

Lite-C FLAGS at a glance (revised for Version 7.60)

Note : Do a search for “flags” or “flag”. In the result column you will get some 158 topics.

Then, from these, use the “search for” column.

I’ve used the search section to get most of the information(in search). On a few occasions I used the index (in index)

The best trail to follow is : **Contents – Lite-C Language – ...**

WHO in Lite C (A...Z)	ON == set	toggle	OFF == reset	SEARCH FOR
	= flag	^= flag	&= ~flag	Migration to...
	set(my,ZNEAR);	toggle(my,ZNEAR);	reset(my,ZNEAR);	Macros

	flags, flags2, eflags, emask, and smask	Migration to...
--	---	-----------------

Buttons, digits	buttons, digits and so on - 'inherit' the flags from the panel		flags (panel)
CAST	set(my,SHADOW CAST);		CAST
CLIPPED (is it ?)	if (!(my.eflags&CLIPPED)) { emit_particles(); }		CLIPPED
collision detection	my.eflags = NARROW;	my.eflags &= ~FAT;	FAT, NARROW
c_move (available flags)	IGNORE_YOU IGNORE_FLAG2 IGNORE_PASSABLE IGNORE_PASSENTS IGNORE_WORLD IGNORE_MAPS IGNORE_MODELS IGNORE_SPRITES IGNORE_PUSH IGNORE_CONTENT ACTIVATE_TRIGGER ACTIVATE_PUSH ACTIVATE_SHOOT ACTIVATE_SONAR USE_AABB GLIDE		c_move (in index and in search)
c_trace	IGNORE_ME IGNORE_YOU IGNORE_FLAG2 IGNORE_PASSABLE IGNORE_PASSENTS IGNORE_WORLD IGNORE_MAPS IGNORE_MODELS IGNORE_SPRITES IGNORE_PUSH IGNORE_CONTENT USE_POLYGON USE_BOX USE_AABB ACTIVATE_SHOOT ACTIVATE_SONAR SCAN_TEXTURE		c_trace (in index and in search)
DECAL (sprite)	set(my, DECAL);		DECAL
DYNAMIC	my.emask = DYNAMIC;	my.emask &= ~DYNAMIC;	DYNAMIC

Entities	Look under: Contents -- Lite-C Language -- Engine Objects -- ENTITY There are four important sections under this title.		
Entity flags (some available flags)	INVISIBLE UNLIT TRANSLUCENT OVERLAY BRIGHT NOFILTER NOFOG SHADOW CAST PASSABLE POLYGON <i>for a full list look under : Content – Lite-C Language – Engine Objects – ENTITY – ...</i>		
ENTITY* name = {}	ENTITY* gun { ... flags = BRIGHT PASSABLE; ... }		gun ...flags
Entity	my.flags = (BRIGHT PASSABLE INVISIBLE); set(mypanel, VISIBLE LIGHT ZNEAR);	my.flags &= ~INVISIBLE ;	Type field
	set (my, INVISIBLE); eName.flags = INVISIBLE;	eName.flags &= ~INVISIBLE;	
	if ((my.flags) & (~VISIBLE)) {...}		
ent_createlayer	flags2 combination : SKY DOME CUBE CYLINDER SCENE		ent_createlayer
ENABLE_... flag	To make an entity sensitive for a certain kind of event , the corresponding ENABLE_... flag of the emask parameter is set.		emask....event
Action name {...}	{ my.emask = (ENABLE_BLOCK ENABLE_ENTITY); }		
	{ my.emask = (ENABLE_SCAN ENABLE_CLICK ENABLE_RIGHTCLICK); }		emask. (in search)
event_type	{ my.emask = ENABLE_CLICK; }	{ my.emask &= ~ENABLE_CLICK; } EVENT_CLICK
EVENT_TRIGGER	{ my.emask = ENABLE_TRIGGER; }	{ my.emask &= ~ENABLE_TRIGGER; }	emask. (in search)
	{ my.emask = ENABLE_RELEASE; }	{ my.emask &= ~ENABLE_RELEASE; }	event (in index)
	{ my.emask = ENABLE_TOUCH; }	{ my.emask &= ~ENABLE_TOUCH; } EVENT_....
	{ my.emask = (ENABLE_BLOCK ENABLE_ENTITY); }		event (in index)
event_type	if (event_type == EVENT_TOUCH); if (event_type == EVENT_CLICK);		

FLAG1...FLAG8	if is(my,FLAG1) { reset(my,FLAG1); my.skill1 += 1; }		FLAG1...FLAG8
light	set(my,LIGHT);		flag ... LIGHT
LODO LOD1	if (!(my.eflags&(LOD1 LOD0))) { emit_particles(); }		LOD0, LOD1
MATERIAL	TANGENT TRANSLUCENT PASS_SOLID AUTORELOAD OVERRIDE ENABLE_RENDER ENABLE_VIEW ENABLE_TREE		flags...flags(tangent)
MATERIAL* mtlname	{flags = ENABLE_RENDER;}		Enable_render...event
	mtlname.flags = TANGENT;	material.flags &= ~TANGENT;	flags...flags...
NOFILTER	set(my, NOFILTER); // for texture		NOFILTER
PANEL* pname = {}	{ flags = VISIBLE OVERLAY TRANSLUCENT FILTER LIGHT CENTER_X CENTER_Y;}		flags (panels)
panel.flags	panel.flags = VISIBLE;	panel.flags &= ~VISIBLE;	flags (panels)
	pMyPanel.flags = (VISIBLE LIGHT ZNEAR);		flags and znear
	set(mypanel, VISIBLE LIGHT ZNEAR);		flags and znear
The PARTICLE struct is defined in include\atypes.h. Look under Contents – Lite-C Language – Engine Objects – PARTICLE			
particle functions	x, y, z, red, green, blue, alpha , TRANSLUCENT, BRIGHT, OVERLAY, NOFILTER, BEAM/STREAK, MOVE, bmap, material, event, lifespan, size, gravity, vel_x, vel_y, vel_z, skill_a...skill_d, skill_x..., skill_z, FLAG1..FLAG8, for example see effect.		
PARTICLE *p	p.flags = (BRIGHT TRANSLUCENT MOVE BEAM STREAK);		PARTICLE EFFECT
	set(p,TRANSLUCENT BEAM BRIGHT MOVE);		Particle effect...move
POLYGON	set(my,POLYGON);		entity.POLYGON
SKY	use next lines		SKY...entities
ENTITY* skycube{}	{ flags2 = SKY CUBE VISIBLE; }	my.flags2 = UNTOUCHABLE	SKY
	ENTITY* eSky = { type = "clouds+2.tga"; flags2 = SKY DOME VISIBLE; }		migrating to...

sky_blend			SKY_BLEND
SEND	smask		flags...NOSEND
	my.smask = NOSEND_ALPHA NOSEND_AMBIENT;		nosend
SPOTLIGHT	set(my,SPOTLIGHT);		SPOTLIGHT(index)
TEXT	VISIBLE TRANSLUCENT FILTER CENTER_X CENTER_Y ARIGHT WWRAP CONDENSED SHADOW OUTLINE		flags...flags(text)
TEXT* name = {}	{ flags = CENTER_X TRANSLUCENT VISIBLE; }		flags...flags(text)
UNLIT	set (my, UNLIT);		UNLIT
VIEW	VISIBLE AUDIBLE ISOMETRIC UNTOUCHABLE PROCESS_SCREEN PROCESS_TARGET CHILD TRANSLUCENT PORTALCLIP CULL_CW NOCULL NOSHADOW SHADOW NOPARTICLE NOSHADER NOFLAG1 NOLOD		flags ... flags(view)
VIEW* name = {}	{ flags = VISIBLE AUDIBLE; }		flags...flags(view)
ENTITY* viewname	gunmodel.flags2 = VISIBLE;		SHOTGUN
<i>extra information</i>			
flags2 are for	my.flags2 = UNTOUCHABLE;		flags2...untouchable
flags2 are used only in Lite-C	my.flags2 = SPOTLIGHT;		flags2...Spotlight
	gunmodel.flags2 = VISIBLE;		flags2...Visible
	ENTITY* skyMountain = { flags2 = SCENE VISIBLE; }		flags2...Scene
	ENTITY* skyClouds = { flags2 = DOME VISIBLE; }		flags2...Dome
	ent_createlayer(String* filename, var flags2,var layer);		flags2...ent_createlayer
	{ flags2 = CUBE VISIBLE;}		flags2...cube