Zinc

QUICK REFERENCE GUIDE

APPLICATION FRAMEWORK™

CONSTRUCTORS

```
int UI_APPLICATION::Main(void);
UI_BGI_DISPLAY(int driver = 0, int mode = 0);
UI_BIGNUM(void)
UI_BIGNUM(ibignum value);
UI_BIGNUM(rbignum value);
UI_BIGNUM(const char *string);
UI_BIGNUM(const UI_BIGNUM &number);
UI_DATE(void);
UI_DATE(const UI_DATE &date);
UI_DATE(int year, int month, int day);
UI_DATE(const char *string, DTF_FLAGS dtFlags = DTF_NO_FLAGS);
UI_DATE(int packedDate);
UI_ERROR_SYSTEM(void);
UI_EVENT_MANAGER(UI_DISPLAY *display, int noOfElements = 100);
UI_FG_DISPLAY(int mode = 0);
UI_GRAPHICS_DISPLAY(int mode = 4) .
UI_HELP_SYSTEM(char *fileName, UI_WINDOW_MANAGER *windowManager = NULL,
   UI_HELP_CONTEXT helpContext = NO_HELP_CONTEXT);
UI_MOTIF_DISPLAY(int *argc = NULL, char **argv = NULL, char *appClass = "ZincApp");
UI_MSC_DISPLAY(int mode = 0);
UI_MSWINDOWS_DISPLAY(HANDLE hInstance, HANDLE hPrevInstance, int nCmdShow);
UI_OS2_DISPLAY(void);
UI_STORAGE(void);
UI_STORAGE(const char *name, UIS_FLAGS pflags = UIS_READWRITE);
UI_STORAGE_OBJECT(void);
UI_STORAGE_OBJECT(UI_STORAGE &file, const char *name, OBJECTID nObjectID,
   UIS_FLAGS pflags = UIS_READWRITE);
UI_TEXT_DISPLAY(TDM_MODE mode = TDM_AUTO);
UI_TIME(void);
UI_TIME(const UI_TIME &time);
UI_TIME(int hour, int minute, int second = 0, int hundredth = 0);
UI_TIME(const char *string, TMF_FLAGS tmFlags = TMF_NO_FLAGS);
UI_TIME(int packedTime);
UID_CURSOR(DEVICE_STATE state = D_OFF, DEVICE_IMAGE image = DC_INSERT);
UID_KEYBOARD(DEVICE_STATE state = D_ON);
UID_PENDOS(void);
UID_MOUSE(DEVICE_STATE state = D_ON, DEVICE_IMAGE image = DM_WAIT);
UI_WINDOW_MANAGER(UI_DISPLAY *display, UI_EVENT_MANAGER *eventManager,
   EVENT_TYPE (*exitFunction)(UI_DISPLAY *display, UI_EVENT_MANAGER *eventManager,
   UI_WINDOW_MANAGER *windowManager) = NULL);
static UI_WINDOW_OBJECT *UI_WINDOW_OBJECT::New(const char *name, UI_STORAGE *file = NULL,
   UI_STORAGE_OBJECT *object = NULL);
```

```
UIW_BIGNUM(int left, int top, int width, UI_BIGNUM *value, const char *range = NULL,
   NMF_FLAGS nmFlags = NMF_NO_FLAGS, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);

UIW_BORDER(BDF_FLAGS bdFlags = BDF_NO_FLAGS);

UIW_BUTTON(int left, int top, int width, char *text, BTF_FLAGS btFlags = BTF_NO_TOGGLE|BTF_AUTO_SIZE,
   WOF_FLAGS woFlags = WOF_JUSTIFY_CENTER, USER_FUNCTION userFunction = NULL,
   EVENT_TYPE value = 0, char *bitmapName = NULL);

UIW_COMBO_BOX(int left, int top, int width, int height,
   int (*compareFunction)(void *element1, void *element2) = NULL,WNF_FLAGS wnFlags = WNF_NO_WRAP,
   WOF_FLAGS woFlags = WOF_NO_FLAGS, WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS);
UIW_COMBO_BOX(int left, int top, int width, int height,
   int (*compareFunction)(void *element1, void *element2), WOF_FLAGS flagSetting, UI_ITEM *item);

UIW_DATE(int left, int top, int width, UI_DATE *date, const char *range = NULL,
   DTF_FLAGS dtFlags = DTF_NO_FLAGS, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);

UIW_FORMATTED_STRING(int left, int top, int width, char *compressedText, char *editMask,
   char *deleteText, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);
```

editMask values:		
	a - Alphabetic	A - Force uppercase Alphabetic
	c - Alphanumeric	C - Force uppercase Alphanumeric
	x - Any character	X - Force uppercase any character
	L - Literal (Skip)	N - Numeric

```
UIW_GROUP(int left, int top, int width, int height, char *text, WNF_FLAGS wnFlags = WNF_AUTO_SELECT,
   WOF_FLAGS woFlags = WOF_NO_FLAGS);

UIW_HZ_LIST(int left, int top, int width, int height, int cellWidth, int cellHeight,
   int (*compareFunction)(void *element1, void *element2) = NULL, WNF_FLAGS wnFlags = WNF_NO_WRAP,
   WOF_FLAGS woFlags = WOF_BORDER, WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS);
UIW_HZ_LIST(int left, int top, int width, int height, int (*compareFunction)(void *element1, void *element2),
   WOF_FLAGS flagSetting, UI_ITEM *item);

UIW_ICON(int left, int top, char *iconName, char *title = NULL, ICF_FLAGS icFlags = ICF_NO_FLAGS,
   WOF_FLAGS woFlags = WOF_JUSTIFY_CENTER | WOF_NON_SELECTABLE,
   USER_FUNCTION userFunction = NULL);

UIW_INTEGER(int left, int top, int width, int *value, const char *range = NULL,
   NMF_FLAGS nmFlags = NMF_NO_FLAGS, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);

UIW_MAXIMIZE_BUTTON(void);

UIW_MINIMIZE_BUTTON(void);

UIW_POP_UP_ITEM(void);
UIW_POP_UP_ITEM(char *text, MNIF_FLAGS mniFlags = MNIF_NO_FLAGS, BTF_FLAGS btFlags = BTF_NO_3D,
   WOF_FLAGS woFlags = WOF_NO_FLAGS, USER_FUNCTION userFunction = NULL, EVENT_TYPE value = 0);

UIW_PROMPT(int left, int top, char *text, WOF_FLAGS woFlags = WOF_NO_FLAGS);
UIW_PROMPT(int left, int top, int width, char *text, WOF_FLAGS woFlags = WOF_NO_FLAGS);

UIW_PULL_DOWN_ITEM(char *text, WNF_FLAGS wnFlags = WNF_NO_FLAGS,
   USER_FUNCTION userFunction = NULL, EVENT_TYPE value = 0);
UIW_PULL_DOWN_ITEM(char *text, WNF_FLAGS wnFlags, UI_ITEM *item);

UIW_PULL_DOWN_MENU(int indentation = 0,
   WOF_FLAGS woFlags = WOF_BORDER | WOF_NON_FIELD_REGION | WOF_SUPPORT_OBJECT,
   WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS);
UIW_PULL_DOWN_MENU(int indentation, UI_ITEM *item);

UIW_REAL(int left, int top, int width, double *value, const char *range = NULL,
   NMF_FLAGS nmFlags = NMF_NO_FLAGS, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);
```

```
UIW_SCROLL_BAR(int left, int top, int width, int height, SBF_FLAGS sbFlags = SBF_VERTICAL,
   WOF_FLAGS woFlags = WOF_BORDER | WOF_SUPPORT_OBJECT | WOF_NON_FIELD_REGION);
UIW_STRING(int left, int top, int width, char *text, int maxLength = -1, STF_FLAGS stFlags = STF_NO_FLAGS,
   WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR, USER_FUNCTION userFunction = NULL);
UIW_SYSTEM_BUTTON(SYF_FLAGS syFlags = SYF_NO_FLAGS);
UIW_SYSTEM_BUTTON(UI_ITEM *item);
static UIW_SYSTEM_BUTTON *UIW_SYSTEM_BUTTON::Generic(void);
UIW_TEXT(int left, int top, int width, int height, char *text, int maxLength = -1,
   WNF_FLAGS wnFlags = WNF_NO_WRAP, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);
UIW_TIME(int left, int top, int width, UI_TIME *time, const char *range = NULL,
   TMF_FLAGS tmFlags = TMF_NO_FLAGS, WOF_FLAGS woFlags = WOF_BORDER | WOF_AUTO_CLEAR,
   USER_FUNCTION userFunction = NULL);
UIW_TITLE(char *text, WOF_FLAGS woFlags = WOF_BORDER | WOF_JUSTIFY_CENTER);
UIW_TOOL_BAR(int left, int top, int width, int height, WNF_FLAGS wnFlags = WNF_NO_FLAGS,
   WOF_FLAGS woFlags = WOF_BORDER | WOF_NON_FIELD_REGION,
   WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS);
UIW_VT_LIST(int left, int top, int width, int height,
   int (*compareFunction)(void *element1, void *element2) = NULL,
   WNF_FLAGS wnFlags = WNF_NO_WRAP, WOF_FLAGS woFlags = WOF_BORDER,
   WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS);
UIW_VT_LIST(int left, int top, int width, int height, int (*compareFunction)(void *element1, void *element2),
   WOF_FLAGS flagSetting, UI_ITEM *item);
UIW_WINDOW(int left, int top, int width, int height, WOF_FLAGS woFlags = WOF_NO_FLAGS,
   WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS,
   UI_HELP_CONTEXT helpContext = NO_HELP_CONTEXT, UI_WINDOW_OBJECT *minObject = NULL);
UIW_WINDOW(const char *name, UI_STORAGE *file = NULL, UI_STORAGE_OBJECT *object = NULL);
static UIW_WINDOW *UIW_WINDOW::Generic(int left, int top, int width, int height, char *title,
   UI_WINDOW_OBJECT *minObject = NULL, WOF_FLAGS woFlags = WOF_NO_FLAGS,
   WOAF_FLAGS woAdvancedFlags = WOAF_NO_FLAGS,
   UI_HELP_CONTEXT helpContext = NO_HELP_CONTEXT);
static UI_WINDOW_OBJECT *UIW_WINDOW::New(const char *name, UI_STORAGE *file = NULL,
   UI_STORAGE_OBJECT *object = NULL);
typedef
   EVENT_TYPE (*USER_FUNCTION)(UI_WINDOW_OBJECT * object, UI_EVENT &event, EVENT_TYPE ccode);
struct UI_ITEM
{
   EVENT_TYPE value;
   void *data
   char *text;
   UIF_FLAGS flags;
};
```

FLAGS, STATUS, AND RESULTS

BDF_FLAGS

BDF_NO_FLAGS	0x0000

BTF_FLAGS

BTF_NO_FLAGS	0x0000
BTF_NO_TOGGLE	0x0001
BTF_DOWN_CLICK	0x0002
BTF_REPEAT	0x0008
BTF_DOUBLE_CLICK	0x0010
BTF_AUTO_SIZE	0x0020
BTF_NO_3D	0x0040
BTF_CHECK_BOX	0x0080
BTF_RADIO_BUTTON	0x0100
BTF_SEND_MESSAGE	0x0200
BTF_STATIC_BITMAPARRAY	0x0400

BTS_STATUS

BTS_NO_STATUS	0x0000
BTS_DEPRESSED	0x0001

DTF_FLAGS

DTF_NO_FLAGS	0x0000
DTF_US_FORMAT	0x0001
DTF_EUROPEAN_FORMAT	0x0002
DTF_JAPANESE_FORMAT	0x0004
DTF_MILITARY_FORMAT	0x0008
DTF_DASH	0x0010
DTF_SLASH	0x0020
DTF_ALPHA_MONTH	0x0040
DTF_DAY_OF_WEEK	0x0080
DTF_UPPER_CASE	0x0100
DTF_SHORT_YEAR	0x0200
DTF_SHORT_MONTH	0x0400
DTF_SHORT_DAY	0x0800
DTF_ZERO_FILL	0x1000
DTF_SYSTEM	0x4000

DTI_RESULT

DTI_OK	0
DTI_INVALID	1
DTI_AMBIGUOUS	2
DTI_INVALID_NAME	3
DTI_VALUE_MISSING	4
DTI_OUT_OF_RANGE	5

FMI_RESULT

FMI_OK	0
FMI_INVALID_CHARACTERS	1

ICF_FLAGS

ICF_NO_FLAGS	0x0000
ICF_DOUBLE_CLICK	0x0001
ICF_MINIMIZE_OBJECT	0x0002
ICF_STATIC_ICONARRAY	0x0400

MNIF_FLAGS

MNIF_NO_FLAGS	0x0000
MNIF_SEPARATOR	0x0001
MNIF_MAXIMIZE	0x0002
MNIF_MINIMIZE	0x0004
MNIF_MOVE	0x0008
MNIF_SIZE	0x0010
MNIF_SWITCH	0x0020
MNIF_RESTORE	0x0040
MNIF_CLOSE	0x0080
MNIF_CHECK_MARK	0x0100
MNIF_SEND_MESSAGE	0x0200
MNIF_NON_SELECTABLE	0x0400

NMF_FLAGS

NMF_NO_FLAGS	0x0000
NMF_CURRENCY	0x0002
NMF_CREDIT	0x0004
NMF_COMMAS	0x0008
NMF_PERCENT	0x0010
NMF_SCIENTIFIC	0x0020
NMF_DECIMAL(decimal)	(((decimal) + 1) << 8)

NMI_RESULT

NMI_OK	0
NMI_OUT_OF_RANGE	1
NMI_INVALID	2

SBF_FLAGS

SBF_NO_FLAGS	0x0000
SBF_CORNER	0x0001
SBF_VERTICAL	0x0002
SBF_HORIZONTAL	0x0004

STF_FLAGS

STF_NO_FLAGS	0x0000
STF_VARIABLE_NAME	0x0001
STF_LOWER_CASE	0x0002
STF_UPPER_CASE	0x0004
STF_PASSWORD	0x0008

SYF_FLAGS

SYF_NO_FLAGS	0x0000
SYF_GENERIC	0x0001

TMF_FLAGS

TMF_NO_FLAGS	0x0000
TMF_SECONDS	0x0001
TMF_HUNDREDTHS	0x0002
TMF_NO_HOURS	0x0004
TMF_NO_MINUTES	0x0008
TMF_TWELVE_HOUR	0x0010
TMF_TWENTY_FOUR_HOUR	0x0020
TMF_ZERO_FILL	0x0040
TMF_COLON_SEPARATOR	0x0080
TMF_NO_SEPARATOR	0x0100
TMF_UPPER_CASE	0x0200
TMF_LOWER_CASE	0x0400
TMF_SYSTEM	0x0800

TMI_RESULT

TMI_OK	0
TMI_INVALID	1
TMI_OUT_OF_RANGE	2
TMI_VALUE_MISSING	3

UIS_FLAGS

UIS_READ	0x0001
UIS_READWRITE	0x0002
UIS_CREATE	0x0004
UIS_OPENCREATE	0x0008
UIS_TEMPORARY	0x0010

WNF_FLAGS

WNF_NO_FLAGS	0x0000
WNF_NO_WRAP	0x0001
WNF_SELECT_MULTIPLE	0x0002
WNF_BITMAP_CHILDREN	0x0004
WNF_AUTO_SORT	0x0008
WNF_CONTINUE_SELECT	0x0010
WNF_AUTO_SELECT	0x0020

WOAF_FLAGS

WOAF_NO_FLAGS	0x0000
WOAF_OUTSIDE_REGION	0x0001
WOAF_NON_CURRENT	0x0002
WOAF_TEMPORARY	0x0004
WOAF_NO_DESTROY	0x0010
WOAF_NORMAL_HOT_KEYS	0x0020
WOAF_NO_SIZE	0x0040
WOAF_NO_MOVE	0x0080
WOAF_MODAL	0x0100
WOAF_LOCKED	0x0200
WOAF_MDI_OBJECT	0x4000
WOAF_DIALOG_OBJECT	0x8000

WOF_FLAGS

WOF_NO_FLAGS	0x0000
WOF_JUSTIFY_CENTER	0x0001
WOF_JUSTIFY_RIGHT	0x0002
WOF_BORDER	0x0004
WOF_NO_ALLOCATE_DATA	0x0008
WOF_VIEW_ONLY	0x0010
WOF_SUPPORT_OBJECT	0x0020
WOF_MINICELL	0x0040
WOF_UNANSWERED	0x0080
WOF_INVALID	0x0100
WOF_NON_FIELD_REGION	0x0200
WOF_NON_SELECTABLE	0x0400
WOF_AUTO_CLEAR	0x0800

WOS_STATUS

WOS_NO_STATUS	0x0000
WOS_GRAPHICS	0x0001
WOS_CURRENT	0x0002
WOS_CHANGED	0x0004
WOS_SELECTED	0x0008
WOS_UNANSWERED	0x0010
WOS_INVALID	0x0020
WOS_MAXIMIZED	0x0040
WOS_MINIMIZED	0x0080
WOS_REDISPLAY	0x0100
WOS_READ_ERROR	0x0200
WOS_OWNERDRAW	0x1000

HOT KEYS

HOT_KEY_MAXIMIZE	252
HOT_KEY_MINIMIZE	253
HOT_KEY_SYSTEM	254
HOT_KEY_SUB_WINDOW	255

DISPLAY INFORMATION

PALETTES

```
struct UI_PALETTE
{
    // —- Text mode —-
    UCHAR fillCharacter;
    COLOR colorAttribute;
    COLOR monoAttribute;

    // —- Text mode —-
    LOGICAL_PATTERN fillPattern
    COLOR colorForeground;
    COLOR colorBackground;
    COLOR bwForeground;
    COLOR bwBackground;
    COLOR grayScaleForeground;
    COLOR grayScaleBackground;
};
stuct UI_PALETTE_MAP
{
    OBJECTID objectID;
    LOGICAL_PALETTE logicalPalatte;
    UI_PALETTE palatte;
};
```

LOGICAL_PALETTES

PM_ANY	0x0000
PM_ACTIVE	0x0001
PM_INACTIVE	0x0002
PM_CURRENT	0x0004
PM_SELECTED	0x0008
PM_NON_SELECTABLE	0x0010
PM_HOT_KEY	0x0020
PM_SPECIAL	0x0040

MONOCHROME COLORS

MONO_BLACK	0x00
MONO_NORMAL	0x07
MONO_DIM	0x08
MONO_HIGH	0x0F

BLACK & WHITE

BW_BLACK	0x00
BW_WHITE	0x01

GRAY SCALE

GS_BLACK	0x00
GS_GRAY	0x01
GS_BLINKING	0x02
GS_WHITE	0x03

COLORS

BLACK	0x00
BLUE	0x01
GREEN	0x02
CYAN	0x03
RED	0x04
MAGENTA	0x05
BROWN	0x06
LIGHTGRAY	0x07
DARKGRAY	0x08
LIGHTBLUE	0x09
LIGHTGREEN	0x0A
LIGHTCYAN	0x0B
LIGHTRED	0x0C
LIGHTMAGENTA	0x0D
YELLOW	0x0E
WHITE	0x0F
BACKGROUND	0xFF

LOGICAL_PATTERNS

PTN_SOLID_FILL	0x0001
PTN_INTERLEAVE_FILL	0x0009
PTN_BACKGROUND_FILL	0x000C

ZIL MSWINDOWS, ZIL OS2, ZIL MOTIF

PTN_SYSTEM_COLOR	0x00F0
PTN_RGB_COLOR	0x00F1

LOGICAL_FONTS

FNT_SMALL_FONT	0x0000
FNT_DIALOG_FONT	0x0001
FNT_SYSTEM_FONT	0x0002

IMAGE_TYPES

MOUSE_IMAGE	0
CURSOR_IMAGE	1

SCREEN_IDs

ZIL MSDOS

ID_DIRECT	0xFFFF
ID_SCREEN	0x0001

ZIL MSWINDOWS

ID_DIRECT	0xFFFF
ID_SCREEN	0x0000

WIN32

ID_DIRECT	0x0000
ID_SCREEN	0x0000

ZIL OS2

ID_DIRECT	0xFFFF
ID_SCREEN	0x0000

ZIL MOTIF

ID_DIRECT	0x0000
ID_SCREEN	0x0000

TDM_MODE

TDM_AUTO	-1
TDM_BW_25x40	0
TDM_25x40	1
TDM_BW_25x80	2
TDM_25x80	3
TDM_MONO_25x80	7
TDM_43x80	64
TDM_NONE	0xFF

IDs

Object IDs

ID_END	-1
ID_BORDER	1
ID_BUTTON	2
ID_DATE	3
D_FORMATTED_STRING	4
ID_ICON	5
ID_INTEGER	6
ID_MAXIMIZE_BUTTON	7
ID_MINIMIZE_BUTTON	8
ID_NUMBER	9
ID_PROMPT	11
ID_REAL	12
ID_STRING	13
ID_TEXT	15
ID_TIME	16
ID_TITLE	17
ID_WINDOW_OBJECT	18
ID_POP_UP_ITEM	19
ID_HELP_CONTEXT	20
ID_BITMAP_IMAGE	21
ID_ICON_IMAGE	22
ID_GROUP	23
ID_BIGNUM	24
ID_MATRIX	1000
ID_LIST	1001
ID_POP_UP_MENU	1002
ID_PULL_DOWN_ITEM	1003
ID_PULL_DOWN_MENU	1004
ID_SCROLL_BAR	1005
ID_SYSTEM_BUTTON	1006
ID_WINDOW	1007
ID_COMBO_BOX	1008
ID_TOOL_BAR	1009
ID_VT_LIST	1010
ID_VLIST	1010
ID_HZ_LIST	1011
ID_HLIST	1011
ID_DISPLAY	2000
ID_EVENT_MANAGER	2001
ID_WINDOW_MANAGER	2002
ID_OUTLINE	2100
ID_WHITE_SHADOW	2101
ID_LIGHT_SHADOW	2102
ID_DARK_SHADOW	2103
ID_BLACK_SHADOW	2104
ID_MENU	2105
ID_MENU_ITEM	2106
ID_HOT_KEY	2107

NUMBER IDS

NUMID_BORDER	0xFFFF
NUMID_MAXIMIZE	0xFFFE
NUMID_MINIMIZE	0xFFFD
NUMID_SYSTEM	0xFFFC
NUMID_TITLE	0xFFFB
NUMID_OPT_RESTORE	0xFFEF
NUMID_OPT_MOVE	0xFFEE
NUMID_OPT_SIZE	0xFFED
NUMID_OPT_MINIMIZE	0xFFEC
NUMID_OPT_MAXIMIZE	0xFFEB
NUMID_OPT_CLOSE	0xFFEA
NUMID_OPT_SWITCH	0xFFE9

INFO REQUESTS

INFO_REQUESTS

GET_NUMBERID_OBJECT	0x0001
GET_STRINGID_OBJECT	0x0002
GET_FLAGS	0x0003
SET_FLAGS	0x0004
CLEAR_FLAGS	0x0005
CHANGED_FLAGS	0x0006
GET_STATUS	0x0007
SET_STATUS	0x0008
CLEAR_STATUS	0x0009
CHANGED_STATUS	0x000A
GET_TEXT	0x000B
SET_TEXT	0x000C
GET_TEXT_LENGTH	0x000D
SET_TEXT_LENGTH	0x000E
COPY_TEXT	0x000F
SET_VSCROLL	0x0085
SET_HSCROLL	0x0086
CHECK SELECTION	0x0087
GET_VALUE	0x0101
SET_VALUE	0x0102
GET_BITMAP_WIDTH	0x0103
GET_ICON_WIDTH	0x0103
SET_BITMAP_WIDTH	0x0104
GET_BITMAP_HEIGHT	0x0105
GET_ICON_HEIGHT	0x0105
SET_BITMAP_HEIGHT	0x0106
GET_BITMAP_ARRAY	0x0107
GET_ICON_ARRAY	0x0107
SET_BITMAP_ARRAY	0x0108
SET_ICON_ARRAY	0x0108
RESET_SELECTION	0x0200

ZIL MSWINDOWS

GET_DWSTYLE	0x0051
SET_DWSTYLE	0x0052
CLEAR_DWSTYLE	0x0053
CHANGED_DWSTYLE	0x0054
HIDE_SUBWINDOW	0x0055

ZIL OS2

GET_FLSTYLE	0x0051
SET_FLSTYLE	0x0052
CLEAR_FLSTYLE	0x0053
CHANGED_FLSTYLE	0x0054
GET_FLFLAG	0x0055
SET_FLFLAG	0x0056
CLEAR_FLFLAG	0x0057
CHANGED_FLFLAG	0x0058

ZIL MOTIF

GET_SYSTEM_OPTIONS	0x0051
SET_SYSTEM_OPTIONS	0x0052
GET_DECORATIONS	0x0053
SET_DECORATIONS	0x0054

EVENT INFORMATION

EVENT STRUCTURE

```
struct UI_EVENT
{
  EVENT_TYPE type;
  RAW_CODE rawCode;
  RAW_CODE modifiers;
#if defined(ZIL_MSWINDOWS)
  MSG message;
#elif defined(ZIL_OS2)
  QSMG message;
#elif defined(ZIL_MOTIF)
  XEvent message;
#endif
  union
  {
    UI_KEY key;
    UI_REGION region;
    UI_POSITION position;
    UI_SCROLL_INFORMATION scroll;
    void *data;
  };
  UI_EVENT(void);
  UI_EVENT(EVENT_TYPE type, RAW_CODE rawCode = 0
  UI_EVENT(EVENT_TYPE type, RAW_CODE rawCode, const UI_KEY &key);
  UI_EVENT(EVENT_TYPE type, RAW_CODE rawCode, const UI_REGION &region);
  UI_EVENT(EVENT_TYPE type, RAW_CODE rawCode, const UI_POSITION &position);
  UI_EVENT(EVENT_TYPE type, RAW_CODE rawCode, const UI_SCROLL_INFORMATION &scroll);
#if defined(ZIL_MSWINDOWS) && defined(WIN32)
  UI_EVENT(EVENT_TYPE type, HWND hWnd, UINT wMsg, WPARAM wParam, LPARAM lParam);
#elif defined(ZIL_MSWINDOWS)
  UI_EVENT(EVENT_TYPE type, HWND hWnd, UINT wMsg, UINT wParam, LONG lParam);
  UI_EVENT(EVENT_TYPE type, HWND hWnd, WORD wMsg, WORD wParam, LONG lParam);
#elif defined(ZIL _OS2)
  UI_EVENT(EVENT_TYPE type, HWND hWnd, ULONG msg, MPARAM mp1, MPARAM mp2);
#elif defined(ZIL_MOTIF)
  UI_EVENT(EVENT_TYPE type, XEvent &xevent);
#endif
};

struct UI_KEY
{
  RAW_CODE shiftState;
  RAW_CODE value;
};

struct UI_REGION
{
  int left, top, right, bottom;
};
struct UI POSITION
{
  int column, line;
};
struct  UI_SCROLL_INFORMATION
{
  int current, minimum, maximum, showing, delta;
};
```

UI_EVENT_MAP

```
struct UI_EVENT_MAP
{
   OBJECTID objectID;
   LOGICAL_EVENT logicalValue;
   EVENT_TYPE eventType;
   RAW_CODE rawCode;
   RAW_CODE modifiers;
};
```

Q_FLAGS

Q_NORMAL	0x0000
Q_BLOCK	0x0000
Q_NO_BLOCK	0x0001
Q_BEGIN	0x0000
Q_END	0x0002
Q_DESTROY	0x0000
Q_NO_DESTROY	0x0004
Q_POLL	0x0000
Q_NO_POLL	0x0008

EVENT RAWCODE RANGES

AVAILABLE TO USER	-32768 through -1000
SYSTEM EVENTS	-999 THROUGH -1
RAW DEVICE TYPES	0 THROUGH 99
LOGICAL EVENTS	100 THROUGH 9999
AVAILABLE TO USER	10000 through 32767

DEVICE_TYPES

E_DEVICE_FIRST	0
E_MSWINDOWS	1
E_OS2	2
E_MOTIF	3
E_KEY	10
E_MOUSE	30
E_PENDOS	30
E_CURSOR	50
E_DEVICE	99
E_DEVICE_LAST	99

DEVICE EVENT TYPES

D_OFF	0x0500
D_ON	0x0501
D_HIDE	0x0502
D_ACTIVATE	0x0503

CURSOR EVENT TYPES

DC_INSERT	0x0510
DC_OVERSTRIKE	0x0511

MOUSE EVENT TYPES

DM_VIEW	0x0510
DM_EDIT	0x0511
DM_WAIT	0x0512
DM_MOVE	0x0513
DM_HORIZONTAL	0x0514
DM_VERTICAL	0x0515
DM_DIAGONAL_ULLR	0x0516
DM_DIAGONAL_LLUR	0x0517
DM_POSITION	0x0518

KEYBOARD SHIFTSTATES

S_RIGHT_SHIFT	0x0001
S_LEFT_SHIFT	0x0002
S_SHIFT	0x0003
S_CTRL	0x0004
S_ALT	0x0008
S_SCROLL_LOCK	0x0010
S_NUM_LOCK	0x0020
S_CAPS_LOCK	0x0040
S_INSERT	0x0080

MOUSE BUTTON STATES

M_LEFT	0x0100
M_RIGHT	0x0200
M_MIDDLE	0x0400
M_TOP_CHANGE	0x0800
M_LEFT_CHANGE	0x1000
M_RIGHT_CHANGE	0x2000
M_MIDDLE_CHANGE	0x4000
M_BOTTOM_CHANGE	0x8000

SYSTEM_EVENTS

S_MDICHILD_EVENT	(Add -500)
S_SYSTEM_LAST	-1
S_ERROR	-1
S_UNKNOWN	-2
S_NO_OBJECT	-3
S_MINIMIZE	-4
S_MAXIMIZE	-5
S_ALT_KEY	-6
S_CONTINUE	-7
S_ERROR_RESPONSE	-8
S_INITIALIZE	-9
S_CREATE	-10
S_CLOSE	-11
S_NON_CURRENT	-13
S_MOVE	-14
S_REDISPLAY	-17
S_CLOSE_TEMPORARY	-18
S_DEINITIALIZE	-20
S_REGION_DEFINE	-21
S_CASCADE	-24
S_RESTORE	-25
S_REGISTER_OBJECT	-26
S_POSITION	-100
S_SIZE	-200
S_CHANGE	-201
S_DISPLAY_ACTIVE	-202
S_DISPLAY_INACTIVE	-203
S_CURRENT	-204
S_VSCROLL	-300
S_HSCROLL	-301
S_VSCROLL_SET	-302
S_HSCROLL_SET	-303
S_VSCROLL_CHECK	-304
S_HSCROLL_CHECK	-305
S_ADD_OBJECT	-400
S_SUBTRACT_OBJECT	-401
S_RESET_DISPLAY	-402
S_SYSTEM_FIRST	-999

LOGICAL_EVENTS

L_MDICHILD_EVENT	(Add 500)
L_LOGICAL_FIRST	100
L_EXIT	1000
L_VIEW	1001
L_SELECT	1002
L_BEGIN_SELECT	1003
L_CONTINUE_SELECT	1004
L_END_SELECT	1005
L_BEGIN_ESCAPE	1006
L_CONTINUE_ESCAPE	1007
L_END_ESCAPE	1008
L_HELP	1009
L_CANCEL	1010
L_EXIT_FUNCTION	1011
L_DOUBLE_CLICK	1012
L_MOVE	1013
L_SIZE	1014
L_NEXT_WINDOW	1040
L_UP	1050
L_DOWN	1051
L_LEFT	1052
L_RIGHT	1053
L_PREVIOUS	1054
L_NEXT	1055
L_FIRST	1056
L_TOP	1056
L_LAST	1057
L_BOTTOM	1057
L_PGUP	1058
L_PGDN	1059
L_BEGIN_MARK	1101
L_CONTINUE_MARK	1102
L_END_MARK	1103
L_CUT	1104
L_PASTE	1105
L_CUT_PASTE	1106
L_MARK	1107
L_DELETE	1108
L_DELETE_WORD	1109
L_DELETE_EOL	1110
L_INSERT_TOGGLE	1114
L_WORD_LEFT	1115
L_WORD_RIGHT	1116
L_BOL	1117
L_EOL	1118
L_BACKSPACE	1119
L_COPY_MARK	1127
L_LOGICAL_LAST	9999

OBJECT HIERARCHY

```
class UI_APPLICATION
class UI_DISPLAY
  ├†class UI_BGI_DISPLAY
  ├†class UI_FG_DISPLAY
  ├†class UI_GRAPHICS_DISPLAY
  ├─class UI_MOTIF_DISPLAY
  ├†class UI_MSC_DISPLAY
  ├─class UI_OS2_DISPLAY
  ├─class UI_MSWINDOWS_DISPLAY
  └†class UI_TEXT_DISPLAY
class UI_ELEMENT
  ├─class UI_DEVICE
  │  ├─class UID_CURSOR
  │  ├─class UID_KEYBOARD
  │  ├─class UID_MOUSE
  │  └─class UID_PENDOS
  ├─class UI_PATH_ELEMENT
  ├─class UI_QUEUE ELEMENT
  ├─class UI_REGION_ELEMENT
  └─class UI_WINDOW_OBJECT
     ├─class UIW_BORDER
     ├─class UIW_BUTTON
     │  ├─class UIW_MAXIMIZE_BUTTON
     │  ├─class UIW_MINIMIZE_BUTTON
     │  ├─class UIW_POP_UP_ITEM
     │  ├─class UIW_PULL_DOWN_ITEM
     │  ├─class UIW_SYSTEM_BUTTON
     │  └─class UIW_TITLE
```

```
     ├─class UIW_ICON
     ├─class UIW_PROMPT
     ├─class UIW_STRING
     │  ├─class UIW_BIGNUM
     │  ├─class UIW_DATE
     │  ├─class UIW_FORMATTED_STRING
     │  ├─class UIW_INTEGER
     │  ├─class UIW_REAL
     │  └─class UIW_TIME
     └†class UIW_WINDOW
        ├─class UI_WINDOW_MANAGER
        ├─class UIW_COMBO_BOX
        ├─class UIW_GROUP
        ├─class UIW_HZ_LIST
        ├─class UIW_POP_UP_MENU
        ├─class UIW_PULL_DOWN_MENU
        ├─class UIW_SCROLL_BAR
        ├─class UIW_TEXT
        ├─class UIW_TOOL_BAR
        └─class UIW_VT_LIST
class UI_ERROR_SYSTEM
class UI_HELP_SYSTEM
class UI_INTERNATIONAL
  ├─class UI_BIGNUM
  ├─class UI_DATE
  └─class UI_TIME
```

```
class UI_LIST
  ├─class UI_EVENT_MANAGER
  ├─class UI_LIST_BLOCK
  │  └─class UI_QUEUE_BLOCK
  ├─class UI_PATH
  ├─class UI_REGION_LIST
  │  ├†class UI_BGI_DISPLAY
  │  ├†class UI_FG_DISPLAY
  │  ├†class UI_GRAPHICS_DISPLAY
  │  ├†class UI_MSC_DISPLAY
  │  └†class UI_TEXT_DISPLAY
  └†class UIW_WINDOW
class UI_STORAGE
class UI_STORAGE_OBJECT
struct UI_EVENT
struct UI_EVENT_MAP
struct UI_ITEM
struct UI_KEY
struct UI_PALETTE
struct UI_PALETTE_MAP
struct UI_POSITION
struct UI_REGION
struct UI_SCROLL_INFORMATION
```

† indicates multiple inheritance

(UI = user interface)

(UID = user interface device)

(UIW = user interface window object)

UNITED STATES

ZINC SOFTWARE INCORPORATED
405 SOUTH 100 EAST, 2ND FLOOR
PLEASANT GROVE, UTAH 84062 USA
CORPORATE OFFICES: (801) 785-8900
TECHNICAL SUPPORT: (801) 785-8998
BBS: (801) 785-8997
FAX: (801) 785-8996

EUROPE

ZINC SOFTWARE (UK) LIMITED
58-60 BERESFORD STREET
LONDON, SE18 6BG UNITED KINGDOM
EUROPEAN SALES AND
TECHNICAL SUPPORT: +44 (0)81 855 9918
BBS: +44 (0)81 317 2310
FAX: +44 (0)81 316 7778