

Instruction Name	Length	Example / Info	History
{00} Nop	01	<pre>00 typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x00 } Nop; // Nop This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</pre>	08-08-2023 Newly Added
{01} Evt_end	02	<pre>01 00 typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x01 UCHAR zAlign; // 0x01 // Always Zero } Evt_end; // End Current Script This bytecode ends the current Main/Sub script</pre>	08-08-2023 Newly Added
{02} Evt_next	01	<pre>02 typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x02 } Evt_next; // This bytecode</pre>	08-08-2023 Newly Added
{03} Evt_chain	02	<pre>03 ?? typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x03 UCHAR Data1; // 0x01 } Evt_chain; This bytecode</pre>	08-08-2023 Newly Added
{04} Evt_exec	04	<pre>04 ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x04 UCHAR Data1; // 0x01 UCHAR Data2; // 0x02 UCHAR Data3; // 0x03 } Evt_exec; This bytecode</pre>	08-08-2023 Newly Added
{05} Evt_kill	02	<pre>05 ?? typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x05 UCHAR Data1; // 0x01 } Evt_kill; This bytecode</pre>	08-08-2023 Newly Added

Instruction Name	Length	Example / Info	History
{06} Ifel_ck	04	<p>06 ?? ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x06 UCHAR zAlign; // 0x01 USHORT data2; // 0x02 } Ifel_ck; This bytecode</pre>	08-08-2023 Newly Added
{07} Else_ck	04	<p>07 ?? ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x07 UCHAR zAlign; // 0x01 USHORT data2; // 0x02 } Else_ck; This bytecode</pre>	08-08-2023 Newly Added
{08} Endif	02	<p>08 ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x08 UCHAR zAlign; // 0x01 } Endif; This bytecode</pre>	08-08-2023 Newly Added
{09} Sleep	04	<p>09 ?? ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x09 UCHAR Data1; // 0x01 USHORT data2; // 0x02 } Sleep; This bytecode</pre>	08-08-2023 Newly Added
{0A} Sleeping	03	<p>0A ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; USHORT data2; } Sleeping; This bytecode</pre>	08-08-2023 Newly Added
{0B} Wsleep	01	<p>0B</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; } Wsleep; This bytecode</pre>	08-08-2023 Newly Added

Instruction Name	Length	Example / Info	History
{0C} Wsleeping	01	<p>0C</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; } Wsleeping; This bytecode</pre>	08-08-2023 Newly Added
{0D} For	06	<p>0D ?? ?? ?? ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign; SHORT data2; USHORT data4; } For; This bytecode</pre>	08-08-2023 Newly Added
{0E} Next	02	<p>0E ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign; } Next; This bytecode</pre>	08-08-2023 Newly Added
{0F} While	04	<p>0F ?? ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign; SHORT data1; } While; This bytecode</pre>	08-08-2023 Newly Added
{10} Ewhile	02	<p>10 ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data1; } Ewhile; This bytecode</pre>	08-08-2023 Newly Added
{11} Do	04	<p>11 ?? ?? ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign; SHORT data2; } Do; This bytecode</pre>	08-08-2023 Newly Added
{12} Edwhile	02	<p>12 ??</p> <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data1; } Edwhile; This bytecode</pre>	08-08-2023 Newly Added

Instruction Name	Length	Example / Info	History
{13} Switch	04	<pre>13 ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data1; USHORT data2; } Switch; This bytecode</pre>	08-08-2023 Newly Added
{14} Case	06	<pre>14 ?? ?? ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data1; USHORT data2; } Case; This bytecode</pre>	08-08-2023 Newly Added
{15} Default	02	<pre>15 ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign; } Default; This bytecode</pre>	08-08-2023 Newly Added
{16} Eswitch	02	<pre>16 ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign; } Eswitch; This bytecode</pre>	08-08-2023 Newly Added
{17} Goto	06	<pre>17 ?? ?? ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data1; UCHAR data2; UCHAR zAlign; SHORT data4; } Goto; This bytecode</pre>	08-08-2023 Newly Added
{18} Gosub	02	<pre>18 ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data1; } Gosub; This bytecode</pre>	08-08-2023 Newly Added

Instruction Name	Length	Example / Info	History
{19} Return	02	19 ?? <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR data[3]; } Return; This bytecode</pre>	08-08-2023 Newly Added
{1A} Break	02	1A ?? <pre>typedef struct { // Ptr // Description UCHAR Opcode; CHAR data1; } Break; This bytecode</pre>	08-08-2023 Newly Added
{1B} For2	06	1B ?? ?? ?? ?? ?? <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR zAlign0; SHORT data2; UCHAR zAlign1; UCHAR data5; } For2; This bytecode</pre>	08-08-2023 Newly Added
{1C} Break_point	01	1C <pre>typedef struct { // Ptr // Description UCHAR Opcode; } Break_point; This bytecode</pre>	08-08-2023 Newly Added
{1D} Work_copy	04	1D ?? ?? ?? <pre>typedef struct { // Ptr // Description UCHAR Opcode; UCHAR Source; UCHAR Destination; UCHAR Typecast; } Work_copy; This bytecode</pre>	08-08-2023 Newly Added
{1E} Nop1E	01	1E <pre>typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x00 } Nop; // Nop This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</pre>	08-08-2023 Newly Added

Instruction Name	Length	Example / Info	History
{1F} Nop1F	01	<p>1F typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x00 } Nop; // Nop This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</p>	08-08-2023 Newly Added
{20} Nop	01	<p>20 typedef struct { // Ptr // Description UCHAR Opcode; // 0x00 // 0x00 } Nop; // Nop This bytecode is used for alignment of 1 byte opcodes and ending Elself blocks.</p>	08-08-2023 Newly Added
{21} Ck	04	<p>21 ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; // 0x21 UCHAR Flag; // System_flg, etc UCHAR Id; // Bit UCHAR OnOff; // 00 } Ck; This bytecode</p>	08-08-2023 Newly Added
{22} Set	04	<p>22 ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; // 0x22 UCHAR Flag; // System_flg, etc UCHAR Id; // Bit UCHAR OnOff; // 00 } Set; This bytecode</p>	08-08-2023 Newly Added
{23} Cmp	06	<p>23 ?? ?? ?? ?? ?? typedef struct { // Ptr // Description UCHAR zAlign; UCHAR Flag; UCHAR Operator; SHORT Value; } Cmp; This bytecode</p>	08-08-2023 Newly Added
{24} Save	04	<p>24 ?? ?? ?? typedef struct { // Ptr // Description UCHAR Opcode; UCHAR Destination; SHORT Source; } Save; This bytecode</p>	08-08-2023 Newly Added

{25} Copy	03	08-08-2023 Newly Added
{26} Calc	06	08-08-2023 Newly Added
{27} Calc2	04	08-08-2023 Newly Added
{28} Sce_rnd	01	08-08-2023 Newly Added
{29} Cut_chg	02	08-08-2023 Newly Added
{2A} Cut_old	01	08-08-2023 Newly Added
{2B} Message_on	06	08-08-2023 Newly Added
{2C} Aot_set	20	08-08-2023 Newly Added
{2D} Obj_model_set	38	08-08-2023 Newly Added
{2E} Work_set	03	08-08-2023 Newly Added
{2F} Speed_set	04	08-08-2023 Newly Added
{30} Add_speed	01	08-08-2023 Newly Added
{31} Add_aspeed	01	08-08-2023 Newly Added
{32} Pos_set	08	08-08-2023 Newly Added
{33} Dir_set	08	08-08-2023 Newly Added
{34} Member_set	04	08-08-2023 Newly Added
{35} Member_set2	03	08-08-2023 Newly Added
{36} Se_on	12	08-08-2023 Newly Added
{37} Sca_id_set	04	08-08-2023 Newly Added
{38} Flr_set	03	08-08-2023 Newly Added
{39} Dir_ck	08	08-08-2023 Newly Added
{3A} Sce_espr_on	16	08-08-2023 Newly Added
{3B} Door_aot_set	32	08-08-2023 Newly Added
{3C} Cut_auto	02	08-08-2023 Newly Added
{3D} Member_copy	03	08-08-2023 Newly Added
{3E} Member_cmp	06	08-08-2023 Newly Added
{3F} Plc_motion	04	08-08-2023 Newly Added
{40} Plc_dest	08	08-08-2023 Newly Added
{41} Plc_neck	10	08-08-2023 Newly Added
{42} Plc_ret	01	08-08-2023 Newly Added
{43} Plc_flg	04	08-08-2023 Newly Added
{44} Sce_em_set	22	08-08-2023 Newly Added
{45} Col_chg_set	05	08-08-2023 Newly Added
{46} Aot_reset	10	08-08-2023 Newly Added
{47} Aot_on	02	08-08-2023 Newly Added
{48} Super_set	16	08-08-2023 Newly Added
{49} Super_reset	08	08-08-2023 Newly Added
{4A} Plc_gun	02	08-08-2023 Newly Added
{4B} Cut_replace	03	08-08-2023 Newly Added
{4C} Sce_espr_kill	05	08-08-2023 Newly Added
{4D} Door_model_set	22	08-08-2023 Newly Added
{4E} Item_aot_set	22	08-08-2023 Newly Added
{4F} Sce_key_ck	04	08-08-2023 Newly Added
{50} Sce_trg_ck	04	08-08-2023 Newly Added
{51} Sce_bgm_control	06	08-08-2023 Newly Added
{52} Sce_espr_control	06	08-08-2023 Newly Added

{53}	Sce_fade_set	06	08-08-2023 Newly Added
{54}	Sce_espr3d_on	22	08-08-2023 Newly Added
{55}	Member_calc	06	08-08-2023 Newly Added
{56}	Member_calc2	04	08-08-2023 Newly Added
{57}	Sce_bgmtbl_set	08	08-08-2023 Newly Added
{58}	Plc_rot	04	08-08-2023 Newly Added
{59}	Xa_on	04	08-08-2023 Newly Added
{5A}	Weapon_chg	02	08-08-2023 Newly Added
{5B}	Plc_cnt	02	08-08-2023 Newly Added
{5C}	Sce_shake_on	03	08-08-2023 Newly Added
{5D}	Mizu_div_set	02	08-08-2023 Newly Added
{5E}	Keep_Item_ck	02	08-08-2023 Newly Added
{5F}	Xa_vol	02	08-08-2023 Newly Added
{60}	Kage_set	14	08-08-2023 Newly Added
{61}	Cut_be_set	04	08-08-2023 Newly Added
{62}	Sce_Item_lost	02	08-08-2023 Newly Added
{63}	Plc_gun_eff	01	08-08-2023 Newly Added
{64}	Sce_espr_on2	16	08-08-2023 Newly Added
{65}	Sce_espr_kill2	02	08-08-2023 Newly Added
{66}	Plc_stop	01	08-08-2023 Newly Added
{67}	Aot_set_4p	28	08-08-2023 Newly Added
{68}	Door_aot_set_4p	40	08-08-2023 Newly Added
{69}	Item_aot_set_4p	30	08-08-2023 Newly Added
{6A}	Light_pos_set	06	08-08-2023 Newly Added
{6B}	Light_kido_set	04	08-08-2023 Newly Added
{6C}	Rbj_reset	01	08-08-2023 Newly Added
{6D}	Sce_scr_move	04	08-08-2023 Newly Added
{6E}	Parts_set	06	08-08-2023 Newly Added
{6F}	Movie_on	02	08-08-2023 Newly Added
{70}	Splc_ret	01	08-08-2023 Newly Added
{71}	Splc_sce	01	08-08-2023 Newly Added
{72}	Super_on	16	08-08-2023 Newly Added
{73}	Mirror_set	08	08-08-2023 Newly Added
{74}	Sce_fade_adjust	04	08-08-2023 Newly Added
{75}	Sce_espr3d_on2	22	08-08-2023 Newly Added
{76}	Sce_Item_get	03	08-08-2023 Newly Added
{77}	Sce_line_start	04	08-08-2023 Newly Added
{78}	Sce_line_main	06	08-08-2023 Newly Added
{79}	Sce_line_end	01	08-08-2023 Newly Added
{7A}	Sce_parts_bomb	16	08-08-2023 Newly Added
{7B}	Sce_parts_down	16	08-08-2023 Newly Added
{7C}	Light_color_set	06	08-08-2023 Newly Added
{7D}	Light_pos_set2	06	08-08-2023 Newly Added
{7E}	Light_kido_set2	06	08-08-2023 Newly Added
{7F}	Light_color_set2	06	08-08-2023 Newly Added
{80}	Se_vol	02	08-08-2023 Newly Added

{81} Sce_Item_cmp	03	08-08-2023 Newly Added
{82} Sce_espr_task	03	08-08-2023 Newly Added
{83} Plc_heal	01	08-08-2023 Newly Added
{84} St_map_hint	02	08-08-2023 Newly Added
{85} Sce_em_pos_ck	06	08-08-2023 Newly Added
{86} Poison_ck	01	08-08-2023 Newly Added
{87} Poison_clr	01	08-08-2023 Newly Added
{88} Sce_Item_lost2	03	08-08-2023 Newly Added
{89} Evt_next2	01	08-08-2023 Newly Added
{8A} Vib_set0	06	08-08-2023 Newly Added
{8B} Vib_set1	06	08-08-2023 Newly Added
{8C} Vib_fade_set	08	08-08-2023 Newly Added
{8D} Item_aot_set2	24	08-08-2023 Newly Added
{8E} Sce_em_set2	24	08-08-2023 Newly Added

From:

<https://classicmodification.com/> - **Classic RE Modification**

Permanent link:

https://classicmodification.com/doku.php?id=re2_opcodes&rev=1691719693

Last update: **2023/08/10 19:08**

