIGHTING (Hitag 108) to shade other sectors on a grade. Also used to copy a lighting effect (flickering, fading, palette) uniformly over multiple sectors. Though its difference from Duke Nukem 3D's lighting may at first seem confusing, this method is actually more convenient. Once all the LIGHTING\_DIFFUSE sprites are in place, you can change the lighting effect for an entire group of sectors just by changing the values of one sprite: the LIGHTING ST1 (Hitag 108).

*EXAMPLES:* DD in EXAMPLE.MAP  
*SEE ALSO:* LIGHTING (Hitag 108)

<b>Hitag 110</b>	<b>VIEW_LEVEL1</b>
------------------	--------------------

Lotag	view match tag
Angle	point up (north) to turn it OFF point down (south) to turn it ON

A matching VIEW\_LEVEL sprite must be in both the upper and lower sectors of any room-over-room areas where the player can look through the ceiling or floor into the other area. Use VIEW\_LEVEL1 in the lower areas. See SWBUILD.DOC for guidelines to building room-over-room areas.

*EXAMPLES:* AC, AD, and FC in EXAMPLE.MAP  
*SEE ALSO:* VIEW\_LEVEL2 (Hitag 111), VIEW\_THRU\_CEILING (Hitag 120), VIEW\_THRU\_FLOOR (Hitag 121), BOUND\_FLOOR\_BASE\_OFFSET (Hitag 202), BOUND\_FLOOR\_OFFSET (Hitag 203)

<b>Hitag 111</b>	<b>VIEW_LEVEL2</b>
------------------	--------------------

Lotag	view match tag
Angle	point up (north) to turn it OFF point down (south) to turn it ON

A matching VIEW\_LEVEL sprite must be in both the upper and lower sectors of any room-over-room areas where the player can look through the ceiling or floor into the other area. Use VIEW\_LEVEL2 in the upper areas. See SWBUILD.DOC for guidelines to building room-over-room areas.



*EXAMPLES:* AB, AE, and FE in EXAMPLE.MAP  
*SEE ALSO:* VIEW\_LEVEL1 (Hitag 110), VIEW\_THRU\_CEILING (Hitag 120), VIEW\_THRU\_FLOOR (Hitag 121), BOUND\_FLOOR\_BASE\_OFFSET (Hitag 202), BOUND\_FLOOR\_OFFSET (Hitag 203)

<b>Hitag 120</b>	<b>VIEW_THRU_CEILING</b>
------------------	--------------------------

Lotag = view match tag

Use this in a sector with a floor mirror texture on the ceiling in the lower half of a room-over-room area. The ST1 sprite must be in the same relative location in the sector as the VIEW\_THRU\_FLOOR ST1 (Hitag 121) sprite in the upper sector. Lotag (TAG2) will be the same as all the VIEW\_LEVEL sprites and the VIEW\_THRU\_FLOOR sprite. See SWBUILD.DOC for guidelines to building room-over-room areas.

*EXAMPLES:* AC, AD, and FC in EXAMPLE.MAP  
*SEE ALSO:* VIEW\_LEVEL1 (Hitag 110), VIEW\_LEVEL2 (Hitag 111), VIEW\_THRU\_FLOOR (Hitag 121), BOUND\_FLOOR\_BASE\_OFFSET (Hitag 202), BOUND\_FLOOR\_OFFSET (Hitag 203)

<b>Hitag 121</b>	<b>VIEW_THRU_FLOOR</b>
------------------	------------------------

Lotag = view match tag

Use this in a sector with a floor mirror texture on the floor in the upper half of a room-over-room area. The ST1 sprite must be in the same relative location in the sector as the VIEW\_THRU\_CEILING ST1 (Hitag 120) sprite in the lower sector. Lotag (TAG2) will be the same as all the VIEW\_LEVEL sprites and the VIEW\_THRU\_CEILING sprite. See SWBUILD.DOC for guidelines to building room-over-room areas.

*EXAMPLES:* AB, AE, and FE in EXAMPLE.MAP  
*SEE ALSO:* VIEW\_LEVEL1 (Hitag 110), VIEW\_LEVEL2 (Hitag 111), VIEW\_THRU\_CEILING (Hitag 120), BOUND\_FLOOR\_BASE\_OFFSET (Hitag 202), BOUND\_FLOOR\_OFFSET (Hitag 203)

<b>Hitag 132</b>	<b>BREAKABLE</b>
------------------	------------------

Lotag	match tag - used to match up with BREAKABLE_SPRITE (Hitag 132 on a non-ST1 sprite) or TAG_BREAK_WALL (Walltag 307)
TAG5	tile number to change to after BROKEN - not needed if TAG8 = „1“
TAG7	number of damage hits to take before a sprite or wall texture is BROKEN - note that if this is set to 2 or greater the tile number will increment toward TAG5 until TAG5 is reached. The graphics in the art file must be imported in the correct order to support this. Nothing in Shadow Warrior uses this, but the functionality is there for anybody who feels like importing their own art.
TAG8	<p>BROKEN type:</p> <p>0 = change the tile number only</p> <p>1 = kill sprite/wallpic when BROKEN - for walls this will turn off masking, blocking, and 1-way.</p> <p>2 = turn off blocking bits when BROKEN - for walls this will just turn off blocking bits, but leave the masking or 1-way alone. Breaking an air vent grate is an example of this.</p>
<p>Sprites and wall breaking info. There are 2 ways to use this:</p> <p>1) As ST1 sprite info for breaking a sprite, or</p> <p>2) As ST1 sprite info for breaking a wall.</p> <p>Example: All chairs that break could use a single ST1 with Hitag 132 to tell them how to break as long as the same matching Lotag (TAG2) is used. When hitting a wall tagged with TAG_WALL_BREAK (Walltag 307) the code will look for a BREAKABLE_SPRITE ST1 sprite with a matching tag and use this info if found.</p> <p><i>NOTE:</i> Can be re-used for many BREAKABLE_SPRITES.</p> <p><i>EXAMPLES:</i> DE in EXAMPLE.MAP</p>	

<b>Hitag 132</b>	<b>BREAKABLE_SPRITE</b> (used on non-ST1 sprites to make them breakable)
------------------	--

Lotag	match tag - used to match up with BREAKABLE ST1's (Hitag 132)
TAG6	match tag - extra match value for activating ST1's like SPAWN_SPOT and LIGHTING. By this method you can spawn debris or turn off a light when this sprite is broken.
BOOL1	<p>if set to:</p> <p>1 = denotes that this is not an ST1 but is tagged like one</p> <p>IMPORTANT: THIS MUST ALWAYS BE SET TO „1“!</p>
BOOL2	<p>if set to:</p> <p>1 = don't break the sprite, use this to override automatic breakables</p>
<p>Use this tagging on the actual sprite you want to break. Lotag (TAG2) should match the Lotag (TAG2) of the BREAKABLE ST1 (Hitag 132) that tells this sprite how to break.</p> <p><i>NOTE:</i> You don't actually need the ST1 132 if you just want to kill the sprite when its broken. Giving it a Hitag of 132 and setting BOOL1 to „1“ is sufficient.</p> <p><i>EXAMPLES:</i> DE in EXAMPLE.MAP</p>	

<b>Hitag 133</b>	<b>QUAKE_SPOT</b>
------------------	-------------------

Lotag	match tag
TAG3	shake Z (up/down) amount - this is the amount the view moves in the Z direction
Angle	radius of quake - distance in which quake is felt; this value is multiplied by 8 internally for the distance. The code is set up so that the quake amount decreases from the epicenter, but this effect can be overridden by setting BOOL1 to „1“
TAG5	duration of quake in seconds
TAG6	timed quake - shake the screen every X seconds, where X = TAG6 multiplied by 10
TAG7	shake angle amount - amount the view's angle moves
TAG8	shake x/y position amount - amount the view's angle moves
TAG9	random quake value (1 - 128) - multiplied by 4 internally. Used with TAG6. Instead of automatically spawning a quake after time it does a random test. The larger the number the greater the chance of a quake. 0 is default, means it will not do random tests.
BOOL1	if set to: 1 = quake does not decrease from epicenter and will basically shake the entire level
BOOL2	if set to: 1 = quake will only happen once and never again
BOOL3	if set to: 1 = it waits for trigger before starting timed quake code. This is only used for timed quakes.
This is used to shake the player's view as though an earthquake were happening.	
<i>EXAMPLES:</i> DE in EXAMPLE.MAP	

<b>Hitag 134</b>	<b>SOUND_SPOT</b>
------------------	-------------------

Lotag	match tag
Angle	1 <sup>st</sup> sound number
TAG5	2 <sup>nd</sup> sound number
TAG6	3 <sup>rd</sup> sound number
BOOL1	if set to: 1 = it plays the sound once, and only once
BOOL2	if set to: 1 = it player the sound as a looping sound, the sound will continue to loop until stopped by STOP_SOUND_SPOT (Hitag 150). It also means that it will be following the sprite and needs to be set if the ST1 sprite will be moving (usually with a Sector Object).
BOOL3	if set to: 1 = plays randomly. When the ST1 is triggered it will randomly choose between sounds 1 and 2, or between sounds 1, 2 and 3.
BOOL4	if set to: 1 = it doesn't pan the sound. This is useful when the sound is very near the player.
BOOL5	if set to: 1 = it uses the „Doppler“ effect (happens with moving sound sources)
BOOL6	set this to 0 0 = BOOL6 must always be zero
BOOL7	if set to: 1 = used for Lo Wang's sayings. Sound is always at maximum volume no matter how far the player is from the SOUND_SPOT.

SOUND\_SPOT will play the sound specified in Angle (TAG4) when the match tag in Lotag (TAG2) is triggered. SOUND\_SPOT can also be triggered by operating a vator, spike, sliding door, or rotating door with the same match tag. Sound numbers can be obtained from the Digital Sound Reference in this document. Don't confuse these with ambient sounds, which are used with the AMBIENT\_SOUND ST1 (Hitag 1002). If SOUND\_SPOT is placed in a Sector Object operational sector, the 1st sound (Angle) will be the 'idle' sound. The 2nd sound (TAG5) will be the 'drive' sound.

*IMPORTANT:* The match tag (Lotag) will still need to be a unique value even though the mapper never sets up a trigger for it. If you do not use a unique value then other SOUND\_SPOTs with the match value will do the same sound processing. This could be used as a feature - drive something and something else on the level makes a noise. Note that you can specify 3 different sounds. Some things like vators can have more than one sound attached for different operations such as moving to the ON and OFF positions. Most of the time only the 1st sound (Angle) will be used.

*EXAMPLES:* DB, DC, DD, and others in EXAMPLE.MAP

*SEE ALSO:* AMBIENT\_SOUND (Hitag 1002)

### Hitag 135 SLIDE\_SECTOR

Lotag = speed value (0 - 32, 0 is fastest)

Tag a sloped sector with this and it will cause the player to slide in the direction of the slope. Actors are not affected. The steeper the slope, and the lower the Lotag (TAG2) value, the faster the slide. Lo Wang will slide even if he is in the air over the sector. Use with SECT\_SPIKE (Hitag 106) to create a variable speed slide.

*EXAMPLES:* DF in EXAMPLE.MAP

### Hitag 136 CEILING\_FLOOR\_PIC\_OVERRIDE

Lotag	tile number of the texture
TAG 3	if set to: <ul style="list-style-type: none"> <li>0 = override texture on the ceiling</li> <li>1 = override texture on the floor</li> </ul>
TAG7	draw type: <ul style="list-style-type: none"> <li>0 = normal</li> <li>1 = masked</li> <li>2 = translucent</li> <li>3 = very translucent</li> </ul>
BOOL1	if set to: <ul style="list-style-type: none"> <li>1 = hitscan bit is set on the overridden texture</li> </ul>

This is primarily used to give a floor or ceiling texture to the floor mirror in room-over-room areas. This is used in Shadow Warrior to create the translucent water, the reflective floors, and the masked grating in \$VOLCANO.MAP. A sloped floor cannot have a masked or translucent texture. Use with FLOOR\_Z\_ADJUST (Hitag 98) to keep the player from falling through into the lower area. Actors can stand on it too, but will usually fall through when they are killed.

*EXAMPLES:* AB and AD in EXAMPLE.MAP  
*SEE ALSO:* FLOOR\_Z\_ADJUST (Hitag 98)

<b>Hitag 140</b>	<b>TRIGGER_SECTOR</b>
------------------	-----------------------

Lotag	match tag
TAG3	type of trigger sector: <ul style="list-style-type: none"> <li>0 = trigger when entering sector</li> <li>1 = trigger when leaving sector</li> <li>2 = continuous as long as player is standing on it</li> <li>3 = operated with the use key (Spacebar)</li> </ul>
Angle	angle player faces when TAG3 is set to 3
TAG5	trigger definition: <ul style="list-style-type: none"> <li>0 = remote Sector Object trigger</li> <li>1 = lava erupt trigger</li> </ul>
TAG7	remote Sector Object number (when TAG5 is set to 0)
BOOL1	if set to: <ul style="list-style-type: none"> <li>0 = player's view will follow the Sector Object</li> <li>1 = player's view will stay at the angle set with Angle</li> </ul>

TRIGGER\_SECTOR is only used for two things: driving remote controlled operational Sector Objects and triggering LAVA\_ERUPT ST1's (Hitag 56). BOOL1 on the LAVA\_ERUPT (Hitag 56) sprite must be set to „1“ for this trigger to work.

*EXAMPLES:* BF and EF in EXAMPLE.MAP  
*SEE ALSO:* SECT\_OPERATIONAL (Hitag 1), LAVA\_ERUPT (Hitag 56)

<b>Hitag 141</b>	<b>DELETE_SPRITE</b>
------------------	----------------------

Lotag	match tag
BOOL2	if set to: <ul style="list-style-type: none"> <li>1 = moves with a Sector Object</li> </ul>

Place a DELETE\_SPRITE ST1 at the same x, y location as another sprite. When the Lotag (TAG2) match is triggered, the sprites at the same x, y will be deleted. Works on most sprites but not all. Some that can not be deleted include decoration sprites, vator ST1's, spike ST1's and room-over-room view sprites.

*EXAMPLES:* DF in EXAMPLE.MAP  
*SEE ALSO:* BREAKABLE (Hitag 132)

**Hitag 143**    **SECT\_ROTATOR**

Lotag	match tag
TAG3	type of rotator: value <ul style="list-style-type: none"> <li><b>0</b> = manual/switch/trigger - if a manual type has a Lotag (TAG2) that is not 0 then <i>ALL</i> rotators, vators, spikes, or whatever with the same Lotag (TAG2) will be operated on. Rotators can be manually operated from on top of sector or alongside of it. A switch resets when the rotator returns to its default position. Trigger operation is the same as a switch but it's a floor trigger instead.</li> <li><b>1</b> = switch/trigger only - the rotator cannot be operated manually.</li> </ul> <i>ALL</i> rotator types with the same Lotag (TAG2) value get operated on.
TAG5	angle move amount (512 = 90 degrees) - use a negative value to make the sector rotate counter-clockwise
TAG6	match tag at destination - when a rotator reaches its destination it will trigger any event with this match tag
TAG7	angle increment - the speed at which the sector rotates (do not use negative values on this)
TAG9	auto return time (1/8 <sup>th</sup> second tics, 8 tics = 1 sec.) - if non-zero then the sector will return to its default position after the allotted time. If 0 then the sector will remain in the ON position until the SECT_ROTATOR is operated again.
BOOL2	if set to: <ul style="list-style-type: none"> <li><b>1</b> = it can only be operated once</li> </ul>
BOOL4	if set to: <ul style="list-style-type: none"> <li><b>0</b> = it can be operated during a WangBang (Deathmatch) game</li> <li><b>1</b> = it can not be operated during a WangBang game</li> </ul>
BOOL5	if set to: <ul style="list-style-type: none"> <li><b>0</b> = it triggers the TAG6 match only when moving to the ON position</li> <li><b>1</b> = it triggers the TAG6 match when moving to the ON and OFF position</li> </ul>
BOOL6	if set to: <ul style="list-style-type: none"> <li><b>0</b> = it can not be operated while in motion</li> <li><b>1</b> = it can be operated while in motion</li> </ul>
BOOL11	if set to: <ul style="list-style-type: none"> <li><b>1</b> = it displays a „Single Play only“ message when operated in WangBang game</li> </ul>

SECT\_ROTATOR is used primarily to create rotating (swinging) doors, although it has many other potential uses. Any number of red sector (2-sided) walls of any shape can be rotated simultaneously around a single pivot point, defined by SECT\_ROTATOR\_PIVOT (Hitag 144).

*EXAMPLES:* EA in EXAMPLE.MAP  
*SEE ALSO:* SECT\_ROTATOR\_PIVOT (Hitag 144), SECT\_VATOR (Hitag 92), SECT\_SPIKE (Hitag 106), SECT\_SLIDOR (Hitag 145).

**Hitag 144**    **SECT\_ROTATOR\_PIVOT**

Lotag = match tag

The CLOSEST SECT\_ROTATOR (Hitag 143) with same matching Lotag (TAG2) will rotate about this sprite.

*EXAMPLES:* EA in EXAMPLE.MAP

*SEE ALSO:* SECT\_ROTATOR (Hitag 143)

**Hitag 145**    **SECT\_SLIDOR**

Lotag	match tag
TAG3	type of slidor: value sector <i>ALL</i>
	<p><b>0</b> = manual/switch/trigger - if a manual type has a Lotag (TAG2) that is not 0 then <i>ALL</i> slidors with the same Lotag (TAG2) will be operated on. Rotators can be manually operated from on top of or alongside of it. A switch resets when the rotator returns to its default position. Trigger operation is the same as a switch but it's a floor trigger instead.</p> <p><b>1</b> = switch/trigger only - the slidor cannot be operated manually.</p> <p><i>ALL</i> slidor types with the same Lotag (TAG2) value get operated on.</p>
TAG5	distance to move
TAG6	match tag at destination - when a slidor reaches its destination it will trigger any even with this match tag
TAG7	speed of movement
TAG9	auto return time (1/8 <sup>th</sup> second tics, 8 tics = 1 sec.) - if non-zero then the slidor will return to its default position after the allotted time. If 0 then it will remain in the ON position until the SECT_ROTATOR is operated again.
BOOL2	if set to: <b>1</b> = it can only be operated once
BOOL3	if set to: <b>1</b> = doesn't crush the player. <i>WARNING:</i> never let a white walled slidor crush the player or the game will quit with an error.
BOOL4	if set to: <b>0</b> = it can be operated during a WangBang (Deathmatch) game <b>1</b> = it can not be operated during a WangBang game
BOOL5	if set to: <b>1</b> = slide door can be built open and will be closed when loading the map in Shadow Warrior. Only used to make level design easier.
BOOL6	if set to: <b>0</b> = it can not be operated while in motion <b>1</b> = it can be operated while in motion
BOOL7	if set to: position <b>0</b> = it triggers the TAG6 match only when moving to the ON position <b>1</b> = it triggers the TAG6 match when moving to the ON and OFF position
BOOL11	if set to: <b>1</b> = it displays a „Single Play only“ message when operated in WangBang game, used with BOOL4

Use SECT\_SLIDOR to create sliding or split doors. Several different types can be built, both with red walls and with white walls. Look to the game maps for various examples.

*IMPORTANT:* The walls that form the edges of the door must be tagged properly in order for the slidor to work. If, in 2D mode, the door slides to the left, tag the wall

TAG\_WALL\_SLIDOR\_LEFT (Walltag 220), if it slides to the right, tag the wall  
TAG\_WALL\_SLIDOR\_RIGHT (Walltag 221), if it slides upwards, tag it  
TAG\_WALL\_SLIDOR\_TOP (Walltag 222), and if it slides downwards, tag it  
TAG\_WALL\_SLIDOR\_BOTTOM (Walltag 223).  
*EXAMPLES:* EA in EXAMPLE.MAP  
*SEE ALSO:* SECT\_ROTATOR (Hitag 143), SECT\_VATOR (Hitag 92),  
SECT\_SPIKE (Hitag 106)

<b>Hitag 146</b>	<b>SECT_CHANGOR</b>
------------------	---------------------

Lotag	match tag
Angle	tile number of new texture
TAG5	Z (up/down) adjustment in pixels, can be negative
TAG6	slope adjustment, can be negative
TAG7	new shade value, can be negative
TAG8	new palette value
TAG9	new visibility value, can be negative
BOOL1	if set to: 0 = affects the floor 1 = affects the ceiling
BOOL2	if set to: 0 = kills this sprite after use 1 = does not kill this sprite after using once
<p>Change attributes of sector when match code is called. Can be used to create breakable floor or ceiling textures, or other special effects. Most commonly used for shootable lights.</p> <p><i>EXAMPLES:</i> EB in EXAMPLE.MAP  <i>SEE ALSO:</i> SECT_COPY_DEST (Hitag 58), SECT_COPY_SOURCE (Hitag 59)</p>	

<b>Hitag 147</b>	<b>SO_DRIVABLE_ATTRIB</b>
------------------	---------------------------



Lotag	angle speed (1 - 16384, 0 means default speed)
TAG3	angle „slide“ amount (1 - 100)
TAG5	match tag, triggered when the driveable is operated with the Spacebar
TAG6	movement speed (1 - 16384, 0 means default speed)
TAG7	movement „slide“ amount (1 - 100)
BOOL1	if set to: 1 = don't shake the screen when running into something
BOOL2	if set to: 1 = trigger match tag TAG5 + 1 when the player presses Spacebar to stop driving the Sector Object
BOOL3	if set to: 1 = Sector Object can only be driven from a remote TRIGGER_SECTOR (Hitag 140)
BOOL4	if set to: 1 = rectangular clipping - the outer 504 tagged bounding wall loop of the Sector Object serves as the clipping box. The wall loop <i>MUST</i> be a 4 wall rectangle. This is the only alternative to the default clipping provided by the operational code. This clipping is better behaved because it never allows the Sector Object to cross sector lines.

This ST1 provides some additional control over the behavior of driveable Sector Objects. It allows speed and angle velocities to be scaled and adds momentum with TAG3 and TAG7 to make the driving a little more realistic. Must be placed in the center tagged sector of the Sector Object.

*EXAMPLES:* EF in EXAMPLES.MAP

*SEE ALSO:* SECT\_OPERATIONAL (Hitag 1), SECT\_SO\_DONT\_ROTATE (Hitag 45), SO\_LIMIT\_TURN (Hitag 68), SO\_TURN\_SPEED (Hitag 55)

#### Hitag 148 WALL\_DONT\_STICK

Angle = point sprite at wall

Point this at a wall, and nothing will stick to that wall. This keeps bullet holes, blood, sticky bombs, and so forth from landing on the wall.

*EXAMPLES:* BB in EXAMPLE.MAP

#### Hitag 149 SPAWN\_ITEMS

Lotag	match tag
TAG3	item to spawn: (see table below)
Angle	direction to throw the item
TAG7	velocity (0 - 255) - set to 0 and the item stays in place
TAG8	jump velocity (0 - 255)
BOOL1	if set to: 0 = kills the ST1 after it was triggered 1 = doesn't kill the ST1 after triggering
BOOL2	if set to: 1 = it doesn't spawn another item if a sprite exists in the exact position of the SPAWN_ITEMS sprite

This is used to spawn an item when the Lotag (TAG2) match is triggered. Used for placing items in trash cans, dropping items from the sky and any number of cool uses. Flag the ST1 as a Multiplayer only item (F8 menu, option 2) and the item will only spawn during a WangBang game.

*EXAMPLES:* EB in EXAMPLE.MAP  
*SEE ALSO:* SPAWN\_SPOT (Hitag 69)

TAG3	Item to spawn
0/32	Uzi
1	Red Key
2	Blue Key
3	Green Key
4	Yellow Key
5	Red Card
6	Blue Card
7	Green Card
8	Yellow Card
9	Gold Skeleton Key
10	Silver Skeleton Key
11	Bronze Skeleton Key
12	Red Skeleton Key
20	Uzi for Floor
40	Repair Kit
41	Shurikens
42	Sticky Mines
43	Rocket Launcher
44	Rocket Battery
45	Grenade Launcher
46	Grenades

TAG3	Item to spawn
47	Rail Gun
48	Rail Ammo
51	Shotgun
52	Shotgun Ammo
53	Head
55	Heart
57	Gray Armor (can't spawn red armor)
58	Portable Medkit
59	Small Medkit
60	Chem Bomb
61	Flash Bomb
62	Nuke
63	Caltrops
64	Cookie
65	Heat seeker card
66	Cloak
67	Night Vision
90	Boy Bunny
91	Girl Bunny
92	Random Bunny

**Hitag 150 STOP\_SOUND\_SPOT**

Lotag	match tag
TAG5	matching SOUND_SPOT whose sound is to stop

Use this to stop a looping sound when Lotag (TAG2) is triggered. TAG5 is the Lotag (TAG2) match tag of the SOUND\_SPOT (Hitag 134) to be stopped.

*EXAMPLES:* EB in EXAMPLE.MAP  
*SEE ALSO:* SOUND\_SPOT (Hitag 134)

**Hitag 202 BOUND\_FLOOR\_BASE\_OFFSET**

Lotag = bound floor number (first one is 0, then 1, 2, 3, etc.)

All areas with this tag in them set up a new BASE OFFSET to move other areas to. Lotag (TAG2) is the order in which they are processed. Consider this to be the "anchor" to which BOUND\_FLOOR\_OFFSET (Hitag 203) areas are moved.

*EXAMPLES:* AB, AC, FB, and FC in EXAMPLE.MAP  
*SEE ALSO:* BOUND\_FLOOR\_OFFSET (Hitag 203)

**Hitag 203 BOUND\_FLOOR\_OFFSET**

Lotag = bound floor number (first one is 0, then 1, 2, 3, etc.)

All bound areas with this tag in them are moved in the game to the last BOUND\_FLOOR\_BASE\_OFFSET sprite. „Bound“ areas means all sectors that are either directly or indirectly attached to the sector with BOUND\_FLOOR\_OFFSET in it. This allows you to build stacked sectors separately and let Shadow Warrior overlap them for you. Useful mainly for room-over-room and special dive sector cases. If a Sector Object is part of a bound floor group, it can have a matching 504 wall loop in the bound floor base group, which will move and behave exactly like the Sector Object. This is incredibly useful for a variety of effects, like the bottom of a boat in the underwater sector stays with the boat in the upper sector, and so on.

*EXAMPLES:* AD, AE, FE, and FF in EXAMPLE.MAP

*SEE ALSO:* BOUND\_FLOOR\_BASE\_OFFSET (Hitag 202)

<b>Hitag 500-596</b>	<b>SECT_SO_BOUNDING</b>
----------------------	-------------------------

Bounding sprites for Sector Objects. Every Sector Object needs bounding sprites, one in the upper left corner (BOUND\_SO\_UPPER) and one in the lower right corner (BOUND\_SO\_LOWER). These two sprites form the opposing points of an imaginary rectangle that must encompass the entire Sector Object, plus any sprites you want to move with it (like a gun barrel.) You are allowed to have a maximum of 20 Sector Objects in a level. The BOUND\_SO\_UPPER sprite Hitag of a Sector Object is calculated as  $500 + (5 * \text{Sector Object number})$ . The BOUND\_SO\_LOWER sprite Hitag of a Sector Object is calculated as  $501 + (5 * \text{Sector Object number})$ . So, Sector Object 0 uses BOUND\_SO\_UPPER 500 and BOUND\_SO\_LOWER 501. Sector Object 1 uses BOUND\_SO\_UPPER 505 and BOUND\_SO\_LOWER 506. This goes up in increments of 5 all the way to sector object 19, which uses BOUND\_SO\_UPPER 595 and BOUND\_SO\_LOWER 596.

*EXAMPLES:* EC, ED, EE, EF, FA, FB, and FC in EXAMPLE.MAP

<b>Hitag 1000</b>	<b>MIRROR_CAM</b>
-------------------	-------------------

Lotag	match tag
Angle	angle of the camera
TAG6	camera maximum turn angle (0 - 2048). This is the angle from center that the camera will rotate back and forth (512 = 90 degrees each way for 180 degrees range of motion).
TAG7	camera look up/down amount (5 - 195)
BOOL1	if set to: <ul style="list-style-type: none"> <li>0 = draws monitor to look like a magic view into another place, looks more like reality</li> <li>1 = draws monitor using draw to tile. This makes it look like a TV screen</li> </ul>
BOOL2	if set to: <ul style="list-style-type: none"> <li>0 = camera doesn't rotate</li> <li>1 = rotates the camera by angle specified in TAG6 from current camera angle</li> </ul>
BOOL3	if set to: <ul style="list-style-type: none"> <li>0 = negative turn direction</li> <li>1 = positive turn direction</li> </ul>
BOOL11	if set to (BOOL1 must be „1“): <ul style="list-style-type: none"> <li>0 = draws normal TV camera view</li> <li>1 = cycles through all players' views in 5 second increments on this monitor</li> </ul>

The camera's view can be displayed in a monitor built using a mirror. Tag the mirror wall with Lotag 306 and a Hitag equal to the MIRROR\_CAM Lotag (TAG2) match tag.

*NOTE:* Mirrors MUST be built using one way walls, not masked walls or they won't operate. A TV monitor requires a TV monitor sprite (tile numbers 3830-3837) with a Hitag equal to the MIRROR\_CAM Lotag (TAG2) match tag. The TV monitor sprite must be placed in front of a mirror.

*EXAMPLES:* GB and GC in EXAMPLE.MAP

### Hitag 1002 AMBIENT\_SOUND

Lotag = sound number of ambient sound to play

Plays an ambient sound. Look at the Ambient Sound Reference in this document to find the ambient sound number you want.

*EXAMPLES:* EC and ED in EXAMPLE.MAP

*SEE ALSO:* SOUND\_SPOT (Hitag 134)

### Hitag 1005 ECHO\_SPOT

Lotag reverb setting (100 - 200)

Angle radius that the reverb effect can be heard

Causes all sounds in the area around the ECHO\_SPOT to reverb by the amount set in Lotag (TAG2). Note that there is no reverb available when using an Ensoniq Soundscape soundcard due to technical reasons.

*EXAMPLES:* \$YAMATO.MAP, plus many other game maps

### Hitag 1006 DRIP\_GENERATOR

Lotag	delay in seconds - if 0 then it uses internally programmed delays
BOOL1	drip type: <b>0</b> = blood drips <b>1</b> = water drips
BOOL2	if set to: <b>0</b> = spawns drips <b>1</b> = spawns bubbles instead
<p>Used to create a dripping or bubbling spot. The Z height of the ST1 is important, because that's the height where the drip or bubble will spawn.</p> <p><i>EXAMPLES:</i> AA in EXAMPLE.MAP</p>	

## 8) Track (T#) Sprite Reference

by Keith Schuler, revised by Steffen Itterheim

### *Track Sprite Reference Contents*

#### I) How to use the Track Sprite 70

- 8.1.1 Meet the Track Sprites
- 8.1.2 Building a Track
- 8.1.3 Tagging a Track Sprite
  - 8.1.3.a Lotag
  - 8.1.3.b Hitag
- 8.1.4 Using a Track
  - 8.1.4.a Sector and Sprite Objects
  - 8.1.4.b Actors

#### II) Track Tag Function Reference 71

- 700 TRACK\_START
- 701 TRACK\_END
- 702 TRACK\_SET\_SPEED
- 703 TRACK\_STOP
- 704 TRACK\_REVERSE
- 705 TRACK\_SPEED\_UP
- 706 TRACK\_SLOW\_DOWN
- 707 TRACK\_VEL\_RATE
- 709 TRACK\_ZUP
- 710 TRACK\_ZDOWN
- 711 TRACK\_ZRATE
- 712 TRACK\_ZDIFF\_MODE
- 715 TRACK\_SPIN
- 716 TRACK\_SPIN\_STOP
- 717 TRACK\_BOB\_START
- 718 TRACK\_BOB\_SPEED
- 719 TRACK\_BOB\_STOP
- 720 TRACK\_SPIN\_REVERSE
- 723 TRACK\_SO\_SINK
- 724 TRACK\_SO\_FORM\_WHIRLPOOL
- 725 TRACK\_MOVE\_VERTICAL
- 726 TRACK\_WAIT\_FOR\_EVENT
- 728 TRACK\_MATCH EVERYTHING
- 729 TRACK\_MATCH EVERYTHING\_ONCE
- 751 TRACK\_ACTOR\_STOP
- 752 TRACK\_ACTOR\_REVERSE
- 753 TRACK\_ACTOR\_SPEED\_UP
- 754 TRACK\_ACTOR\_SLOW\_DOWN
- 755 TRACK\_ACTOR\_VEL\_RATE
- 771 TRACK\_ACTOR\_JUMP
- 772 TRACK\_ACTOR\_CRAWL
- 780 TRACK\_ACTOR\_CLOSE\_ATTACK1
- 782 TRACK\_ACTOR\_ATTACK1
- 791 TRACK\_ACTOR\_OPERATE
- 792 TRACK\_ACTOR\_CLIMB\_LADDER
- 799 TRACK\_ACTOR\_WAIT\_FOR\_PLAYER

## 8.1) How to use the Track Sprite

Tracks are a powerful feature of Shadow Warrior. Similar to the "Locators" in Duke Nukem 3D, tracks are a series of sprites which determine the path certain game objects can follow. Sector objects, sprite objects, and some actors can follow tracks.

### 8.1.1) Meet the Track Sprites

You can find the track sprites in the tileset starting at tile 1900 and proceeding all the way up to tile 1999. They are blue numbers counting up from 0 to 99. A single track uses one of these tiles, (i.e. track #0 only uses tile 1900), so this means you can have up to 100 different tracks in a single level. In 2D mode, the sprite will have a sign that reads „T0“, „T1“ or whatever depending on which track number it is.

### 8.1.2) Building a Track

Build a track by placing track sprites all along the route you want an object to take. All of the track sprites in a single track use the same tile (i.e. the sprites for track #0 will all use tile 1900). When an object follows a track, it will move in a straight line from one track sprite to the next, then change direction and move in a straight line to the next track sprite. It will do this around the entire track until it reaches the end, then it will head in a straight line back to the start and begin the process all over again. All tracks must loop like this, and a track can never cross its own path.

### 8.1.3) Tagging a Track Sprite

Track sprites only make use of their Hitag and Lotag. Not all track sprites need a lotag, if you want the object to travel to that point, but don't want to alter its behavior.

#### 8.1.3.a) Lotag

The track sprite's lotag is its „Tracktag“. It identifies what kind of behavior should occur at that point when an object reaches it while following the track. Different Tracktag values allow for changing speeds, bobbing, firing, waiting, and much more.

#### 8.1.3.b) Hitag

The Hitag is used to give additional information for those tracktags that need it. For instance, TRACK\_SET\_SPEED (Tracktag 702) uses its Hitag as the actual speed value.

### 8.1.4) Using a Track

Some tracktags only work for Sector Objects, while others only work for actors. Most tracktags work for both Sector Objects and actors.

#### 8.1.4.a) Sector and Sprite Objects

To make a Sector Object (or a Sector Object tagged as a sprite object) follow a track, just set the sector hitag of the center sector to the number of the track. So if you've got a track #5 in your level, you can make a Sector Object follow that track by setting the center sector hitag to 5.

#### 8.1.4.b) Actors

Some actors will follow a track, with varying degrees of success. To make an actor follow a track, give the actor sprite a lotag of 30000 + the track number. So to make a green guardian follow track #10, you would give him a lotag of 30010. Angle the actor to the right to make him move in the forward direction (from start to end) or angle the actor to the left to make him move in the backward direction (from end to start.)

## 8.II) Track Tag Function Reference

Examples of most of these can be found throughout Shadow Warrior. A few simple tracktags are also demonstrated in EXAMPLE.MAP.

<b>Lotag 700</b>	<b>TRACK_START</b>
	Every track needs a TRACK_START and a TRACK_END (tracktag 701). This is the first sprite on a track, and it marks the beginning.

<b>Lotag 701</b>	<b>TRACK_END</b>
	Every track needs a TRACK_START (Tracktag 700) and a TRACK_END. This is the last sprite on a track. When an object encounters this, it loops back to the TRACK_START.

<b>Lotag 702</b>	<b>TRACK_SET_SPEED</b>
------------------	------------------------

Hitag = target speed (0 - 128)
Set the target speed at this point. Setting the hitag to 0 is a good way to make sure the object stops moving once it reaches this point. (See the boat in \$WHIRL.MAP)

<b>Lotag 703</b>	<b>TRACK_STOP</b>
------------------	-------------------

Hitag = seconds to stop
Object will pause at this point on the track for the number of seconds in the hitag.

<b>Lotag 704</b>	<b>TRACK_REVERSE</b>
	Reverses the object's direction. It will now travel along the track in the opposite direction. Causes some tags to have an opposite effect (e.g. Tracktags 705 and 706)



<b>Lotag 705</b>	<b>TRACK_SPEED_UP</b>
	Hitag = speed offset (0 - 128)
	Target speed will increase by the amount in the hitag.
<b>Lotag 706</b>	<b>TRACK_SLOW_DOWN</b>
	Hitag = speed offset (0 - 128)
	Target speed will decrease by the amount in the hitag.
<b>Lotag 707</b>	<b>TRACK_VEL_RATE</b>
	Hitag = acceleration rate (1 - 12, default is 6)
	The rate at which the actual speed approaches the target speed.
<b>Lotag 709</b>	<b>TRACK_ZUP</b>
	Hitag = amount of pixels to go up
	Moves the floor of the object up by the number of pixels in the hitag.
<b>Lotag 710</b>	<b>TRACK_ZDOWN</b>
	Hitag = amount of pixels to go down
	Moves the floor of the object down by the number of pixels in the hitag.
<b>Lotag 711</b>	<b>TRACK_ZRATE</b>
	Hitag = rate of movement for TRACK_ZUP/TRACK_ZDOWN (default is 256)
	Sets the rate at which TRACK_ZUP/TRACK_ZDOWN moves.
<b>Lotag 712</b>	<b>TRACK_ZDIFF_MODE</b>
	Hitag = Z offset
	Special mode where the object looks at the Z-height of the next track point and moves up or down to achieve the destination. In the hitag is a negative delta Z from the placement of the track sprites' Z value.
<b>Lotag 715</b>	<b>TRACK_SPIN</b>
	Hitag = spin speed
	Object spins clockwise as it goes along its track.
<b>Lotag 716</b>	<b>TRACK_SPIN_STOP</b>

Hitag = spin angle (0 - 2048)  
Object stops spinning. It will then point in the direction indicated by the hitag.

**Lotag 717**    **TRACK\_BOB\_START**

Hitag = pixel height to bob  
Causes a Sector Object to start bobbing up and down like a cork.

**Lotag 718**    **TRACK\_BOB\_SPEED**

Hitag = speed at which to bob  
Sets the speed at which a Sector Object will bob.

**Lotag 719**    **TRACK\_BOB\_STOP**

The Sector Object will stop bobbing.

**Lotag 720**    **TRACK\_SPIN\_REVERSE**

Hitag = spin speed  
Object spins counter-clockwise as it goes along its track.

**Lotag 723**    **TRACK\_SO\_SINK**

Hitag = speed of sinking  
Start Sector Object sinking to ST1 SECT\_SO\_SINK\_DEST (hitag 33). Will sink at the speed in the hitag. Look at the boat in \$WHIRL.MAP for an example.

**Lotag 724**    **TRACK\_SO\_FORM\_WHIRLPOOL**

Will cause a Sector Object to lower the sectors tagged with a SECT\_SO\_FORM\_WHIRLPOOL ST1 (Hitag 37). Look at the boat in \$WHIRL.MAP for an example.

**Lotag 725**    **TRACK\_MOVE\_VERTICAL**

Hitag = speed of movement (default is 256)  
Move sprite objects (not Sector Objects) straight up or down in the Z direction to the next point's Z height then continue along the track.

**Lotag 726**    **TRACK\_WAIT\_FOR\_EVENT**

Hitag = match tag

Object will wait at this point on the track for a trigger or switch to be activated. Most types of triggers and switches with the same match tag can be used to free the Sector Object to continue. Look at the moving crane in \$SEABASE.MAP for an example.

**Lotag 728 TRACK\_MATCH\_EVERYTHING**

Hitag = match tag

When the object or actor reaches this point, it will trigger everything with this match tag.

**Lotag 729 TRACK\_MATCH\_EVERYTHING\_ONCE**

Hitag = match tag

When the object or actor reaches this point, it will trigger everything with this match tag once, and never again.

**Lotag 751 TRACK\_ACTOR\_STOP**

Hitag = seconds to stop

Causes an actor to stop for the number of seconds in the hitag.

**Lotag 752 TRACK\_ACTOR\_REVERSE**

Reverses the actor's direction. It will now travel along the track in the opposite direction. Note that Tracktags 753 and 754 will have the opposite effect when traveling the reverse direction

**Lotag 753 TRACK\_ACTOR\_SPEED\_UP**

Hitag = speed offset (0 - 128)

The actor's target speed will increase by the amount in the hitag.

**Lotag 754 TRACK\_ACTOR\_SLOW\_DOWN**

Hitag = speed offset (0 - 128)

The actor's target speed will decrease by the amount in the hitag.

**Lotag 755 TRACK\_ACTOR\_VEL\_RATE**

Hitag = acceleration rate (1 - 12, default is 6)

The rate at which the actor's actual speed approaches the target speed.

**Lotag 771 TRACK\_ACTOR\_JUMP**

Hitag = height to jump (default is 384)
Actor will jump. Only works with actors who can jump, like evil ninjas, rippers, and bunnies.

<b>Lotag 772</b>	<b>TRACK_ACTOR_CRAWL</b>
Causes an actor to crawl. Only works with evil ninjas.	

<b>Lotag 780</b>	<b>TRACK_ACTOR_CLOSE_ATTACK1</b>
An actor will do his close range attack, like the sword attack in the case of the green guardian.	

<b>Lotag 782</b>	<b>TRACK_ACTOR_ATTACK1</b>
An actor will do his long range attack, like shooting fireballs in the case of the green guardian.	

<b>Lotag 791</b>	<b>TRACK_ACTOR_OPERATE</b>
Point the track sprite angle in the direction of the operable sector, wall, or switch. Actor will operate the switch, door, or whatever, just like a player pressing Spacebar. May only work in certain cases.	

<b>Lotag 792</b>	<b>TRACK_ACTOR_CLIMB_LADDER</b>
The track sprite angle must be facing the ladder. Only evil ninjas can climb ladders.	

<b>Lotag 799</b>	<b>TRACK_ACTOR_WAIT_FOR_PLAYER</b>
Hitag = range (1024 = largest grid square in Build)	
The actor will wait for the player to come into range before moving from this point. Look at the rabbit on the mountain in \$SHRINE.MAP for an example of this.	

## 9) Misc Sprites Reference

by Keith Schuler, revised by Steffen Itterheim

### Misc Sprites Reference Contents

<b>9.I) Actors</b>	<b>77</b>
9.I.1 Hornet	(Tile # 800)
9.I.2 Bouncing Betty	(Tile # 817)
9.I.3 Accursed Head	(Tile # 820)
9.I.4 Sumo Boss	(Tile #1210)
9.I.5 Serpent Boss	(Tile #1300)
9.I.6 Coolie	(Tile #1400)
9.I.7 Coolie Ghost	(Tile #1441)
9.I.8 Green Guardian	(Tile #1469)
9.I.9 Little Ripper	(Tile #1580)
9.I.10 Fish	(Tile #3780)
9.I.11 Evil Ninja	(Tile #4096)
9.I.12 Crouching Evil Ninja	(Tile #4162)
9.I.13 Big Ripper	(Tile #4320)
9.I.14 Bunny	(Tile #4550)
9.I.15 Female Warrior	(Tile #5162)
9.I.16 Zilla Boss	(Tile #5426)
9.I.17 Leaders and Followers	
9.I.18 Spawning Actors with a Trigger	
<b>9.II) Cracks (TAG_SPRITE_HIT_MATCH)</b>	<b>79</b>
257 TAG_SPRITE_HIT_MATCH	
<b>9.III) Switches</b>	<b>79</b>
	(Tile #551-584)
206 TAG_SPRITE_SWITCH_VATOR	
210 TAG_LIGHT_SWITCH	
211 TAG_SWITCH_EVERYTHING	
212 TAG_SWITCH_EVERYTHING_ONCE	
213 TAG_COMBO_SWITCH_EVERYTHING	
214 TAG_COMBO_SWITCH_EVERYTHING_ONCE	
216 TAG_SPAWN_ACTOR_SWITCH	
304 TAG_ROTATE_SO_SWITCH	
9.III.1 Shootable Switch	(Tile # 577)
9.III.2 Exit Switch	(Tile #2470)
<b>9.IV) Keys and Keyholes</b>	<b>81</b>
9.IV.1 Keys	(Tile #1765-1779)
9.IV.2 Keyholes	(Tile #1846, 1850, 1852)
<b>9.V) TV Monitors</b>	<b>82</b>
	(Tile #3830-3837)
<b>9.VI) QuickJump and QuickJumpDown</b>	<b>82</b>
9.VI.1 QuickJump	(Tile #2309)
9.VI.2 QuickJumpDown	(Tile #2310)
<b>9.VII) Gratings</b>	<b>82</b>
<b>9.VIII) DeathFlag(tm) Capture the Flag</b>	<b>83</b>
9.VIII.1 The Flag Sprite	(Tile #2520)

## 9.1) Actors

### 9.1.1) Hornet (Tile #800)

Use a sprite with tile number 800 to place a hornet in your map.

### 9.1.2) Bouncing Betty (Tile #817)

This sprite also comes in voxel (3D sprite) format. This actor is functionally the same as an „Accursed Head“. A Betty can be tagged to produce different effects. Set the Hitag to these values for the following effects:

Hitag	if set to: <b>0</b> = explosion (the Betty just explodes) <b>1</b> = chem bombs <b>2</b> = caltrops <b>3</b> = flash effect (like a flash bomb) <b>4</b> = grenades
Lotag	number of items to spawn, depending on Hitag: Hitag <b>1</b> = Lotag: # of chem bombs (max 3, default 2) Hitag <b>2</b> = Lotag: # of caltrops (max 10, default 5) Hitag <b>4</b> = Lotag: # of grenades (max 10, default 5)

### 9.1.3) Accursed Head (Tile #820)

This is functionally the same as a Bouncing Betty and can be tagged similarly (see above).

### 9.1.4) Sumo Boss (Tile #1210)

If you set the sprite's palette to 16, he will be a „mini-boss“, which is smaller and less powerful.

### 9.1.5) Serpent Boss (Tile #1300)

If you set the sprite's palette to 16, he will be a „mini-boss“, which is smaller and less powerful.

### 9.1.6) Coolie (Tile #1400)

Use a sprite with tile number 1400 to place a coolie in your map.

### 9.1.7) Coolie Ghost (Tile #1441)

Use a sprite with tile number 1441 to place a coolie ghost in your map.

### 9.1.8) Green Guardian (Tile #1469)

Use a sprite with tile number 1469 to place a guardian in your map.

### 9.1.9) Little Ripper (Tile #1580)

If you set the sprite's palette to 7, he will become a big boss ripper.

### 9.1.10) Fish (Tile #3780)

Use a sprite with tile number 3780 to place a fish in your map. Note that fish are not well placed on land.

### 9.1.11) Evil Ninja (Tile #4096)

You can set the palette on an Evil Ninja to make him fire different kinds of weaponry.

Palette No.	Type of Ninja
0	Default Ninja, his pants appear blue in Build but will be brown in the game. This Ninja throws shurikens (darts in the UK version).
17	Gray Ninja, he's tough and fires grenades.
19	Red Ninja, he's tough and fired rockets.
21	Shadow Ninja, he's really tough and can fire two kinds of napalm plus use flash bombs.
24	Orange Ninja, like a Red Ninja but fires heat seeking rockets.

### 9.1.12) Crouching Evil Ninja (Tile #4162)

This type of ninja stays crouched except to fire at the player. This is great for ducking behind crates and whatnot. His palette can be set just like the regular evil ninja, and he will take on the toughness of that color, but he won't fire the heavy weaponry.

### 9.1.13) Big Ripper (Tile #4320)

If you set the sprite's palette to 7, he will become a big boss ripper.

### 9.1.14) Bunny (Tile #4550)

Bunnies come in three flavors. Set their palette to define which kind the actor is. You can also scale a bunny down to make it younger. It will grow into an adult before mating.

Palette No.	Type of Bunny
0	Female bunny. Appears blue in Build but will be light brown in the game.
17	Holy Handgrenade!! A white bunny! It's a Holy Grail guarding, Monty Python style, monstrous, fearsome killer rabbit! Run for your lives!!!! AAAaaarrghh.....
24	Male bunny. Will be dark brown in the game.

### 9.1.15) Female Warrior (Tile #5162)

Use a sprite with tile number 5162 to place a female warrior in your map.

### 9.1.16) Zilla Boss (Tile #5426)

Use a sprite with tile number 5426 to place a Zilla boss in your map. There is no mini-boss version of Zilla.

### 9.1.17) Leaders and Followers

Give an actor a hitag of 1008 and it will be part of a group mentality. If the actor has a lotag of 1, it will be a follower, while if it has a lotag of 2, it will be a leader. Followers will hover around the nearest leader unless the player is within range. This works especially well with rippers and fish.

### 9.1.18) Spawning Actors with a Trigger

Give any actor a TAG\_SPAWN\_ACTOR tag (lotag 203), and it will not appear in the map immediately when you start out. The hitag of the actor is the match tag. When that match tag is triggered, with a floor trigger or a switch or whatever, the actor will be magically teleported in.

## 9.II) Cracks (TAG\_SPRITE\_HIT\_MATCH)

Any sprite in the game can be given a lotag of 257 and it will then become a TAG\_SPRITE\_HIT\_MATCH. This means that when the sprite is shot by the player, it will trigger any event matched to the hitag of the TAG\_SPRITE\_HIT\_MATCH sprite. By default, the sprite will be invisible, unless it uses tile number 80, in which case it will be visible. Tile number 188 is another crack sprite, designed to be usable on ceilings or floors, but it will still be invisible by default.

<b>Lotag 257</b>	<b>TAG_SPRITE_HIT_MATCH</b>
<b>Hitag</b> = match tag, can trigger anything	
<b>TAG3</b> = default is 32; the larger this number, the more sensitive the sprite is to explosion radius	
<b>TAG7</b> = reacts to ammo type: <b>0</b> = default; react only to explosions with radius damage <b>1</b> = react only to a direct hit by a weapon with explosive radius (e.g. hit it directly with a rocket) <b>2</b> = react to any explosive weapon or any hitscan weapon (uzis or riot gun) <b>3</b> = react to any weapon hit, including shurikens, sword and fist <b>4</b> = react only to a Sector Object weapon fire. This means any type of Sector Object weapon, including machine gun	
<b>TAG8</b> = this is kind of a weird tag. Add the numbers of the effects you want together and put the total in TAG8: <b>+1</b> = force the sprite to be invisible <b>+2</b> = force the sprite to be visible <b>+4</b> = do not kill this sprite when it is hit. Can be triggered repeatedly <b>+8</b> = bouncing a grenade off of the sprite counts as a hit	
<i>EXAMPLES:</i> Rooms CA and EB in EXAMPLE.MAP	

## 9.III) Switches

(Tile #551-584)

Any sprite can be tagged as a switch, but it will only animate if it's using one of the tile numbers 551 through 584. There are several different switch tags, used to activate different types of events. A switch sprite's lotag identifies what kind of switch it is, while its hitag is a match tag to the event you wish to trigger. In most cases a switch will not make a sound. You must place a SOUND\_SPOT (hitag 134) ST1 nearby with a Lotag (TAG2) that matches the switch's hitag. I recommend sound #201 for most switches.

<b>Lotag 206</b>	<b>TAG_SPRITE_SWITCH_VATOR</b>
<b>Hitag</b> = match tag	
This switch is used to activate vators, spikes, rotators, and slidors - basically any kind of door. This is the best switch to use with a door because multiple switches with the same match will animate, and the switch pays attention to auto-closing doors, doors that have a player holding them open, etc. etc. When placing the switch in the BUILD editor, be sure to use the frame that matches the door's starting position. For instance, if BOOL1 is set to „1“ on a vator, be sure to use the second frame of a switch, because that door considers itself to be „on“. Otherwise, the switch might not animate the first time you operate it.	



<b>Lotag 210</b>	<b>TAG_LIGHT_SWITCH</b>
------------------	-------------------------

Hitag = match tag
This switch can only be used to activate LIGHTING ST1's (Hitag 108), but there's no real good reason to do this, since a TAG_SWITCH_EVERYTHING (Lotag 211) will work just as well. It's only included here because you'll find it in several Shadow Warrior maps.

<b>Lotag 211</b>	<b>TAG_SWITCH_EVERYTHING</b>
------------------	------------------------------

Hitag = match tag
This switch is the one you will use most often, because it can trigger any event in the game.

<b>Lotag 212</b>	<b>TAG_SWITCH_EVERYTHING_ONCE</b>
------------------	-----------------------------------

Hitag = match tag
This is the same as TAG_SWITCH_EVERYTHING (Lotag 211) in that it can be used to trigger any event, but this switch can only be activated once, and never again.

<b>Lotag 213</b>	<b>TAG_COMBO_SWITCH_EVERYTHING</b>
------------------	------------------------------------

Hitag	match tag
TAG3	if set to: 0 = switch must be OFF to trigger the event 1 = switch must be ON to trigger the event
Every COMBO switch in the level with the same Hitag must be in the state specified by their TAG3 before the event will be triggered. This switch can trigger any event. COMBO switches make sound automatically, without a SOUND_SPOT ST1 (Hitag 134).	

<b>Lotag 214</b>	<b>TAG_COMBO_SWITCH_EVERYTHING_ONCE</b>
------------------	---

Hitag	match tag
TAG3	if set to: 0 = switch must be OFF to trigger the event 1 = switch must be ON to trigger the event
Every COMBO switch in the level with the same Hitag must be in the state specified by their TAG3 before the event will be triggered. This switch can trigger any event. COMBO switches make sound automatically, without a SOUND_SPOT ST1 (Hitag 134). Once the combo is properly triggered, it can never be triggered again.	

<b>Lotag 216</b>	<b>TAG_SPAWN_ACTOR_SWITCH</b>
------------------	-------------------------------

Hitag = match tag

This switch can only be used to spawn actors with a Lotag of 203. The actors and the switch must have the same hitag.

### **Lotag 304 TAG\_ROTATE\_SO\_SWITCH**

Hitag = number of the Sector Object to rotate

Operating this switch causes a Sector Object to rotate smoothly clockwise 90 degrees. We never found a place to use this in Shadow Warrior, but it's a pretty cool feature nevertheless. This switch has an automatic sound associated with it, and is demonstrated in EXAMPLE.MAP, room FD.

#### 9.III.1) Shootable Switch

(Tile #577)

Switch tile number 577 is unique, because it can be shot by the player to activate and deactivate. Don't match this switch to a locked door, though, because the door can be activated by shooting the switch, despite its locked status.

#### 9.III.2) Exit Switch

(Tile #2470)

An exit switch causes the level to end when the player operates it, and the bonus or frag screen is displayed. Any switch can be tagged as an exit switch, but tile 2470 is the graphic that Shadow Warrior usually uses. An exit switch must have a lotag of 116. The hitag specifies which level to go to next, but this only works if the map has the same filename as one of the official Shadow Warrior game maps. For a standard external map, a hitag of 0 will suffice.

## **9.IV) Keys and Keyholes**

#### 9.IV.1) Keys

(Tile #1765-1779)

These are the keys used to unlock doors (ignore the little dog sprites.) You can mix and match key types in a level, but be aware that some keys will overwrite the others in the status bar when the player picks it up. A gold master key will overwrite a red keycard, a silver master key will overwrite a blue keycard, a bronze master key will overwrite a green keycard, and a red master key will overwrite a yellow keycard.

#### 9.IV.2) Keyholes

(Tile #1846, 1850, 1852)

These are the sprites the player operates to unlock a door with a key. Always set the lotag to 29. Set the hitag equal to the lotag of the SECT\_LOCK\_DOOR (hitag 29) ST1. The BUILD editor will automatically set the palette of the keyhole when you re-enter 3D mode from 2D mode. Use the correct style of keyhole to match the desired key.

## **9.V) TV Monitors**

(Tile #3830-3837)

These sprites are used in conjunction with mirrors and MIRROR\_CAM ST1 (hitag 1000). Rotate the monitor sprite flat and place it in front of the mirror, facing out away from the wall. Give it a hitag equal to Lotag (TAG2) of the MIRROR\_CAM. This is only necessary for TV monitors, not for magic monitors. See EXAMPLE.MAP for a demonstration of how to build this.

## **9.VI) QuickJump and QuickJumpDown**

### 9.VI.1) QuickJump

(Tile #2309)

This is a special sector effector. When an actor steps on the QJ sprite, he will jump in the direction the sprite is pointing. This is a good way to get actors to jump up onto ledges, and it looks pretty cool. Only jumping actors (ninjas, bunnies and rippers) are affected by this.

### 9.VI.2) QuickJump

(Tile #2310)

This is another special sector effector, similar to QuickJump. When an actor steps on the QJD sprite, he will jump down off a ledge in the direction the sprite is pointing. Make ninjas leap down on the player from high ledges. Only jumping actors (ninjas, bunnies and rippers) are affected by this.

## **9.VII) Gratings**

Gratings can be tagged so the player can operate them by pressing the SPACEBAR on them. Point the grating sprite in the direction you want it to move. The lotag must be 200. The hitag is the distance the sprite will move, 1024 is a good distance. This feature may have other applications.

## 9.VIII) *DeathFlag(tm) Capture the Flag*

Shadow Warrior has a pretty cool Capture the Flag facility that we didn't really take advantage of in the game.

### 9.VIII.1) The Flag Sprite

(Tile #2520)

A level can have multiple flags of any type or color, allowing for a variety of possibilities, including up to eight teams.

*NOTE:* A flag will automatically tag the sector it's in with a sector hitag 9000, and a sector lotag equal to the flag's palette. You can create a scoring sector anywhere in the map in addition to the flag's sector just by using these tags.

Hitag	if set to: <ul style="list-style-type: none"> <li><b>0</b> = default, flag explodes if carried for longer than 30 seconds or if carrier dies. The flag respawns every 30 seconds.</li> <li><b>1003</b> = flag carrier must die or score before the flag will respawn back at the base.</li> </ul>
Lotag	match tag, when the carrier scores with this flag this match tag will be triggered. Reward a score with free items, trigger a sound, whatever.
TAG5	if non-zero, the match in TAG6 will be triggered when this flag is scored TAG5 times. Make a special reward happen every ten points or open a secret passage to the exit when the score reaches 25.
TAG6	secondary match tag, used together with TAG5
TAG12	set the palette (TAG12) of the flag to change its color (there is no default value): <ul style="list-style-type: none"> <li><b>16</b> = brown flag</li> <li><b>17</b> = gray flag</li> <li><b>18</b> = purple flag</li> <li><b>19</b> = red flag</li> <li><b>20</b> = yellow flag</li> <li><b>21</b> = olive flag</li> <li><b>22</b> = green flag</li> <li><b>23</b> = blue flag</li> </ul>
BOOL1	if set to: <ul style="list-style-type: none"> <li><b>0</b> = the score is incremented when the carrier enters the score sector</li> <li><b>1</b> = the score is not incremented when the carrier enters the score sector. This feature can be used for a flag that you just want to trigger things with, but no score. For example: a map might have a red base and a blue base, and a neutral yellow flag. Teams race to get the yellow flag which when returned to their base, causes the path to the opposing team's flag to open up.</li> </ul>

## 10) EXAMPLE.MAP Reference

by Keith Schuler, revised by Steffen Itterheim

### **EXAMPLE.MAP Reference Contents**

10.I) Explanation	85
10.II) Column A	85
AA. Sector Depth, Sector Current, Water Drips	
AB. Translucent Water (bound with area AD)	
AC. Room under Room (bound with area AE)	
AD. Under Translucent Water (bound with area AB)	
AE. Room over Room (bound with area AC)	
10.III) Column B	86
BA. Diving Water	
BB. Wall and Floor Texture Panning, Conveyor Belt	
BC. Ladders, Actor Block	
BD. Missile Traps	
BE. Parallax Level, Don't Copy Palette	
BF. Continous and Triggered Lava Eruption	
10.IV) Column C	87
CA. Visible/Invisible Crack Triggers, Exploding Ceiling/Floor	
CB. Copying Sectors	
CC. Spawning Various Debris	
CD. Floor and Sector Damage, Teleporters	
CE. Warp Copy Sprite, Ceiling/Floor Teleporters, Slope Don't Draw, Floor Depth	
10.V) Column D	87
DA. Dentable Walls	
DB. Vators (Doors, Stompers, Lifts)	
DC. Spikes (Dynamic Slopes, Ramps)	
DD. Lighting Effects, Colored Light, Shootable Lights, Door Open/Light On	
DE. Earthquake, Breakable Masked and Solid Walls, Breakable/Unbreakable Sprites	
DF. Slide, Deleting Sprites, Ladder	
10.VI) Column E	88
EA. Rotating Doors, Masked and Solid Sliding Doors, Combo Switch	
EB. Changing Sector Attributes, Spawning Items, Locked Doors, Sound Shutoff	
EC. Sector Objects, Sprite Sector Objects, Amobea, Ambient Sound	
ED. Tornado, Ambient Sound	
EE. Scaleable and Destructible Sector Objects, Driveable Turrets, Auto Turrets	
EF. Driveable, Remote Driveable	
10.VII) Column F	90
FA. Actors on Tracks, Sector Objects on Tracks	
FB. Binding SOs over Dive Sectors, SO Waiting for Events, SO not rotating (bound with area FF)	
FC. Binding Sector Objects under Rooms (bound with area FE)	
FD. Mirrors, Demo Cameras, Speed Affecting Wall Tags	
FE. Binding Sector Objects over Rooms (bound with area FC)	
FF. Binding Sector Objects under Dive Sectors (bound with area FB)	
10.VIII) Column G	91
GA. Sine Wave Floors and Ceilings	
GB. Sine Wave Walls, Secret Areas, TV Cameras	

**10.I) Explanation**

EXAMPLE.MAP demonstrates most of the map functionality in Shadow Warrior, usually in the simplest possible way. For more advanced examples, look in the game maps. This reference describes what can be found in EXAMPLE.MAP and where. The map is divided into several similar rooms arranged in a grid-like fashion. Columns are labeled by blue letters, while rows are labeled by red letters. For example, to find room BC, you would start in room AA (upper left corner) then look two rooms to the right and three rooms down.

**10.II) Column A**

AA. Sector Depth, Sector Current, Water Drips TAGS: 0 SECT_SINK 3 SECT_CURRENT 1006 DRIP_GENERATOR
AB. Translucent Water (bound with area AD) TAGS: 98 FLOOR_Z_ADJUST 111 VIEW_LEVEL2 121 VIEW_THRU_FLOOR 136 CEILING_FLOOR_PIC_OVERRIDE 202 BOUND_FLOOR_BASE_OFFSET
AC. Room under Room (bound with area AE) TAGS: 110 VIEW_LEVEL1 120 VIEW_THRU_CEILING 202 BOUND_FLOOR_BASE_OFFSET
AD. Under Translucent Water (bound with area AB) TAGS: 8 SECT_UNDERWATER 110 VIEW_LEVEL1 120 VIEW_THRU_CEILING 136 CEILING_FLOOR_PIC_OVERRIDE 203 BOUND_FLOOR_OFFSET
AE. Room over Room (bound with area AC) TAGS: 111 VIEW_LEVEL2 121 VIEW_THRU_FLOOR 203 BOUND_FLOOR_OFFSET

## 10.III) Column B

<b>BA. Diving Water</b>
<i>TAGS:</i> 7 SECT_DIVE_AREA 8 SECT_UNDERWATER 9 SECT_UNDERWATER2
<b>BB. Wall and Floor Texture Panning, Conveyor Belt</b>
<i>TAGS:</i> 0 SECT_SINK 3 SECT_CURRENT 19 SECT_FLOOR_PAN 21 SECT_CEILING_PAN 23 WALL_PAN_SPEED 148 WALL_DONT_STICK
<b>BC. Ladders, Actor Block</b>
<i>TAGS:</i> 30 SPRI_CLIMB_MARKER 38 SECT_ACTOR_BLOCK
<i>WALL TAGS:</i> 305 TAG_WALL_CLIMB
<b>BD. Missile Traps</b>
<i>TAGS:</i> 43 FIREBALL_TRAP 44 BOLT_TRAP 63 SPEAR_TRAP
<i>SECTOR TAGS:</i> 43 TAG_TRIGGER_MISSILE_TRAP
<b>BE. Parallax Level, Don't Copy Palette</b>
<i>TAGS:</i> 46 PARALLAX_LEVEL 47 SECT_DONT_COPY_PALETTE
<b>BF. Continuous and Triggered Lava Eruption</b>
<i>TAGS:</i> 56 LAVA_ERUPT 140 TRIGGER_SECTOR

## 10.IV) Column C

CA. Visible/Invisible Crack Triggers, Exploding Ceiling/Floor
TAGS: 57 SECT_EXPLODING_CEIL_FLOOR 257 TAG_SPRITE_HIT_MATCH
CB. Copying Sectors
TAGS: 58 SECT_COPY_DEST 59 SECT_COPY_SOURCE 211 TAG_SWITCH_EVERYTHING
CC. Spawning Various Debris
TAGS: 69 SPAWN_SPOT 211 TAG_SWITCH_EVERYTHING
CD. Floor and Sector Damage, Teleporters
TAGS: 82 SECT_DAMAGE 84 WARP_TELEPORTER
CE. Warp Copy Sprite, Ceiling/Floor Teleporters, Slope Don't Draw, Floor Depth
TAGS: 85 WARP_CEILING_PLANE 86 WARP_FLOOR_PLANE 87 WARP_COPY_SPRITE1 88 WARP_COPY_SPRITE2 98 FLOOR_Z_ADJUST 99 FLOOR_SLOPE_DONT_DRAW

## 10.V) Column D

DA. Dentable Walls
TAGS: 103 SECT_WALL_MOVE 104 SECT_WALL_MOVE_CANSEE
DB. Vators (Doors, Stompers, Lifts)
TAGS: 92 SECT_VATOR 134 SOUND_SPOT 206 TAG_SPRITE_SWITCH_VATOR
DC. Spikes (Dynamic Slopes, Ramps)
TAGS: 106 SECT_SPIKE 206 TAG_SPRITE_SWITCH_VATOR



DD. Lighting Effects, Colored Light, Shootable Lights, Door Open/Light On

*TAGS:*

108 LIGHTING  
211 TAG\_SWITCH\_EVERYTHING

DE. Earthquake, Breakable Masked and Solid Wall, Breakable/Unbreakable Sprites

*TAGS:*

69 SPAWN\_SPOT  
132 BREAKABLE\_SPRITE  
133 QUAKE\_SPOT

*WALL TAGS:*

307 TAG\_WALL\_BREAK

*SECTOR TAGS:*

211 TAG\_TRIGGER\_EVERYTHING

DF. Slide, Deleting Sprites, Ladder

*TAGS:*

30 SPRI\_CLIMB\_MARKER  
135 SLIDE\_SECTOR  
141 DELETE\_SPRITE  
211 TAG\_SWITCH\_EVERYTHING

*WALL TAGS:*

305 TAG\_WALL\_CLIMB

## 10.VI) Column E

EA. Rotating Doors, Masked and Solid Sliding Doors, Combo Switch

*TAGS:*

143 SECT\_ROTATOR  
144 SECT\_ROTATOR\_PIVOT  
145 SECT\_SLIDOR  
213 TAG\_COMBO\_SWITCH\_EVERYTHING

EB. Changing Sector Attributes, Spawning Items, Locked Doors, Sound Shutoff

*TAGS:*

29 SECT\_LOCK\_DOOR  
69 SPAWN\_SPOT  
134 SOUND\_SPOT  
146 SECT\_CHANGOR  
148 WALL\_DONT\_STICK  
149 SPAWN\_ITEMS  
150 STOP\_SOUND\_SPOT  
257 TAG\_SPRITE\_HIT\_MATCH

*SECTOR TAGS:*

212 TAG\_TRIGGER\_EVERYTHING\_ONCE

EC. Sector Objects, Sprite Sector Objects, Amoeba, Ambient Sound

*TAGS:*

5 SECT\_NO\_RIDE  
31 SECT\_SO\_SPRITE\_OBJ  
50 SO\_SPIN  
51 SO\_SPIN\_REVERSE

<p>75 SO_AMOEBA  500-600 SECT_SO_BOUNDING  1002 AMBIENT_SOUND  <i>WALL TAGS:</i>  504 TAG_WALL_LOOP_OUTER  <i>SECTOR TAGS:</i>  500-600 TAG_OBJECT_CENTER</p>
<p>ED. Tornado, Ambient Sound  <i>TAGS:</i>  72 SO_SLOPE_CEILING_TO_POINT  73 SO_TORNADO  500-600 SECT_SO_BOUNDING  1002 AMBIENT_SOUND  <i>WALL TAGS:</i>  504 TAG_WALL_LOOP_OUTER  550 TAG_WALL_ALIGN_SLOPE_TO_POINT  <i>SECTOR TAGS:</i>  500-600 TAG_OBJECT_CENTER</p>
<p>EE. Scaleable and Destructible Sector Objects, Driveable Turrets, Auto Turrets  <i>TAGS:</i>  1 SECT_OPERATIONAL  16 SO_ANGLE  62 SHOOT_POINT  68 SO_LIMIT_TURN  70 SO_MATCH_EVENT  76 SO_MAX_DAMAGE  80 SO_KILLABLE  81 SO_AUTO_TURRET  100 SO_SCALE_INFO  500-600 SECT_SO_BOUNDING  <i>WALL TAGS:</i>  504 TAG_WALL_LOOP_OUTER  <i>SECTOR TAGS:</i>  500-600 TAG_OBJECT_CENTER</p>
<p>EF. Driveable, Remote Driveable  <i>TAGS:</i>  1 SECT_OPERATIONAL  16 SO_ANGLE  62 SHOOT_POINT  140 TRIGGER_SECTOR  147 SO_DRIVEABLE_ATTRIB  500-600 SECT_SO_BOUNDING  <i>WALL TAGS:</i>  504 TAG_WALL_LOOP_OUTER  <i>SECTOR TAGS:</i>  500 TAG_OBJECT_CENTER</p>

## 10.VII) Column F

### FA. Actors on Tracks, Sector Objects on Tracks

*TAGS:*

16 SO\_ANGLE  
55 SO\_TURN\_SPEED  
500-600 SECT\_SO\_BOUNDING

*TRACK TAGS:*

700 TRACK\_START  
701 TRACK\_END  
702 TRACK\_SET\_SPEED  
782 TRACK\_ACTOR\_ATTACK1

*WALL TAGS:*

504 TAG\_WALL\_LOOP\_OUTER

*SECTOR TAGS:*

500-600 TAG\_OBJECT\_CENTER

### FB. Binding SOs over Dive Sectors, SOs waiting for events, SO not rotating (bound with area FF)

*TAGS:*

0 SECT\_SINK  
7 SECT\_DIVE\_AREA  
45 SECT\_SO\_DONT\_ROTATE  
202 BOUND\_FLOOR\_BASE\_OFFSET  
211 TAG\_SWITCH\_EVERYTHING  
500-600 TAG\_OBJECT\_CENTER

*TRACK TAGS:*

700 TRACK\_START  
701 TRACK\_END  
702 TRACK\_SET\_SPEED  
726 TRACK\_WAIT\_FOR\_EVENT

*WALL TAGS:*

504 TAG\_WALL\_LOOP\_OUTER

*SECTOR TAGS:*

500-600 TAG\_OBJECT\_CENTER

### FC. Binding Sector Objects under Rooms (bound with area FE)

*TAGS:*

5 SECT\_NO\_RIDE  
30 SPRI\_CLIMB\_MARKER  
75 SO\_AMOEBA  
110 VIEW\_LEVEL1  
120 VIEW\_THRU\_CEILING  
202 BOUND\_FLOOR\_BASE\_OFFSET  
500-600 TAG\_OBJECT\_CENTER

*TRACK TAGS:*

700 TRACK\_START  
701 TRACK\_END

*WALL TAGS:*

305 TAG\_WALL\_CLIMB  
504 TAG\_WALL\_LOOP\_OUTER

*SECTOR TAGS:*

500-600 TAG OBJECT CENTER

FD. Mirrors, Demo Cameras, Speed Affecting Wall Tags

*TAGS:*

5 SECT\_NO\_RIDE  
66 DEMO\_CAMERA  
304 TAG\_ROTATE\_SO\_SWITCH  
500-600 TAG\_OBJECT\_CENTER

*WALL TAGS:*

500 TAG\_WALL\_LOOP\_DONT\_SPIN  
503 TAG\_WALL\_LOOP\_SPIN\_4X  
504 TAG\_WALL\_LOOP\_OUTER

*SECTOR TAGS:*

500-600 TAG\_OBJECT\_CENTER

FE. Binding Sector Objects over Rooms (bound with area FC)

*TAGS:*

5 SECT\_NO\_RIDE  
111 VIEW\_LEVEL2  
121 VIEW\_THRU\_FLOOR  
203 BOUND\_FLOOR\_OFFSET

*WALL TAGS:*

305 TAG\_WALL\_CLIMB  
504 TAG\_WALL\_LOOP\_OUTER

FF. Binding Sector Objects under Dive Sectors (bound with area FB)

*TAGS:*

5 SECT\_NO\_RIDE  
8 SECT\_UNDERWATER  
203 BOUND\_FLOOR\_OFFSET

*WALL TAGS:*

504 TAG\_WALL\_LOOP\_OUTER

## 10.VIII) Column G

GA. Sine Wave Floors and Ceilings

*SECTOR TAGS:*

400 TAG\_SINE\_WAVE\_FLOOR  
420 TAG\_SINE\_WAVE\_CEILING  
440 TAG\_SINE\_WAVE\_BOTH

GB. Sine Wave Walls, Secret Areas, TV Cameras

*TAGS:*

1000 MIRROR\_CAM

*WALL TAGS:*

300 TAG\_WALL\_SINE\_Y\_BEGIN  
301 TAG\_WALL\_SINE\_X\_BEGIN  
302 TAG\_WALL\_SINE\_Y\_END  
303 TAG\_WALL\_SINE\_X\_END  
306 TV\_CAMERA\_SCREEN

*SECTOR TAGS:*

217 TAG\_SECRET\_AREA\_TRIGGER

GC. Exit Switch and Triggers, Actor Spawn Triggers, Magic Cameras, Activate Wall

Sound

*TAGS:*

116 TAG\_LEVEL\_EXIT\_SWITCH

203 TAG\_SPAWN\_ACTOR

1000 MIRROR\_CAM

*WALL TAGS:*

306 TV\_CAMERA\_SCREEN

900 TAG\_WALL\_PLAY\_SOUND (hitag)

*SECTOR TAGS:*

116 TAG\_LEVEL\_EXIT\_TRIGGER

216 TAG\_SPAWN\_ACTOR\_TRIGGER