

Lite-C FLAGS at a glance (revised for Version 7.70)

Note : Do a search for “flags” or “flag”. In the result column you will get some 152 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 – Game Programming – Engine Objects ...**

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	flag.....flags
--	---	----------------

Buttons, digits	buttons, digits - '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 Engine	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 -- Game Programming -- Engine Objects -- ENTITY		
Entity flags (some available flags)	INVISIBLE UNLIT TRANSLUCENT OVERLAY BRIGHT NOFILTER NOFOG SHADOW CAST PASSABLE POLYGON <i>for a full list look under : Content – Game Programming – Engine Objects – ENTITY</i>		
ENTITY* name = {}	ENTITY* gun { ... flags = BRIGHT PASSABLE; ... }		gun ...flags
Entity (level entities)	my.flags = (BRIGHT PASSABLE INVISIBLE);	my.flags &= ~INVISIBLE ;	invisible
	set (my, INVISIBLE); eName.flags = INVISIBLE;	eName.flags &= ~INVISIBLE;	
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. (The search list found 23 items and 15 are EVENT...)		emask....(in search)
	Look under: Contents -- Game Programming -- Engine Objects -- ENTITY –EVENTS		
Action name {...}	{ my.emask = (ENABLE_BLOCK ENABLE_ENTITY); }		event
	{ 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);		
event_type	switch (event_type) {...case EVENT_BLOCK:....}		event (in index)

FLAG1...FLAG8	if is(my,FLAG1) { reset(my,FLAG1); my.skill1 += 1; }		FLAG1...FLAG8
light	set(my,LIGHT);		flag ... LIGHT
LOD0, LOD1	if (!(my.eflags&(LOD1 LOD0))) { emit_particles(); }		LOD0, LOD1
MATERIAL	TANGENT TRANSLUCENT PASS_SOLID AUTORELOAD OVERRIDE ENABLE_RENDER ENABLE_TREE ENABLE_VIEW		MATERIAL...flags (in search)
	Look under Contents – Game Programming – Engine objects – MATERIAL – Material Parameters – flags (material)		
MATERIAL* mtlname	{ flags = ENABLE_RENDER; }		Enable_render...event
	mtlname.flags = TANGENT;	material.flags &= ~TANGENT;	
NOFILTER	set(my, NOFILTER); // for texture		NOFILTER
PANEL* pname = {}	{ flags = SHOW OVERLAY TRANSLUCENT FILTER LIGHT CENTER_X CENTER_Y; }		flags (panels)
	Look under Contents – Game Programming – Engine objects – PANEL – Panel properties – flags (panels)		
panel.flags	panel.flags = SHOW;	panel.flags &= ~SHOW;	flags (panels)
	pMyPanel.flags = (SHOW LIGHT);		
	set(mypanel, SHOW LIGHT);		
The PARTICLE struct is defined in include\atypes.h. Look under Contents – Game Programming – Engine Objects – PARTICLE			
particle functions <i>for examples see effect.</i>	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,		
PARTICLE *p	p.flags = (BRIGHT MOVE);		PARTICLE... effect
	set(p,TRANSLUCENT BEAM BRIGHT MOVE);		Particle effect...move
POLYGON	set(my,POLYGON);		entity.POLYGON

SKY	look under : Content – Game Programming – ENTITY – Sky parameters		SKY...entities
ENTITY* skycube{}	{ flags2 = SKY CUBE SHOW; }	my.flags2 = UNTOUCHABLE	SKY...entities
	ENTITY* eSky = { type = "clouds+2.tga"; flags2 = SKY DOME SHOW; }		migrating to...
SEND / NOSEND	my.smask = NOSEND_ALPHA NOSEND_AMBIENT;		flags...NOSEND
SPOTLIGHT	my.flags = SPOTLIGHT; (see updating to 7.7)		SPOTLIGHT(index)
TEXT	SHOW TRANSLUCENT FILTER CENTER_X CENTER_Y ARIGHT WWRAP CONDENSED SHADOW OUTLINE		flags...flags(text)
TEXT* name = {}	{ flags = CENTER_X TRANSLUCENT SHOW; }		flags...flags(text)
UNLIT	set (my, UNLIT);		UNLIT
VIEW	SHOW 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 = SHOW AUDIBLE; }		flags....flags(view)
<i>extra information</i>	my.flags = SPOTLIGHT; (See updating to 7.7)		flags...spotlight
flags2 are for	flags2 aren't updated over the network		
ENTITY* viewname	{ type = "shotgun.mdl"; flags2 = SHOW; }		Shotgun ...entities
flags2 are used only in Lite-C	my.flags2 = UNTOUCHABLE;		flags2...untouchable
	gunmodel.flags2 = SHOW;		flags2...visible
	ENTITY* skyMountain = { flags2 = SCENE SHOW; }		flags2...scene
	ENTITY* skyClouds = { flags2 = DOME SHOW; }		flags2...dome
	ent_createlayer(String* filename, var flags2,var layer); // this is definition at the top		flags2...ent_createlayer
	{ flags2 = CUBE SHOW;}		flags2...cube