XVT Portability Toolkit ™
Quick Reference

Published By

Providence Software Solutions, Inc.
201 Shannon Oaks Circle, Suite 200
Cary, NC 27511 USA

Printing History

First Printing ..... March, 1996 ..... XVT PTK Version 4.5
Second Printing..... October, 1996 ..... XVT PTK Version 4.55
Third Printing..... January, 1999 ..... XVT PTK Version 5
Fourth Printing..... June, 1999 ..... XVT PTK Version 5
Revision ..... November, 2002 .... XVT PTK Version 5.5
Revision ..... December, 2004 .... XVT PTK Version 5.6
Revision ..... April, 2007 ..... XVT PTK Version 5.8
Revision..... March, 2009 ..... XVT/PTK Version 2009.1
Revision..... January, 2010 ..... XVT/PTK Version 2010.1

QUICK REFERENCE

CONTENTS

XVT Portability Toolkit

Application Objects

See Also: Events, Visible Objects (Application Configuration Attributes)

```
void                xvt_app_allow_quit   (void);
void                xvt_app_create   (int argc, char * argv[], unsigned long flags,
                        EVENT_HANDLER eh, XVT_CONFIG * config);
void                xvt_app_destroy   (void);
BOOLEAN             xvt_app_escape   (int esc_code, ...);
DRAW_CTOOLS *       xvt_app_get_default_ctools   (DRAW_CTOOLS * ctoolsp);
FILE_SPEC *         xvt_app_get_file   (void);
void                xvt_app_get_files_count   (BOOLEAN * printp, int * countp);
void                xvt_app_process_pending_events   (void);
void                xvt_app_set_file_processed   (void);
```

XVT_CONFIG Type

```
typedef struct s_xvt_config {
    short menu_bar_ID;                  /* menu bar resource ID */
    short about_box_ID;                 /* about box resource ID */
    char * base_appl_name;              /* application's "file" name */
    char * appl_name;                   /* application's name */
    char * taskwin_title;               /* title for task window */
} XVT_CONFIG;
```

Attributes

See: Visible Objects

Carets

See Also: Cursors

```
void                xvt_win_set_caret_pos   (WINDOW win, PNT pos);
void                xvt_win_set_caret_size   (WINDOW win, int width, int height);
void                xvt_win_set_caret_visible   (WINDOW win, BOOLEAN visible);
```

PNT Type

```
typedef struct s_pnt {                  /* mathematical point */
    short v;                            /* vertical (y) coordinate */
    short h;                            /* horizontal (x) coordinate */
} PNT;
```

Characters

See: Strings

Clipboard

```
DATA_PTR        xvt_cb_alloc_data   (long size);
BOOLEAN         xvt_cb_close   (void);
void            xvt_cb_free_data   (void);
DATA_PTR        xvt_cb_get_data   (CB_FORMAT cbfmt, char * name, long * sizep);
BOOLEAN         xvt_cb_has_format   (CB_FORMAT cbfmt, char * name);
BOOLEAN         xvt_cb_open   (BOOLEAN writing);
BOOLEAN         xvt_cb_put_data   (CB_FORMAT cbfmt, char * name, long size, PICTURE pic);
```

CB_FORMAT Enumeration Type

```
typedef enum e_cb {
    CB_TEXT,                            /* ASCII text */
    CB_PICT,                            /* encapsulated picture */
    CB_APPL                             /* application's own type (must have name) */
} CB_FORMAT;
```

Colors

See Also: Controls, Drawing, Palettes

```
#define XVT_COLOR_GET_BLUE   (color)   ...
#define XVT_COLOR_GET_GREEN   (color)   ...
#define XVT_COLOR_GET_RED   (color)   ...
#define XVT_MAKE_COLOR   (r, g, b)   ...
```

Color Type

```
typedef unsigned long COLOR;            /* color encapsulation */
```

Color Constants

```
#define COLOR_BLACK ...
#define COLOR_BLUE ...
#define COLOR_CYAN ...
#define COLOR_DKGRAY ...
#define COLOR_GRAY ...
#define COLOR_GREEN ...
#define COLOR_LTGRAY ...
#define COLOR_MAGENTA ...
#define COLOR_RED ...
#define COLOR_WHITE
#define COLOR_YELLOW ...

#define COLOR_INVALID ...
```

Container Extension Objects

See Also: Controls

```
long                    xvt_cxo_call_next   (XVT_CXO cxo, EVENT * ep);
XVT_CXO                 xvt_cxo_create   (WINDOW win, const long state_data,
                            XVT_CXO_INSERTION where, XVT_CXO_EVENT_HANDLER cxo_eh,
                            EVENT_MASK mask, char * class_name, long cxo_id);
void                    xvt_cxo_destroy   (XVT_CXO cxo);
long                    xvt_cxo_dispatch_msg   (XVT_CXO cxo, long msg_id, void * data);
char *                  xvt_cxo_get_class_name   (XVT_CXO cxo, char * class_name,
                            int sz_class_name);
WINDOW                  xvt_cxo_get_win   (XVT_CXO cxo);
long                    xvt_cxo_get_data   (XVT_CXO cxo);
XVT_CXO_EVENT_HANDLER   xvt_cxo_get_event_handler   (XVT_CXO cxo);
EVENT_MASK              xvt_cxo_get_event_mask   (XVT_CXO cxo);
BOOLEAN                 xvt_cxo_is_valid   (XVT_CXO cxo);
void                    xvt_cxo_set_data   (XVT_CXO cxo, long state_data);
void                    xvt_cxo_set_event_handler   (XVT_CXO cxo,
                            XVT_CXO_EVENT_HANDLER cxo_eh);
void                    xvt_cxo_set_event_mask   (XVT_CXO cxo, EVENT_MASK mask);

SLIST                   xvt_win_list_cxos   (WINDOW win);
XVT_CXO                 xvt_win_get_cxo   (WINDOW win, char * class_name, long cxo_id);
```

XVT_CXO Type

```
typedef long XVT_CXO;
```

*CXO_EVENT_HANDLER Function Prototype

```
typedef long (* XVT_CXO_EVENT) (XVT_CXO cxo, EVENT * ep);
```

use prototype:

```
long XVT_CALLCONV1 cxo_event_handler (XVT_CXO cxo, EVENT * ep);
```

CXO Constants

```
#define SZ_CLASS_NAME ...               /* maximum length of a class name */
#define XVT_CXO_CREATE_MSG ...          /* standard XVT E_CXO messages that are dispatched */
#define XVT_CXO_DESTROY_MSG ...         /* standard XVT E_CXO messages that are dispatched */
```

XVT_CXO_INSERTION Enumeration Type

```
typedef enum {
    XVT_CXO_POS_FIRST,
    XVT_CXO_POS_LAST,
} XVT_CXO_INSERTION;
```

Controls

See Also: Colors, Fonts, HTML Browser, List Box, List Edit Controls, Scrollbar Controls, Resource Manager
Text Edit Objects, Treeview controls, Visible Objects

```
void                    xvt_ctl_check_radio_button   (WINDOW ctl_win, WINDOW * ctls, int nctls);
WINDOW                  xvt_ctl_create   (WIN_TYPE wtype, RCT * rct_p, char * title,
                            WINDOW parent_win, long ctl_flags, long app_data, int ctrl_id);
WINDOW                  xvt_ctl_create_def   (WIN_DEF * win_def_p, WINDOW parent_win,
                            long app_data);
```

color **xvt_ctl_get_color_component** (WINDOW ctl_win, XVT_COLOR_TYPE ctype);
XVT_COLOR_COMPONENT* **xvt_ctl_get_colors** (WINDOW ctl_win);
XVT_FNTID **xvt_ctl_get_font** (WINDOW ctl_win);
int **xvt_ctl_get_id** (WINDOW ctl_win);
COLOR **xvt_ctl_get_native_color_component** (WIN_TYPE wtype, XVT_COLOR_TYPE ctype);
XVT_COLOR_COMPONENT* **xvt_ctl_get_native_colors** (WIN_TYPE wtype);
void **xvt_ctl_get_text_sel** (WINDOW ctl_win, int * first, int * last);
BOOLEAN **xvt_ctl_is_checked** (WINDOW ctl_win);
void **xvt_ctl_set_checked** (WINDOW ctl_win, BOOLEAN check);
void **xvt_ctl_set_color_component** (WINDOW ctl_win, XVT_COLOR_TYPE ctype, COLOR color);
void **xvt_ctl_set_colors** (WINDOW ctl_win, XVT_COLOR_COMPONENT * colors, XVT_COLOR_ACTION action);
void **xvt_ctl_set_font** (WINDOW ctl_win, XVT_FNTID font_id);
void **xvt_ctl_set_text_sel** (WINDOW ctl_win, int first, int last);
void **xvt_ctl_unset_color_component** (WINDOW ctl_win, XVT_COLOR_TYPE ctype);

COLOR **xvt_win_get_ctl_color_component** (WINDOW ctl_win, XVT_COLOR_TYPE ctype);
XVT_COLOR_COMPONENT* **xvt_win_get_ctl_colors** (WINDOW win);
XVT_FNTID **xvt_win_get_ctl_font** (WINDOW win);
void **xvt_win_set_ctl_color_component** (WINDOW ctl_win, XVT_COLOR_TYPE ctype, COLOR color);
void **xvt_win_set_ctl_colors** (WINDOW win, XVT_COLOR_COMPONENT * colors, XVT_COLOR_ACTION action);
void **xvt_win_set_ctl_font** (WINDOW win, XVT_FNTID font_id);
void **xvt_win_unset_ctl_color_component** (WINDOW ctl_win, XVT_COLOR_TYPE ctype);

Control Creation Flag Constants

```
#define CTL_FLAG_CENTER_JUST ...        /* centered text */
#define CTL_FLAG_CHECKED ...
#define CTL_FLAG_DEFAULT ...
#define CTL_FLAG_DISABLED ...
#define CTL_FLAG_GROUP ...
#define CTL_FLAG_INVISIBLE ...
#define CTL_FLAG_LEFT_JUST ...          /* left justified text */
#define CTL_FLAG_MULTIPLE ...
#define CTL_FLAG_NATIVE_JUST ....       /* default text justification */
#define CTL_FLAG_READONLY ...
#define CTL_FLAG_RIGHT_JUST ...         /* right justified text */
#define CTL_FLAG_MAC_GENEVA9 ...        /* opt1 */
#define CTL_FLAG_MAC_MONACO9 ...        /* opt2 */
#define CTL_FLAG_MAC_MULTILINE ...      /* opt3 */
#define CTL_FLAG_MAC_WORDWRAP ...       /* opt4 */
#define CTL_FLAG_PASSWORD ...
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

Codeset Mapping

```
typedef struct s_codeset_map { ... } *XVT_CODESET_MAP;

XVT_CODESET_MAP xvt_str_create_codeset_map( XVT_IOSTREAM fromcodeset,
        XVT_IOSTREAM tocodeset );

void xvt_str_destroy_codeset_map( XVT_CODESET_MAP codeset_map );

long xvt_str_translate_codeset( XVT_CODESET_MAP codeset_map,
        char * string, char * strbuf, size_t bufsize );
```

WIN_DEF Type (Partial)

```
typedef struct s_win_def {
    WIN_TYPE wtype;
    RCT rct;
    char * text;
    UNIT_TYPE units;
    XVT_COLOR_COMPONENT * ctlcolors;
    union {
    ...
        struct {
            short int ctrl_id;
            short int icon_id;
            long flags;
            XVT_FNTID font_id;
        } ctl;
        struct {
            unsigned short attrib;
            XVT_FNTID font_id;
            short int margin;
            short int limit;
            short int tx_id;
        } tx;
    } v;
} WIN_DEF;
```

WIN_TYPE Enumeration Type (Partial)

```
typedef enum e_win_type {               /* type of window */
    ...
    WC_PUSHBUTTON,                      /* button control */
    WC_RADIOBUTTON,                     /* radio button control */
    WC_CHECKBOX,                        /* check box control */
    WC_HSCROLL,                         /* horizontal scrollbar control */
    WC_VSCROLL,                         /* vertical scrollbar control */
    WC_EDIT,                            /* edit control */
    WC_TEXT,                            /* static text control */
    WC_LBOX,                            /* list box control */
    WC_LISTBUTTON,                      /* button with list */
    WC_LISTEDIT,                        /* edit field with list */
    WC_GROUPBOX,                        /* group box */
    WC_TEXTEDIT,                        /* text-edit object */
    WC_ICON,                            /* icon control */
    WC_NOTEBOOK,                        /* notebook control */
    WC_TREEVIEW,                        /* treeview control */
    WC_HTML,                            /* html control */
    ...
} WIN_TYPE;
```

UNIT_TYPE Enumeration Type

```
typedef enum e_unit_type {
    U_PIXELS,
    U_CHARS,
    U_SEMICHARS
} UNIT_TYPE;
```

XVT_COLOR_ACTION Enumeration Type

```
typedef enum e_xvt_color_action {
    XVT_COLOR_ACTION_SET,
    XVT_COLOR_ACTION_UNSET,
} XVT_COLOR_ACTION;
```

XVT_COLOR_COMPONENT Type

```
typedef struct s_xvt_color_component {
    XVT_COLOR_TYPE type;
    COLOR color;
} XVT_COLOR_COMPONENT;
```

XVT_COLOR_TYPE Type

```
typedef unsigned long  XVT_COLOR_TYPE;
```

Color Component Constants (for XVT_COLOR_TYPE)

```
#define XVT_COLOR_BACKGROUND ...
#define XVT_COLOR_BLEND ...
#define XVT_COLOR_BORDER ...
#define XVT_COLOR_FOREGROUND ...
#define XVT_COLOR_HIGHLIGHT ...
#define XVT_COLOR_SELECT ...
#define XVT_COLOR_TROUGH ...
#define XVT_COLOR_NULL ...
```

Cursors

See Also: Carets

```
void        xvt_scr_hide_cursor   (void);
void        xvt_scr_set_busy_cursor   (void);

CURSOR      xvt_win_get_cursor   (WINDOW win);
void        xvt_win_set_cursor   (WINDOW win, CURSOR c);
```

CURSOR Type

```
typedef short CURSOR;                   /* cursor shape */
```

Cursor Type Constants

```
#define CURSOR_ARROW ...                /* arrow */
#define CURSOR_CROSS ...                /* crosshair */
#define CURSOR_IBEAM ...                /* I-beam */
#define CURSOR_HELP ...                 /* help system */
#define CURSOR_PLUS ...                 /* plus sign (fatter than crosshair) */
#define CURSOR_USER ...                 /* user defined shape (>= 11) */
#define CURSOR_WAIT ...                 /* waiting symbol (e.g., hourglass) */
```

Debugging Facility

See Also: Error Handling Facility, Visible Objects (Debugging Attributes)

void **xvt_debug** (char * fmt, ...);
void **xvt_debug_printf** (char * fmt, ...);

Dialogs

See Also: Events, Resource Manager, Windows, Visible Objects

```
WINDOW          xvt_dlg_create_def   (WIN_DEF * win_def_p, EVENT_MASK mask,
                    EVENT_HANDLER eh, long app_data);
WINDOW          xvt_dlg_create_res   (WIN_TYPE wtype, int rid, EVENT_MASK mask,
                    EVENT_HANDLER eh, long app_data);

void            xvt_dm_post_about_box   (void);
ASK_RESPONSE    xvt_dm_post_ask   (char * lbl_dflt, char * lbl2, char * lbl3, char * fmt ...);
BOOLEAN         xvt_dm_post_color_sel   (COLOR * color, long reserved);
BOOLEAN         xvt_dm_post_ctools_sel   (DRAW_CTOOLS * ctoolsp, long attr);
FL_STATUS       xvt_dm_post_dir_sel   (FILE_SPEC * fsp, char * msg);
void            xvt_dm_post_error   (char * fmt, ...);
void            xvt_dm_post_fatal_exit   (char * fmt, ...);
FL_STATUS       xvt_dm_post_file_open   (FILE_SPEC * fsp, char * msg);
FL_STATUS       xvt_dm_post_file_save   (FILE_SPEC * fsp, char * msg);
BOOLEAN         xvt_dm_post_font_sel   (WINDOW win, XVT_FNTID font_id,
                    PRINT_RCD * precp, unsigned long reserved);
void            xvt_dm_post_message   (char * fmt, ...);
void            xvt_dm_post_note   (char * fmt, ...);
BOOLEAN         xvt_dm_post_page_setup   (PRINT_RCD * precp);
char *          xvt_dm_post_string_prompt   (char * msg, char * resp, int sz_resp);
void            xvt_dm_post_warning   (char * fmt, ...);
```

ASK_RESPONSE Enumeration Type

```
typedef enum e_ask_resp {           /* response from xvt_dm_post_ask fcn */
    RESP_DEFAULT,                   /* default button */
    RESP_2,                         /* second button */
    RESP_3                          /* third button */
} ASK_RESPONSE;
```

Dialog Creation Flags Constants

```
#define DLG_FLAG_DISABLED ...
#define DLG_FLAG_INVISIBLE ...
```

Dialog Push Button Control ID Constants

```
#define DLG_CANCEL ...              /* cancel button was clicked */
#define DLG_NO ...                  /* other button was clicked */
#define DLG_OK ...                  /* default button was clicked */
#define DLG_OUTLINE ...             /* ID of userItem on Mac (internal use) */
#define DLG_YES ...                 /* synonym */
```

DRAW_CTOOLS Type

```
typedef struct s_drawct {
    CPEN pen;                       /* color pen */
    CBRUSH brush;                   /* color brush */
    DRAW_MODE mode;                 /* drawing mode */
    COLOR fore_color;               /* foreground color */
    COLOR back_color;               /* background color */
    BOOLEAN opaque_text;            /* is text drawn opaquely? */
} DRAW_CTOOLS;
```

EVENT_HANDLER Function Prototype

```
typedef long (*EVENT_HANDLER) (WINDOW win, EVENT * ep);
```

use prototype:

```
long XVT_CALLCONV1 event_handler (WINDOW win, EVENT * ep);
```

FL_STATUS Enumeration Type

```
typedef enum e_file {                   /* result from file open and save dialogs */
    FL_BAD,                             /* error occurred */
    FL_CANCEL,                          /* cancel button clicked */
    FL_OK                               /* OK button clicked */
} FL_STATUS;
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

WIN_DEF Type (Partial)

```
typedef struct s_win_def {
    WIN_TYPE wtype;
    RCT rct;
    char * text;
    UNIT_TYPE units;
    XVT_COLOR_COMPONENT * ctlcolors;
    union {
        ...
        struct {
            long flags;
            XVT_FNTID ctl_font_id;
        } dlg;
        ...
    } v;
} WIN_DEF;
```

WIN_TYPE Enumeration Type

```
typedef enum e_win_type {               /* type of window */
    ...
    WD_MODAL,                           /* modal dialog */
    WD_MODELESS,                        /* modeless dialog */
    ...
} WIN_TYPE;
```

UNIT_TYPE Enumeration Type

```
typedef enum e_unit_type {
    U_PIXELS,
    U_CHARS,
    U_SEMICHARS
} UNIT_TYPE;
```

XVT_CTOOLS Constants

```
#define XVT_CTOOLS_PEN ...
#define XVT_CTOOLS_PEN_ALL ...
#define XVT_CTOOLS_BRUSH ...
#define XVT_CTOOLS_FORE_COLOR ...
#define XVT_CTOOLS_BACK_COLOR ...
#define XVT_CTOOLS_CTOOL ...
#define XVT_CTOOLS_ALL ...
```

***XVT_FONT_DIALOG Function Prototype**

```
/* Prototype for the application supplied font selection dialog */
typedef BOOLEAN (* XVT_FONT_DIALOG) (
    WINDOW win,                         /* window to send E_FONT event to */
    XVT_FNTID font_id,                  /* default font id */
    PRINT_RCD * precp,                  /* print record or NULL */
    unsigned long reserved              /* reserved */
);
```

use prototype:

```
BOOLEAN XVT_CALLCONV1 xvt_font_dialog (WINDOW win, XVT_FNTID font_id,
    PRINT_RCD * precp, unsigned long reserved);
```

Drawing

See Also: Colors, Fonts, Image Objects, Palettes, Picture Objects, Pixmap Objects, Rectangle Objects

DRAW_CTOOLS *	**xvt_app_get_default_ctools** (DRAW_CTOOLS * ctoolsp);
void	**xvt_dwin_clear** (WINDOW win, COLOR color);
PICTURE	**xvt_dwin_close_pict** (WINDOW win);
void	**xvt_dwin_draw_aline** (WINDOW win, PNT pnt, BOOLEAN start_arrow, BOOLEAN end_arrow);
void	**xvt_dwin_draw_arc** (WINDOW win, RCT * lrctp, int start_x, int start_y, int stop_x, int stop_y);
void	**xvt_dwin_draw_icon** (WINDOW win, int x, int y, int rid);
void	**xvt_dwin_draw_image** (WINDOW win, XVT_IMAGE image, RCT * p_rctDst, RCT * p_rctSrc);
void	**xvt_dwin_draw_line** (WINDOW win, PNT pnt);
void	**xvt_dwin_draw_oval** (WINDOW win, RCT * lrctp);
void	**xvt_dwin_draw_pict** (WINDOW win, PICTURE pic, RCT * lrctp);
void	**xvt_dwin_draw_pie** (WINDOW win, RCT * lrctp, int start_x, int start_y, int stop_x, int stop_y);
void	**xvt_dwin_draw_pmap** (WINDOW win, XVT_PIXMAP pixmap, RCT * p_rctDst, RCT * p_rctSrc);
void	**xvt_dwin_draw_polygon** (WINDOW win, PNT * lpntp, int npnts);
void	**xvt_dwin_draw_polyline** (WINDOW win, PNT * lpntp, int npnts);
void	**xvt_dwin_draw_rect** (WINDOW win, RCT * lrctp);
void	**xvt_dwin_draw_roundrect** (WINDOW win, RCT * lrctp, int oval_width, int oval_height);
void	**xvt_dwin_draw_set_pos** (WINDOW win, PNT lpnt);
void	**xvt_dwin_draw_text** (WINDOW win, int x, int y, char * s, int len);
RCT *	**xvt_dwin_get_clip** (WINDOW win, RCT * lrctp);
DRAW_CTOOLS *	**xvt_dwin_get_draw_ctools** (WINDOW win, DRAW_CTOOLS * ctoolsp);
XVT_FNTID	**xvt_dwin_get_font** (WINDOW win);
long	**xvt_dwin_get_font_app_data** (WINDOW win);
BOOLEAN	**xvt_dwin_get_font_family** (WINDOW win, char * buf, long max_buf);
BOOLEAN	**xvt_dwin_get_font_family_mapped** (WINDOW win, char * buf, long max_buf);
void	**xvt_dwin_get_font_metrics** (WINDOW win, int * leadingp, int * ascentp, int * descentp);
BOOLEAN	**xvt_dwin_get_font_native_desc** (WINDOW win, char * buf, long max_buf);
long	**xvt_dwin_get_font_size** (WINDOW win);
long	**xvt_dwin_get_font_size_mapped** (WINDOW win);
XVT_FONT_STYLE_MASK	**xvt_dwin_get_font_style** (WINDOW win);

XVT_FONT_STYLE_MASK	**xvt_dwin_get_font_style_mapped** (WINDOW win);
int	**xvt_dwin_get_text_width** (WINDOW win, char * s, int len);
void	**xvt_dwin_invalidate_rect** (WINDOW win, RCT * lrctp);
BOOLEAN	**xvt_dwin_is_update_needed** (WINDOW win, RCT * lrctp);
BOOLEAN	**xvt_dwin_open_pict** (WINDOW win, RCT * lrctp);
void	**xvt_dwin_scroll_rect** (WINDOW win, RCT * lrctp, int dh, int dv);
void	**xvt_dwin_set_back_color** (WINDOW win, COLOR color);
void	**xvt_dwin_set_cbrush** (WINDOW win, CBRUSH * cbrush);
void	**xvt_dwin_set_clip** (WINDOW win, RCT * rctp);
void	**xvt_dwin_set_cpen** (WINDOW win, CPEN * cpen);
void	**xvt_dwin_set_draw_ctools** (WINDOW win, DRAW_CTOOLS * ctoolsp);
void	**xvt_dwin_set_draw_mode** (WINDOW win, DRAW_MODE mode);
void	**xvt_dwin_set_font** (WINDOW win, XVT_FNTID font_id);
void	**xvt_dwin_set_font_app_data** (WINDOW win, long app_data);
void	**xvt_dwin_set_font_family** (WINDOW win, char * family);
void	**xvt_dwin_set_font_native_desc** (WINDOW win, char * native_desc);
void	**xvt_dwin_set_font_size** (WINDOW win, long size);
void	**xvt_dwin_set_font_style** (WINDOW win, XVT_FONT_STYLE_MASK mask);
void	**xvt_dwin_set_fore_color** (WINDOW win, COLOR color);
void	**xvt_dwin_set_std_cbrush** (WINDOW win, long flag);
void	**xvt_dwin_set_std_cpen** (WINDOW win, long flag);
void	**xvt_dwin_update** (WINDOW win);

CBRUSH Type

```
typedef struct s_cbrush {
    PAT_STYLE pat;
    COLOR color;
} CBRUSH;
```

CPEN Type

```
typedef struct s_cpen {
    short width;
    PAT_STYLE pat;
    PEN_STYLE style;
    COLOR color;
} CPEN;
```

DRAW_CTOOLS Type

```
typedef struct s_drawct {
    CPEN pen;                   /* color pen */
    CBRUSH brush;               /* color brush */
    DRAW_MODE mode;             /* drawing mode */
    COLOR fore_color;           /* foreground color */
    COLOR back_color;           /* background color */
    BOOLEAN opaque_text;        /* is text drawn opaquely? */
} DRAW_CTOOLS;
```

DRAW_MODE Enumeration Type

```
typedef enum e_mode {           /* drawing (transfer) mode */
    M_COPY,                     /* patCopy */
    M_OR,                       /* patOr */
    M_XOR,                      /* patXor */
    M_CLEAR,                    /* patBic */
    M_NOT_COPY,                 /* notPatCopy */
    M_NOT_OR,                   /* notPatOr */
    M_NOT_XOR,                  /* notPatXor */
    M_NOT_CLEAR                 /* notPatBic */
} DRAW_MODE;
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

PAT_STYLE Enumeration Type

```
typedef enum e_pat {
    PAT_NONE,
    PAT_HOLLOW,
    PAT_SOLID,
    PAT_HORZ,
    PAT_VERT,
    PAT_FDIAG,
    PAT_BDIAG,
    PAT_CROSS,
    PAT_DIAGCROSS,
    PAT_RUBBER,
    PAT_SPECIAL
} PAT_STYLE;
```

PEN_STYLE Enumeration Type

```
typedef enum e_pen_style {
    P_SOLID,
    P_DOT,
    P_DASH
} PEN_STYLE;
```

PNT Type

```
typedef struct s_pnt {                  /* mathematical point */
    short v;                            /* vertical (y) coordinate */
    short h;                            /* horizontal (x) coordinate */
} PNT;
```

Tool Brush Constants (xvt_dwin_set_std_cbrush flags)

```
#define TL_BRUSH_BLACK ...
#define TL_BRUSH_WHITE ...
```

Tool Pen Constants (xvt_dwin_set_std_cpen flags)

```
#define TL_PEN_BLACK ...
#define TL_PEN_HOLLOW ...
#define TL_PEN_RUBBER ...
#define TL_PEN_WHITE ...
```

Error Handling Facility

See Also: Debugging Facility, Visible Objects (Error Handling Attributes)

const char * **xvt_errmsg_get_api_name** (XVT_ERRMSG errmsg);
const char * **xvt_errmsg_get_cat_text** (XVT_ERRMSG errmsg);
long int **xvt_errmsg_get_code_line** (XVT_ERRMSG errmsg);
const char * **xvt_errmsg_get_code_file** (XVT_ERRMSG errmsg);
XVT_ERRID **xvt_errmsg_get_msg_id** (XVT_ERRMSG errmsg);
const char * **xvt_errmsg_get_msg_text** (XVT_ERRMSG errmsg);
WINDOW **xvt_errmsg_get_req_object** (XVT_ERRMSG errmsg);
XVT_ERRSEV **xvt_errmsg_get_sev_id** (XVT_ERRMSG errmsg);
const char * **xvt_errmsg_get_sev_text** (XVT_ERRMSG errmsg;
const char * **xvt_errmsg_get_text** (XVT_ERRMSG errmsg, XVT_ERRID msg_id, char * buf, long int bufsiz);
WINDOW **xvt_errmsg_get_tgt_object** (XVT_ERRMSG errmsg);
void **xvt_errmsg_pop_handler** (XVT_ERRMSG_HANDLER handler);
void **xvt_errmsg_push_handler** (XVT_ERRMSG_HANDLER handler, DATA_PTR context);

#define **xvt_errmsg_sig** (win, sev, cat, suffix, num, text) ...
#define **xvt_errmsg_sig_if** (cond, win, sev, cat, suffix, num, text) ...
#define **xvt_errmsg_sig_std** (win, sev, msg_id) ...
#define **xvt_errmsg_sig_std_if** (cond, win, sev, msg_id) ...

Error Categories Used to Define and Group Error Categories

#define ERR_APP .... /* Application errors */
#define ERR_ARG ... /* Invalid argument */
#define ERR_ASSERT ... /* XVT Release 3 assert */
#define ERR_EMF ... /* Error messaging facility */
#define ERR_ERRNO ... /* XVT_ERRNO set */
#define ERR_FAIL ... /* Correctly requested operation failed */
#define ERR_REQ ... /* Invalid request or request context */
#define ERR_SYS ... /* Underlying system generated error*/

Error Categories Used to Define and Group Error Messages

#define ERR_ARG_FORMAT ... /* Wrong argument format */
#define ERR_ARG_INCOMP ... /* Incomplete argument */
#define ERR_ARG_INV ... /* Invalid object */
#define ERR_ARG_NULL ... /* NULL handle */
#define ERR_ARG_SIZE ... /* Bad size */
#define ERR_ARG_TYPE ... /* Incorrect object type */
#define ERR_ARG_VALUE ... /* Bad value */
#define ERR_ASSERT_0 ... /* Signaled assert0 */
#define ERR_ASSERT_2 ... /* Signaled assert2 */
#define ERR_ASSERT_4 ... /* Signaled assert4 */
#define ERR_EMF_FRAME ... /* Error message facility frame */
#define ERR_EMF_LABEL ... /* Error message facility label */
#define ERR_EMF_SEV ... /* Error message facility severity*/
#define ERR_FAIL_ACCESS ... /* Failed to access */
#define ERR_FAIL_ADD ... /* Failed to add */
#define ERR_FAIL_CREATE ... /* Failed to create */
#define ERR_FAIL_DEVICE ... /* Device driver failure */
#define ERR_FAIL_FIND ... /* Failed to find */
#define ERR_FAIL_INTERN ... /* Internal error */
#define ERR_FAIL_IO ... /* Failed I/O operation */
#define ERR_FAIL_NOMEM ... /* Failed to allocate memory */
#define ERR_FAIL_PARSE ... /* Parse failure */
#define ERR_REQ_AFTER... /* Request after xvt_app_create */
#define ERR_REQ_ATTR ... /* Operation target attribute is wrong */

```
#define ERR_REQ_DURING ...          /* Request invalid during operation */
#define ERR_REQ_INUSE ...           /* Operation target is in use */
#define ERR_REQ_RECURSIVE ...       /* Recursive request */
#define ERR_REQ_STATE ...           /* Operation target is in wrong state */
#define ERR_SYS_OS ...              /* Operating system error */
#define ERR_SYS_TEDIT ...           /* Internal text edit object error */
#define ERR_SYS_TK ...              /* Portability Toolkit error */
#define ERR_SYS_WS ...              /* Window system error */
```

XVT_ERRID Type

```
/* Error messaging facility data type */
typedef unsigned long XVT_ERRID;    /* error message identifier */
```

*XVT_ERRMSG Type

```
/* Error messaging facility data type */
typedef struct {...} * XVT_ERRMSG;  /* error message object handle*/
```

*XVT_ERRMSG_HANDLER Function Prototype

```
/* Error message handler */
typedef BOOLEAN (* XVT_ERRMSG_HANDLER) (
    XVT_ERRMSG err_msg,             /* message object handle */
    DATA_PTR context                /* context from push */
);
```

use prototype:

```
BOOLEAN XVT_CALLCONV1 errmsg_handler (XVT_ERRMSG err_msg, DATA_PTR context);
```

XVT_ERRSEV Enumeration Type

```
/* Error messaging facility data type */
typedef enum e_xvt_errsev {         /* error severity */
    SEV_NONE = 0,                   /* undefined */
    SEV_WARNING,                    /* no functionality loss */
    SEV_ERROR,                      /* failure, but may continue */
    SEV_FATAL                       /* must terminate execution */
} XVT_ERRSEV;
```

Events

See Also: Application Objects, Dialogs, Visible Objects (Event Attributes), Windows

```
void        xvt_app_process_pending_events  (void);

XVT_FNTID   xvt_event_get_font   (EVENT * ep);
BOOLEAN     xvt_event_is_virtual_key   (EVENT * ep);
void        xvt_event_set_font   (EVENT * ep, XVT_FNTID font_id);
```

CONTROL_INFO Type

```
typedef struct s_ctlinfo {                  /* info passed with E_CONTROL event */
    WIN_TYPE type;                          /* WC_* control type */
    WINDOW win;                             /* WINDOW id of the control being operated */
    union {
        struct s_pushbutton {
            int reserved;                   /* reserved...no usage yet */
        } pushbutton;
        struct s_radiobutton {
            int reserved;                   /* reserved...no usage yet */
        } radiobutton;
        struct s_checkbox {
            int reserved;                   /* reserved...no usage yet */
        } checkbox;
        struct s_scroll {                   /* scrollbar action */
            SCROLL_CONTROL what;            /* site of activity */
            short pos;                      /* thumb position */
        } scroll;
        struct s_edit {
            BOOLEAN focus_change;           /* is event a focus change? */
            BOOLEAN active;                 /* if so, gaining focus? (vs. losing) */
        } edit;
        struct s_statictext {
             int reserved;                  /* reserved...no usage yet */
        } statictext;
        struct s_lbox {                     /* list box action */
            BOOLEAN dbl_click;              /* double click (vs. single)? */
        } lbox;
        struct s_listbutton {
            int reserved;                   /* reserved...no usage yet.*/
        } listbutton;
        struct s_listedit {
            BOOLEAN focus_change;           /* did the edit field part change focus? */
            BOOLEAN active;                 /* if so, focus gained (vs. lost)? */
        } listedit;
        struct s_groupbox {
            int reserved;                   /* reserved...no usage yet */
        } groupbox;
        struct s_textedit {
            BOOLEAN focus_change;
            BOOLEAN active;
        } textedit;
        struct s_treeview {
            XVT_TREEVIEW_NODE node;         /* Node */
            BOOLEAN sgl_clik;               /* Single click */
            BOOLEAN dbl_click;              /* Double click */
            BOOLEAN expanded;               /* Node was expanded */
            BOOLEAN collapsed;              /* Node was collapsed */
        } treeview;
        struct s_icon {
            int reserved;                   /* reserved...no usage yet */
        } icon;
    } v;
} CONTROL_INFO;
```

EVENT Type

```
typedef struct s_event {
    EVENT_TYPE type;
    union {
        struct s_mouse {                    /* E_MOUSE_DOWN, E_MOUSE_UP,
                                               E_MOUSE_MOVE, E_MOUSE_DBL,
                                               E_MOUSE_SCROLL */
            PNT where;                      /* location of event (window relative) */
            BOOLEAN shift;                  /* shift key down? */
            BOOLEAN control;                /* control or option key down? */
            short button;                   /* button number */
            XVT_INT 32 scroll_x             /* scroll delta in the X axis - _MOUSE_SCROLL only */
            XVT_INT 32 scroll_y             /* scroll delta in the Y axis - _MOUSE_SCROLL only */
        } mouse;
        struct s_char {                     /* E_CHAR */
            XVT_WCHAR ch;                   /* wide character */
            BOOLEAN shift;                  /* shift key down? */
            BOOLEAN control;                /* control or option key down? */
            BOOLEAN virtual_key;            /* ch contains virtual key or not? */
            unsigned long modifiers;        /* bit field of key modifiers */
        } chr;
        BOOLEAN active;                     /* E_FOCUS: activation? (vs. deactivation) */
        BOOLEAN query;                      /* E_QUIT: query only? (app calls quit_OK) */
        struct s_scroll_info {              /* E_VSCROLL, E_HSCROLL */
            SCROLL_CONTROL what;            /* site of activity */
            short pos;                      /* thumb position, if SC_THUMB */
        } scroll;
        struct s_cmd {                      /* E_COMMAND */
            MENU_TAG tag;                   /* menu item tag */
            BOOLEAN shift;                  /* shift key down? */
            BOOLEAN control;                /* control or option key down? */
        } cmd;
        struct s_size {                     /* E_SIZE */
            short height;                   /* new height */
            short width;                    /* new width */
        } size;
        struct s_efont {                    /* E_FONT */
            XVT_FNTID font_id;              /* font id of selected font */
            } font;
        struct s_ctl {                      /* E_CONTROL */
            short id;                       /* control's ID */
            CONTROL_INFO ci;                /* control info */
        } ctl;
        struct s_update {                   /* E_UPDATE */
            RCT rct;                        /* update rectangle */
        } update;
        struct s_timer {                    /* E_TIMER */
            long id;                        /* timer ID */
        } timer;
        struct s_cxo {                      /* E_CXO */
            long msg_id;                    /* CXO message ID*/
            void * ptr;                     /* message data pointer */
        } cxo;
        struct s_user {                     /* E_USER */
            long id;                        /* application ID */
            void * ptr;                     /* application pointer */
        } user;
        struct s_help {                     /* E_HELP */
            WINDOW obj;                     /* help for control, window, dialog */
            MENU_TAG tag;                   /* help for menu item */
            XVT_HELP_TID tid;               /* predefined help topic */
```

```
        } help;
    } v;
} EVENT;
```

EVENT_HANDLER Function Prototype

```
typedef long (* EVENT_HANDLER) (WINDOW win, EVENT * ep);
```

use prototype:

```
long XVT_CALLCONV1 event_handler (WINDOW win, EVENT * ep);
```

*CXO_EVENT_HANDLER Function Prototype

```
typedef long (* XVT_CXO_EVENT) (XVT_CXO cxo, EVENT * ep);
```

use prototype:

```
long XVT_CALLCONV1 cxo_event_handler (XVT_CXO cxo, EVENT * ep);
```

EVENT_MASK Type

```
typedef unsigned long EVENT_MASK;          /* event delivery mask */
```

Event Mask Constants

```
#define EM_ALL ...
#define EM_CHAR ...
#define EM_CLOSE ...
#define EM_COMMAND ...
#define EM_CONTROL ...
#define EM_CREATE ...
#define EM_DESTROY ...
#define EM_FOCUS ...
#define EM_FONT ...
#define EM_HELP ...
#define EM_HSCROLL ...
#define EM_MOUSE_DBL ...
#define EM_MOUSE_DOWN ...
#define EM_MOUSE_MOVE ...
#define EM_MOUSE_SCROLL ...
#define EM_MOUSE_UP ...
#define EM_NONE ...
#define EM_QUIT ...
#define EM_SIZE ...
#define EM_TIMER ...
#define EM_UPDATE ...
#define EM_USER ...
#define EM_VSCROLL ...
```

EVENT_TYPE Type

```
typedef enum_event_type {
    E_CREATE,                      /* creation */
    E_DESTROY,                     /* destruction */
    E_FOCUS,                       /* window focus gain/loss */
    E_SIZE,                        /* resize */
    E_UPDATE,                      /* update */
    E_CLOSE,                       /* close window request */
    E_CXO,                         /* container extension object activity */
    E_MOUSE_DOWN,                  /* mouse down */
    E_MOUSE_UP,                    /* mouse up */
    E_MOUSE_MOVE,                  /* mouse move */
    E_MOUSE_DBL,                   /* mouse double click */
    E_MOUSE_SCROLL,                /* mouse scroll wheel event */
```

```
    E_CHAR,                     /* character typed */
    E_VSCROLL                   /* vertical window scrollbar activity */
    E_HSCROLL,                  /* horizontal window scrollbar activity */
    E_COMMAND,                  /* menu command */
    E_FONT,                     /* font menu selection */
    E_CONTROL,                  /* control activity */
    E_TIMER,                    /* timer */
    E_QUIT,                     /* application shutdown request */
    E_HELP,                     /* help invoked */
    E_USER,                     /* user defined */
} EVENT_TYPE;
```

Modifier Key Constants

```
#define XVT_MOD_KEY_ALT ...
#define XVT_MOD_KEY_CMD ...
#define XVT_MOD_KEY_COMPOSE ...
#define XVT_MOD_KEY_CTL ...
#define XVT_MOD_KEY_LSHIFT ...
#define XVT_MOD_KEY_NONE ...
#define XVT_MOD_KEY_OPTION ...
#define XVT_MOD_KEY_RSHIFT ...
#define XVT_MOD_KEY_SHIFT ...
```

PNT Type

```
typedef struct s_pnt {          /* mathematical point */
    short v;                    /* vertical (y) coordinate */
    short h;                    /* horizontal (x) coordinate */
} PNT;
```

RCT Type

```
typedef struct s_rct {          /* mathematical rectangle */
    short top;                  /* top coordinate */
    short left;                 /* left coordinate */
    short bottom;               /* bottom coordinate */
    short right;                /* right coordinate */
} RCT;
```

File System

```
FILE_SPEC*      xvt_app_get_file   (void);
void            xvt_app_get_files_count   (BOOLEAN * printp, int * countp);
void            xvt_app_set_file_processed   (void);

BOOLEAN         xvt_fsys_build_pathname   (char * mbs, const char * volname,
                    const char * dirname, const char * leafroot, const char * leafext,
                    const char * leafvers);
BOOLEAN         xvt_fsys_convert_dir_to_str   (DIRECTORY * dirp, char * path, int sz_path);
BOOLEAN         xvt_fsys_convert_str_to_dir   (char * path, DIRECTORY * dirp);
void            xvt_fsys_get_default_dir   (DIRECTORY * dirp);
BOOLEAN         xvt_fsys_get_dir   (DIRECTORY * dirp);
long            xvt_fsys_get_file_attr   (FILE_SPEC * file, long attr);
SLIST           xvt_fsys_list_files   (char * type, char * pat, BOOLEAN dirs);
BOOLEAN         xvt_fsys_parse_pathname   (const char * mbs, char * volname, char * dirname,
                    char * leafroot, char * leafext, char * leafvers);
BOOLEAN         xvt_fsys_rem_file   (FILE_SPEC * file);
void            xvt_fsys_restore_dir   (void);
void            xvt_fsys_save_dir   (void);
BOOLEAN         xvt_fsys_set_dir   (DIRECTORY * dirp);
void            xvt_fsys_set_dir_startup   (void);
BOOLEAN         xvt_fsys_set_file_attr   (FILE_SPEC * file, long attr, long value);
```

Constant for xvt_fsys_list_files

```
#define DIR_TYPE ...                    /* file type indicates list directories */
```

DIRECTORY Type

```
typedef struct s_dir {...} DIRECTORY;
```

File Attribute Constants

```
#define XVT_FILE_ATTR_ATIME ...
#define XVT_FILE_ATTR_CREATORSTR ...
#define XVT_FILE_ATTR_CTIME ...
#define XVT_FILE_ATTR_DIRECTORY ...
#define XVT_FILE_ATTR_DIRSTR ...
#define XVT_FILE_ATTR_EXECUTE ...
#define XVT_FILE_ATTR_EXIST ...
#define XVT_FILE_ATTR_FILESTR ...
#define XVT_FILE_ATTR_MAXIMUM ...
#define XVT_FILE_ATTR_MINIMUM ...
#define XVT_FILE_ATTR_MTIME ...
#define XVT_FILE_ATTR_NUMLINKS ...
#define XVT_FILE_ATTR_TYPESTR ...
#define XVT_FILE_ATTR_READ ...
#define XVT_FILE_ATTR_SIZE ...
#define XVT_FILE_ATTR_WRITE ...
```

File Size Constants

```
#define SZ_FNAME ...                    /* filenames size including path */
#define SZ_LEAFNAME ...                 /* leafname size */
```

FILE_SPEC Type

```
typedef struct s_filespec {             /* file specification */
    DIRECTORY dir;                      /* directory */
    char type[6];                       /* file type or extension */
    char name[SZ_FNAME + 1];            /* file name (may be partial path) */
    char creator[6];                    /* file creator */
} FILE_SPEC;
```

Focus

See Also: Navigation Objects

WINDOW	**xvt_scr_get_focus_topwin** (void);
WINDOW	**xvt_scr_get_focus_vobj** (void);
void	**xvt_scr_set_focus_vobj** (WINDOW win);
BOOLEAN	**xvt_vobj_is_focusable** (WINDOW win);

Fonts

See Also: Controls, Drawing, Visible Objects (Font Attributes)

long	**xvt_fmap_get_families** (PRINT_RCD * precp, char ** family_array, long max_families);
long	**xvt_fmap_get_family_sizes** (PRINT_RCD * precp, char * family, long * size_array, BOOLEAN *scalable, long max_sizes);
long	**xvt_fmap_get_family_styles** (PRINT_RCD * precp, char * family, XVT_FONT_STYLE_MASK * style_array, long max_styles);
long	**xvt_fmap_get_familysize_styles** (PRINT_RCD * precp, char * family, long size, XVT_FONT_STYLE_MASK * style_array, long max_styles);
long	**xvt_fmap_get_familystyle_sizes** (PRINT_RCD * precp, char * family, XVT_FONT_STYLE_MASK style, long * size_array, BOOLEAN *scalable, long max_sizes);
void	**xvt_font_copy** (XVT_FNTID dest_font_id, XVT_FNTID src_font_id, XVT_FONT_ATTR_MASK mask);
XVT_FNTID	**xvt_font_create** (void);
BOOLEAN	**xvt_font_deserialize** (XVT_FNTID font_id, char * buf);
void	**xvt_font_destroy** (XVT_FNTID font_id);
long	**xvt_font_get_app_data** (XVT_FNTID font_id);
BOOLEAN	**xvt_font_get_family** (XVT_FNTID font_id, char * buf, long max_buf);
BOOLEAN	**xvt_font_get_family_mapped** (XVT_FNTID font_id, char * buf, long max_buf);
void	**xvt_font_get_metrics** (XVT_FNTID font_id, int * leadingp, int * ascentp, int * descentp);
BOOLEAN	**xvt_font_get_native_desc** (XVT_FNTID font_id, char * buf, long max_buf);
long	**xvt_font_get_size** (XVT_FNTID font_id);
long	**xvt_font_get_size_mapped** (XVT_FNTID font_id);
XVT_FONT_STYLE_MASK	**xvt_font_get_style** (XVT_FNTID font_id);
XVT_FONT_STYLE_MASK	**xvt_font_get_style_mapped** (XVT_FNTID font_id);
WINDOW	**xvt_font_get_win** (XVT_FNTID font_id);
BOOLEAN	**xvt_font_has_valid_native_desc** (XVT_FNTID font_id);
BOOLEAN	**xvt_font_is_mapped** (XVT_FNTID font_id);
BOOLEAN	**xvt_font_is_print** (XVT_FNTID font_id);
BOOLEAN	**xvt_font_is_scalable** (XVT_FNTID font_id);
BOOLEAN	**xvt_font_is_valid** (XVT_FNTID font_id);
void	**xvt_font_map** (XVT_FNTID font_id, WINDOW win);
void	**xvt_font_map_using_default** (XVT_FNTID font_id);
long	**xvt_font_serialize** (XVT_FNTID font_id, char * buf, long max_buf);

void **xvt_font_set_app_data** (XVT_FNTID font_id, long app_data);
void **xvt_font_set_family** (XVT_FNTID font_id, char * family);
void **xvt_font_set_native_desc** (XVT_FNTID font_id, char * native_desc);
void **xvt_font_set_size** (XVT_FNTID font_id, long size);
void **xvt_font_set_style** (XVT_FNTID font_id, XVT_FONT_STYLE_MASK mask);
void **xvt_font_unmap** (XVT_FNTID font_id);

Font Attribute Constants

#define XVT_FA_ALL ...
#define XVT_FA_APP_DATA ...
#define XVT_FA_FAMILY ...
#define XVT_FA_NATIVE ...
#define XVT_FA_SIZE ...
#define XVT_FA_STYLE ...
#define XVT_FA_WIN ...

Font Family Constants (Guaranteed Portable Support)

#define XVT_FFN_COURIER ...
#define XVT_FFN_FIXED ...
#define XVT_FFN_HELVETICA ...
#define XVT_FFN_SYSTEM ...
#define XVT_FFN_TIMES ...

Font Style Constants

#define XVT_FS_BLINK ...
#define XVT_FS_BOLD ...
#define XVT_FS_INVERSE ...
#define XVT_FS_ITALIC ...
#define XVT_FS_NONE ...
#define XVT_FS_OUTLINE ...
#define XVT_FS_SHADOW ...
#define XVT_FS_STRIKEOUT ...
#define XVT_FS_UNDERLINE ...
#define XVT_FS_USER1 ...
#define XVT_FS_USER2 ...
#define XVT_FS_USER3 ...
#define XVT_FS_USER4 ...
#define XVT_FS_USER5 ...
#define XVT_FS_WILDCARD ...

NULL Font ID Constant

#define NULL_FNTID ... /* convenience macro for identifying a null font id */

*XVT_FNTID Type

typedef struct {...} * XVT_FNTID; /* font object handle */

XVT_FONT_ATTR_MASK Type

typedef unsigned long XVT_FONT_ATTR_MASK;

*XVT_FONT_DIALOG Function Prototype

```
/* Prototype for the application supplied font selection dialog */
typedef BOOLEAN (* XVT_FONT_DIALOG) (
    WINDOW win,                         /* window to send E_FONT event to */
    XVT_FNTID font_id,                  /* default font id */
    PRINT_RCD * precp,                  /* print record or NULL */
    unsigned long reserved              /* reserved */
 );
```

use prototype:

```
BOOLEAN XVT_CALLCONV1 xvt_font_dialog (WINDOW win, XVT_FNTID font_id,
    PRINT_RCD * precp, unsigned long reserved);
```

*XVT_FONT_MAPPER Function Prototype

```
/* Prototype for the application supplied font mapper */
typedef BOOLEAN (* XVT_FONT_MAPPER) (XVT_FNTID font_id);
```

use prototype:

```
BOOLEAN XVT_CALLCONV1 font_mapper (XVT_FNTID font_id);
```

XVT_FONT_STYLE_MASK Type

```
typedef unsigned long XVT_FONT_STYLE_MASK;
```

Global Memory

See: Memory

Help System

See Also: XRC Statements (Predefined Help IDs)

void	**xvt_help_assoc_all** (XVT_HELP_INFO hi, WINDOW win, long rid, WIN_DEF * wdef);
void	**xvt_help_begin_objclick** (XVT_HELP_INFO hi, WINDOW win, unsigned long flags);
void	**xvt_help_close_helpfile** (XVT_HELP_INFO hi);
void	**xvt_help_disassoc_all** (XVT_HELP_INFO hi, WINDOW win);
void	**xvt_help_display_topic** (XVT_HELP_INFO hi, XVT_HELP_TID topic);
void	**xvt_help_end_objclick** (XVT_HELP_INFO hi);
XVT_HELP_FLAVOR	**xvt_help_get_flavor** (void);
XVT_HELP_TID	**xvt_help_get_menu_assoc** (XVT_HELP_INFO hi, WINDOW win, MENU_TAG tag);
XVT_HELP_TID	**xvt_help_get_win_assoc** (XVT_HELP_INFO hi, WINDOW win);
XVT_HELP_INFO	**xvt_help_open_helpfile** (FILE_SPEC * hfile, unsigned long flags);
BOOLEAN	**xvt_help_process_event** (XVT_HELP_INFO hi, WINDOW win, EVENT * ev);
void	**xvt_help_search_topic** (XVT_HELP_INFO hi, XVT_HELP_TID topic, char * pattern);
void	**xvt_help_set_menu_assoc** (XVT_HELP_INFO hi, WINDOW win, MENU_TAG tag, XVT_HELP_TID topic, unsigned long flags);
void	**xvt_help_set_win_assoc** (XVT_HELP_INFO hi, WINDOW win, XVT_HELP_TID topic, unsigned long flags);

HSF_* Help File Open Constants

```
#define HSF_APPNAME_TITLE ...          /* show application name in title */
#define HSF_INDEX_ON_DISK ...          /* default value is in-memory */
#define HSF_NO_BEEP_MODAL ...          /* don't beep for help on modal dialog */
#define HSF_NO_HELPMENU_ASSOC ...      /* do not associate topics to helpmenu */
#define HSF_NO_TOPIC_WARNING ...       /* no warning for missing topics */
```

NULL Topic ID Constant

```
#define NULL_TID ...                   /* convenience macro for identifying a null help topic id */
```

XVT_HELP_FLAVOR Enumeration Type

```
typedef enum e_xvt_help_flavor {
    XVT_HELP_FLAVOR_NONE,              /* no viewer */
    XVT_HELP_FLAVOR_NTVSRV,            /* native standalone (server) viewer */
    XVT_HELP_FLAVOR_NTVBND,            /* native bound viewer */
    XVT_HELP_FLAVOR_PORTSRV,           /* portable standalone (server) viewer */
    XVT_HELP_FLAVOR_PORTBND            /* portable bound viewer */
} XVT_HELP_FLAVOR;
```

XVT_HELP_INFO Type

```
typedef struct s_xvt_help_info {...} XVT_HELP_INFO;
```

XVT_HELP_TID Type

```
typedef long XVT_HELP_TID;
```

HTML Control

char *	**xvt_html_get_xrc** (WINDOW win, char *xrc, int sz_xrc);
void	**xvt_html_set_xrc** (WINDOW win, char *xrc);
XVT_HTML_XRC_INTERCEPT_HANDLER	**xvt_get_xrc_intercept** (WINDOW win);
void	**xvt_set_xrc_intercept** (WINDOW win, XVT_HTML_XRC_INTERCEPT_HANDLER fcn);
void	**xvt_html_refresh** (WINDOW win);
void	**xvt_html_home** (WINDOW win);
void	**xvt_html_back** (WINDOW win);
void	**xvt_html_forward** (WINDOW win);
void	**xvt_html_search** (WINDOW win);
void	**xvt_html_stop** (WINDOW win);

XVT_HTML_XRC_INTERCEPT_HANDLER Function Prototype

```
typedef BOOLEAN (*XVT_HTML_XRC_INTERCEPT_HANDLER) (WINDOW win, char **xrc);
```

Image Objects

See Also: Colors, Drawing, Input/Output Streams, Palettes, Picture Objects, Pixmap Objects

XVT_IMAGE	**xvt_image_create** (XVT_IMAGE_FORMAT format, short width, short height, XVT_IMAGE_ATTR reserved);
void	**xvt_image_destroy** (XVT_IMAGE image);
XVT_IMAGE	**xvt_image_duplicate** (XVT_IMAGE image);
void	**xvt_image_fill_rect** (XVT_IMAGE image, COLOR color, RCT * rectp);
COLOR	**xvt_image_get_clut** (XVT_IMAGE image, short index);
void	**xvt_image_get_dimensions** (XVT_IMAGE image, short * width, short * height);
XVT_IMAGE_FORMAT	**xvt_image_get_format** (XVT_IMAGE image);

```
void        xvt_image_get_from_pmap  (XVT_IMAGE dst_image,
                XVT_PIXMAP src_pixmap, RCT * dst_rect, RCT * src_rect);
short       xvt_image_get_ncolors  (XVT_IMAGE image);
COLOR       xvt_image_get_pixel  (XVT_IMAGE image, short x, short y);
void        xvt_image_get_resolution  (XVT_IMAGE image, long * hres, long * vres);
DATA_PTR    xvt_image_get_scanline  (XVT_IMAGE image, short linenum);
XVT_IMAGE   xvt_image_read  (char * filename);
XVT_IMAGE   xvt_image_read_bmp  (char * filename);
XVT_IMAGE   xvt_image_read_bmp_from_iostr  (XVT_IOSTREAM iostr);
XVT_IMAGE   xvt_image_read_macpict  (char * filename);
XVT_IMAGE   xvt_image_read_macpict_from_iostr  (XVT_IOSTREAM iostr);
XVT_IMAGE   xvt_image_read_xbm  (char * filename);
XVT_IMAGE   xvt_image_read_xbm_from_iostr  (XVT_IOSTREAM iostr);
XVT_IMAGE   xvt_image_read_xpm  (char * filename);
XVT_IMAGE   xvt_image_read_xpm_from_iostr  (XVT_IOSTREAM iostr);
void        xvt_image_set_clut  (XVT_IMAGE image, short index, COLOR color);
void        xvt_image_set_ncolors  (XVT_IMAGE image, short ncolors);
void        xvt_image_set_pixel  (XVT_IMAGE image, short x, short y, COLOR color);
void        xvt_image_set_resolution  (XVT_IMAGE image, long hres, long vres);
void        xvt_image_transfer  (XVT_IMAGE dst_image, XVT_IMAGE src_image,
                RCT * dst_rect, RCT * src_rect);
BOOLEAN     xvt_image_write_bmp_to_iostr  (XVT_IMAGE image, XVT_IOSTREAM iostr);
BOOLEAN     xvt_image_write_macpict_to_iostr  (XVT_IMAGE image,
                XVT_IOSTREAM iostr);
```

Color Lookup Table Size Constant

```
#define XVT_CLUT_SIZE ...          /* maximum colors in an image color lookup table */
```

NULL Image Handle Constant

```
#define NULL_IMAGE ...
```

RCT Type

```
typedef struct s_rct {             /* mathematical rectangle */
    short top;                     /* top coordinate */
    short left;                    /* left coordinate */
    short bottom;                  /* bottom coordinate */
    short right;                   /* right coordinate */
} RCT;
```

*XVT_IMAGE Type

```
typedef struct {...} * XVT_IMAGE;
```

*XVT_IMAGE_ATTR Type

```
typedef struct {...} * XVT_IMAGE_ATTR;
```

XVT_IMAGE_FORMAT Enumeration Type

```
typedef enum {
    XVT_IMAGE_NONE,
    XVT_IMAGE_CL8,
    XVT_IMAGE_RGB,
    XVT_IMAGE_MONO
} XVT_IMAGE_FORMAT;
```

Input/Output Streams

See Also: Image Objects

```
XVT_IOSTREAM        xvt_iostr_create_fread   (FILE * filep);
XVT_IOSTREAM        xvt_iostr_create_fwrite  (FILE * filep);
XVT_IOSTREAM        xvt_iostr_create_read    (XVT_IOSTR_CONTEXT context,
                        XVT_IOSTR_RWFUNC get_bytes, XVT_IOSTR_SZFUNC num_bytes);
XVT_IOSTREAM        xvt_iostr_create_write   (XVT_IOSTR_CONTEXT context,
                        XVT_IOSTR_RWFUNC put_bytes);
void                xvt_iostr_destroy   (XVT_IOSTREAM iostream);
XVT_IOSTR_CONTEXTxvt_iostr_get_context   (XVT_IOSTREAM iostream);
```

XVT_IOSTR_CONTEXT Type

```
/* Type for IOSTREAM interface */
typedef DATA_PTR XVT_IOSTR_CONTEXT;
```

*XVT_IOSTR_RWFUNC Function Prototype

```
/* Type for IOSTREAM interface */
typedef short (* XVT_IOSTR_RWFUNC) (XVT_IOSTREAM iostr, unsigned short nbytes, char * buf);
```

use prototype:

```
short XVT_CALLCONV1 xvt_iostr_rwfunc (XVT_IOSTREAM iostr, unsigned short nbytes, char * buf);
```

*XVT_IOSTR_SZFUNC Function Prototype

```
/* Type for IOSTREAM interface */
typedef long (* XVT_IOSTR_SZFUNC) (XVT_IOSTREAM iostr);
```

use prototype:

```
long XVT_CALLCONV1 xvt_iostr_szfunc (XVT_IOSTREAM iostr);
```

*XVT_IOSTREAM Type

```
/* Type for IOSTREAM interface */
typedef struct {...} * XVT_IOSTREAM;
```

Keyboard Focus

See: Focus

Keyboard Navigation

See: Navigation Objects

Key Codes

```
#define K_BTAB ...          /* back tab */
#define K_DEL ...           /* delete (same as ASCII) */
#define K_DOWN ...          /* down arrow */
#define K_CLEAR ...         /* clear */
#define K_COPY ...          /* copy */
#define K_CUT ...           /* cut */
#define K_END ...           /* end */
#define K_F1 ...            /* function key 1 */
#define K_F2 ...
```

```
#define K_F3 ...
#define K_F4 ...
#define K_F5 ...
#define K_F6 ...
#define K_F7 ...
#define K_F8 ...
#define K_F9 ...
#define K_F10 ...
#define K_F11 ...
#define K_F12 ...
#define K_F13 ...
#define K_F14 ...
#define K_F15 ...
#define K_F16 ...
#define K_F17 ...
#define K_F18 ...
#define K_F19 ...
#define K_F20 ...
#define K_F21 ...
#define K_F22 ...
#define K_F23 ...
#define K_F24 ...
#define K_HELP ...                  /* help */
#define K_HOME ...                  /* home */
#define K_INS ...                   /* insert */
#define K_KP0 ...                   /* keypad '0' */
#define K_KP1 ...
#define K_KP2 ...
#define K_KP3 ...
#define K_KP4 ...
#define K_KP5 ...
#define K_KP6 ...
#define K_KP7 ...
#define K_KP8 ...
#define K_KP9 ...                   /* keypad '9' */
#define K_KPADD ...                 /* keypad '+' */
#define K_KPDIV ...                 /* keypad '/' */
#define K_KPMULT ...                /* keypad '*' */
#define K_KPSUB ...                 /* keypad '-' */
#define K_LEFT ...                  /* left arrow */
#define K_LEND ...                  /* line end */
#define K_LHOME ...                 /* line home */
#define K_NEXT ...                  /* next screen */
#define K_PASTE ...                 /* paste */
#define K_PREV ...                  /* previous screen */
#define K_RIGHT ...                 /* right arrow */
#define K_WLEFT ...                 /* word left */
#define K_WRIGHT ...                /* word right */
#define K_UP ...                    /* up arrow */
```

List Box, List Edit Controls

See Also: Controls, Visible Objects

```
void        xvt_ctl_get_text_sel  (WINDOW win, int * first, int * last);
void        xvt_ctl_set_text_sel  (WINDOW win, int first, int last);

BOOLEAN     xvt_list_add   (WINDOW win, int index, char * sx);
BOOLEAN     xvt_list_clear   (WINDOW win);
```

```
int         xvt_list_count_all   (WINDOW win);
int         xvt_list_count_sel   (WINDOW win);
SLIST       xvt_list_get_all   (WINDOW win);
BOOLEAN     xvt_list_get_elt   (WINDOW win, int index, char * s, int sz_s);
BOOLEAN     xvt_list_get_first_sel   (WINDOW win, char * s, int sz_s);
SLIST       xvt_list_get_sel   (WINDOW win);
int         xvt_list_get_sel_index   (WINDOW win);
BOOLEAN     xvt_list_is_sel   (WINDOW win, int index);
BOOLEAN     xvt_list_rem   (WINDOW win, int index);
void        xvt_list_resume   (WINDOW win);
BOOLEAN     xvt_list_set_sel   (WINDOW win, int index, BOOLEAN select);
void        xvt_list_suspend   (WINDOW win);
```

Lists

See: SLIST (String List) Objects

Memory

```
GHANDLE     xvt_gmem_alloc   (long size);
BOOLEAN     xvt_gmem_free   (GHANDLE h);
long        xvt_gmem_get_size   (GHANDLE h);
DATA_PTR    xvt_gmem_lock   (GHANDLE h);
GHANDLE     xvt_gmem_realloc   (GHANDLE h, long size);
BOOLEAN     xvt_gmem_unlock   (GHANDLE h);

DATA_PTR    xvt_mem_alloc   (size_t size);
DATA_PTR    xvt_mem_realloc   (DATA_PTR p, size_t size);
DATA_PTR    xvt_mem_rep   (DATA_PTR dst, DATA_PTR src, UINT srclen, long reps);
void        xvt_mem_free   (DATA_PTR p);
DATA_PTR    xvt_mem_zalloc   (size_t size);
```

GHANDLE Type

```
typedef unsigned long GHANDLE;          /* handle to global memory block */
```

XVT_MEM Function Prototype

```
/* Structure to contain memory management functions for use with attribute
   ATTR_MEMORY_MANAGER */
typedef struct s_mem {
    DATA_PTR  (* malloc)  (size_t size);
    void  (* free)  (DATA_PTR data);
    DATA_PTR  (* realloc)  (DATA_PTR data, size_t size);
    DATA_PTR  (* zalloc)  (size_t size);
} XVT_MEM;
```

use prototypes:

```
DATA_PTR  XVT_CALLCONV1  malloc  (size_t size);
void  XVT_CALLCONV1 free  (DATA_PTR data);
DATA_PTR  XVT_CALLCONV1  realloc  (DATA_PTR data, size_t size);
DATA_PTR  XVT_CALLCONV1  zalloc  (size_t size);
```

Menus

See Also: Windows

XVT_FNTID	**xvt_menu_get_font_sel** (WINDOW win);
MENU_ITEM *	**xvt_menu_get_tree** (WINDOW win);
BOOLEAN	**xvt_menu_popup** (MENU_ITEM * menu_p, WINDOW win, PNT pos, XVT_POPUP_ALIGNMENT alignment, MENU_TAG item);
void	**xvt_menu_set_font_sel** (WINDOW win, XVT_FNTID font_id);
void	**xvt_menu_set_item_checked** (WINDOW win, MENU_TAG tag, BOOLEAN check);
void	**xvt_menu_set_item_enabled** (WINDOW win, MENU_TAG tag, BOOLEAN enable);
void	**xvt_menu_set_item_title** (WINDOW win, MENU_TAG tag, char * text);
void	**xvt_menu_set_tree** (WINDOW win, MENU_ITEM * mip);
void	**xvt_menu_update** (WINDOW win);

DEFAULT_*_MENU Constants

#define DEFAULT_EDIT_MENU ...
#define DEFAULT_FILE_MENU ...
#define DEFAULT_FONT_MENU ...
#define DEFAULT_HELP_MENU ...

M_* Menu Tag Constants

#define M_FILE ...
#define M_FILE ...
#define M_FILE_NEW ...
#define M_FILE_OPEN ...
#define M_FILE_CLOSE ...
#define M_FILE_SAVE ...
#define M_FILE_SAVE_AS ...
#define M_FILE_REVERT ...
#define M_FILE_PG_SETUP ...
#define M_FILE_PRINT ...
#define M_FILE_QUIT ...
#define M_FILE_ABOUT ...

#define M_EDIT ...
#define M_EDIT_UNDO ...
#define M_EDIT_CUT ...
#define M_EDIT_COPY ...
#define M_EDIT_PASTE ...
#define M_EDIT_CLEAR ...
#define M_EDIT_SEL_ALL ...
#define M_EDIT_CLIPBOARD ...

#define M_FONT ...
#define M_STYLE ...

#define M_HELP ...
#define M_HELP_HELPMENU ...
#define M_HELP_ONCONTEXT ...
#define M_HELP_HELPONHELP ...
#define M_HELP_ONWINDOW ...
#define M_HELP_KEYBOARD ...
#define M_HELP_INDEX ...
#define M_HELP_TUTORIAL ...
#define M_HELP_SEARCH ...
#define M_HELP_OBJCLICK ...
#define M_HELP_VERSION ...
#define M_HELP_GOTO ...

```
#define M_HELP_GLOSSARY ...
#define M_HELP_CONTENTS ...
```

MENU_ITEM Type

```
typedef struct s_mitem {
    MENU_TAG tag;
    char * text;
    short mkey;
    unsigned enabled:1;
    unsigned checked:1;
    unsigned checkable:1;
    unsigned separator:1;
    struct s_mitem * child;
     ...                                /* platform-specific */
} MENU_ITEM;
```

PNT Type

```
typedef struct s_pnt {                  /* mathematical point */
    short v;                            /* vertical (y) coordinate */
    short h;                            /* horizontal (x) coordinate */
} PNT;
```

XVT_POPUP_ALIGNMENT Enumeration Type

```
typedef enum e_xvt_alignment {
    XVT_POPUP_CENTER,
    XVT_POPUP_LEFT_ALIGN,
    XVT_POPUP_RIGHT_ALIGN,
    XVT_POPUP_OVER_ITEM
} XVT_POPUP_ALIGNMENT;
```

MENU_TAG Type

```
typedef short MENU_TAG;                 /* menu item tag */
```

Miscellaneous

```
#define max   (x, y)   ...              /* only if not already defined */
#define min   (x, y)   ...              /* only if not already defined */
#define PTR_LONG   (p)   ...            /* cast pointer to long */
```

Data Types and Values

```
typedef char XVT_BYTE;
typedef unsigned char XVT_UBYTE;
typedef XVT_BYTE * DATA_PTR;
#define NULL ...
```

Escape Code Value Ranges

```
#define XVT_ESC_COMMON_BASE ...
#define XVT_ESC_INTERNAL_BASE ...
```

Machine-related Constants

```
#define CHAR_MAX ...
#define INT_MAX ...
#define LONG_MAX ...
#define SHRT_MAX ...
#define UCHAR_MAX ...
#define UINT_MAX ...
#define ULONG_MAX ...
#define USHRT_MAX ...
```

XVT Software Version Identifiers

```
#define XVT_HELP_VERSION ...          /* These constants define the version number */
#define XVT_HELP_VERSION_MAJOR ...    /* for the XVT Help system */
#define XVT_HELP_VERSION_MINOR ...
#define XVT_HELP_VERSION_PATCH ...

#define XVT_PTK_VERSION ...           /* These constants define the version number */
#define XVT_PTK_VERSION_MAJOR ...     /* for the XVT Portability Toolkit */
#define XVT_PTK_VERSION_MINOR ...
#define XVT_PTK_VERSION_PATCH ...

#define XVT_TX_VERSION ...            /* These constants define the version number */
#define XVT_TX_VERSION_MAJOR ...      /* for the Text Edit Object */
#define XVT_TX_VERSION_MINOR ...
#define XVT_TX_VERSION_PATCH ...
```

XVT-supported File System Identifiers (with Default Values)

```
#define XVT_FILESYS_MAC FALSE         /* Apple Macintosh file system */
#define XVT_FILESYS_HPFS FALSE        /* High Performance file system */
#define XVT_FILESYS_NTFS FALSE        /* Win32 file system */
#define XVT_FILESYS_UNIX FALSE        /* UNIX file system */
```

XVT-supported Window System Identifiers

```
#define MACWS ...                     /* Apple Macintosh */
#define WIN32WS ...                   /* Win32 systems*/
#define MTFWS ...                     /* Motif */
```

Navigation Objects

See Also: Focus

```
XVT_NAV     xvt_nav_create   (WINDOW win, SLIST win_list);
void        xvt_nav_destroy  (XVT_NAV nav);
BOOLEAN     xvt_nav_add_win  (XVT_NAV nav, WINDOW win, WINDOW refwin,
                              XVT_NAV_INSERTION where);
BOOLEAN     xvt_nav_rem_win  (XVT_NAV nav, WINDOW win);
SLIST       xvt_nav_list_wins (XVT_NAV nav);

XVT_NAV     xvt_win_get_nav  (WINDOW win);
```

XVT_NAV Type

```
typedef long XVT_NAV;
```

XVT_NAV_INSERTION Enumeration Type

```
typedef enum e_nav_insertion {
    XVT_NAV_POS_FIRST,
    XVT_NAV_POS_LAST,
    XVT_NAV_POS_BEFORE,
    XVT_NAV_POS_AFTER
} XVT_NAV_INSERTION;
```

Notebook Control

See Also: Windows

void	**xvt_notebk_add_page** (WINDOW notebk, short tab_no, short page_no, char * title, long page_data);
void	**xvt_notebk_add_tab** (WINDOW notebk, short tab_no, char * title, XVT_IMAGE image);
WINDOW	**xvt_notebk_create_face** (WINDOW notebk, short tab_no, short page_no, EVENT_MASK mask, EVENT_HANDLER face_eh, long app_data);
WINDOW	**xvt_notebk_create_face_def** (WINDOW notebk, short tab_no, short page_no, WIN_DEF * win_def_p, EVENT_MASK mask, EVENT_HANDLER face_eh, long app_data);
WINDOW	**xvt_notebk_create_face_res** (WINDOW notebk, short tab_no, short page_no, int rid, EVENT_MASK mask, EVENT_HANDLER face_eh, long app_data);
BOOLEAN	**xvt_notebk_enum_pages** (WINDOW notebk, XVT_NOTEBK_ENUM_PAGES func, long data);
WINDOW	**xvt_notebk_get_face** (WINDOW notebk, short tab_no, short page_no);
WINDOW	**xvt_notebk_get_front_page** (WINDOW notebk, short * tab_no, short * page_no);
short	**xvt_notebk_get_num_pages** (WINDOW notebk, short tab_no);
short	**xvt_notebk_get_num_tabs** (WINDOW notebk);
long	**xvt_notebk_get_page_data** (WINDOW notebk, short tab_no, short page_no);
void	**xvt_notebk_get_page_from_face** (WINDOW face, WINDOW * notebk, short * tab_no, short * page_no);
char *	**xvt_notebk_get_page_title** (WINDOW notebk, short tab_no, short page_no, char * buf, size_t size);
XVT_IMAGE	**xvt_notebk_get_tab_image** (WINDOW notebk, short tab_no);
char *	**xvt_notebk_get_tab_title** (WINDOW notebk, short tab_no, char * buf, size_t size);
void	**xvt_notebk_rem_page** (WINDOW notebk, short tab_no, short page_no);
void	**xvt_notebk_rem_tab** (WINDOW notebk, short tab_no);
void	**xvt_notebk_set_page_data** (WINDOW notebk, short tab_no, short page_no, long data);
void	**xvt_notebk_set_page_title** (WINDOW notebk, short tab_no, short page_no, char * title);
void	**xvt_notebk_set_front_page** (WINDOW notebk, short tab_no, short page_no);
void	**xvt_notebk_set_tab_image** (WINDOW notebk, short tab_no, XVT_IMAGE image);
void	**xvt_notebk_set_tab_title** (WINDOW notebk, short tab_no, char * title);

EVENT_HANDLER Function Prototype

typedef long (*EVENT_HANDLER) (WINDOW win, EVENT * ep);

use prototype:

long XVT_CALLCONV1 event_handler (WINDOW win, EVENT * ep);

XVT_NOTEBK_ENUM_PAGES Function Prototype

typedef BOOLEAN (* XVT_NOTEBK_ENUM_PAGES) (WINDOW notebk, short tab_no, short page_no, long data);

use prototype:

long XVT_CALLCONV1 enum_func (WINDOW notebk, short tab_no, short page_no, long data);

WIN_DEF Type (Partial)

```
typedef struct s_win_def {
    WIN_TYPE wtype;
    RCT rct;
    char * text;
    UNIT_TYPE units;
    XVT_COLOR_COMPONENT * ctlcolors;
    union {
        struct {
            short int menu_rid;
            MENU_ITEM * menu_p;
            long flags;
            XVT_FNTID ctl_font_id;
            } win;
        ...
    } v;
} WIN_DEF;
```

WINDOW Type

```
typedef long WINDOW;                    /* window descriptor */
```

*XVT_IMAGE Type

```
typedef struct {...} * XVT_IMAGE;
```

Palettes

See Also: Colors, Drawing

```
short               xvt_palet_add_colors   (XVT_PALETTE palet,
                        COLOR * colorsp, short numcolors);
short               xvt_palet_add_colors_from_image   (XVT_PALETTE palet,
                        XVT_IMAGE image);
XVT_PALETTE         xvt_palet_create   (XVT_PALETTE_TYPE type,
                        XVT_PALETTE_ATTR reserved);
XVT_PALETTE         xvt_palet_default   (void);
void                xvt_palet_destroy   (XVT_PALETTE palet);
short               xvt_palet_get_colors   (XVT_PALETTE palet, COLOR * colorsp,
                        short maxcolors);
short               xvt_palet_get_ncolors   (XVT_PALETTE palet);
short               xvt_palet_get_size   (XVT_PALETTE palet);
short               xvt_palet_get_tolerance   (XVT_PALETTE palet);
XVT_PALETTE_TYPE    xvt_palet_get_type   (XVT_PALETTE palet);
void                xvt_palet_set_tolerance   (XVT_PALETTE palet, short tolerance);
```

NULL Palette Constant

```
#define NULL_PALETTE ...
```

Palette Constants

```
#define XVT_CLUT_SIZE ...               /* maximum colors in an image color lookup table */
#define XVT_PALETTE_SIZE ...            /* maximum colors in a palette object */
```

XVT_DISPLAY_TYPE Enumeration Type (Value of ATTR_DISPLAY_TYPE)

```
typedef enum e_display_type {
    XVT_DISPLAY_MONO,           /* monochrome display */
    XVT_DISPLAY_GRAY_16,        /* 16-entry grayscale */
    XVT_DISPLAY_GRAY_256,       /* 256-entry grayscale */
    XVT_DISPLAY_COLOR_16,       /* 16-entry color */
    XVT_DISPLAY_COLOR_256,      /* 256-entry color */
    XVT_DISPLAY_DIRECT_COLOR    /* full color capabilities */
} XVT_DISPLAY_TYPE;
```

*XVT_PALETTE Type

```
typedef struct {...} * XVT_PALETTE;
```

*XVT_PALETTE_ATTR Type

```
typedef struct {...} * XVT_PALETTE_ATTR;    /* reserved for future use */
```

XVT_PALETTE_TYPE Enumeration Type

```
typedef enum {
    XVT_PALETTE_NONE,
    XVT_PALETTE_STOCK,
    XVT_PALETTE_CURRENT,
    XVT_PALETTE_CUBE16,
    XVT_PALETTE_CUBE256,
    XVT_PALETTE_USER
} XVT_PALETTE_TYPE;
```

Picture Objects

See Also: Colors, Drawing, Image Objects, Pixmap Objects

```
PICTURE     xvt_pict_create   (DATA_PTR data, long size, RCT * rctp);
void        xvt_pict_destroy  (PICTURE pict);
DATA_PTR    xvt_pict_lock     (PICTURE pict, long * sizep);
void        xvt_pict_unlock   (PICTURE pict);
```

RCT Type

```
typedef struct s_rct {          /* mathematical rectangle */
    short top;                  /* top coordinate */
    short left;                 /* left coordinate */
    short bottom;               /* bottom coordinate */
    short right;                /* right coordinate */
} RCT;
```

PICTURE Type

```
typedef long PICTURE;           /* encapsulated picture descriptor */
```

NULL Picture Constant

```
#define NULL_PICTURE ...
```

Pixmap Objects

See Also: Colors, Drawing, Image Objects, Picture Objects

```
XVT_PIXMAP      xvt_pmap_create   (WINDOW parent, XVT_PIXMAP_FORMAT format,
                                   short width, short height, XVT_PIXMAP_ATTR reserved);
void            xvt_pmap_destroy  (WINDOW win);
```

NULL Pixmap Constant

```
#define NULL_PIXMAP ...
```

WIN_TYPE Enumeration Type (Partial)

```
typedef enum e_win_type {                   /* type of window */
    ...
    W_PIXMAP,                               /* pixmap */
    ...
} WIN_TYPE;
```

XVT_PIXMAP Type

```
typedef WINDOW XVT_PIXMAP;
```

*XVT_PIXMAP_ATTR Type

```
typedef struct {...} * XVT_PIXMAP_ATTR;     /* reserved for future use */
```

XVT_PIXMAP_FORMAT Enumeration Type

```
typedef enum {
    XVT_PIXMAP_NONE,
    XVT_PIXMAP_DEFAULT
} XVT_PIXMAP_FORMAT;
```

Printing

See Also: Dialogs

```
void            xvt_print_close          (void);
BOOLEAN         xvt_print_close_page     (PRINT_RCD * precp);
PRINT_RCD *     xvt_print_create         (int * sizep);
WINDOW          xvt_print_create_win     (PRINT_RCD * precp, char * title);
void            xvt_print_destroy        (PRINT_RCD * precp);
RCT *           xvt_print_get_next_band  (void);
BOOLEAN         xvt_print_is_valid       (PRINT_RCD * precp);
BOOLEAN         xvt_print_open           (void);
BOOLEAN         xvt_print_open_page      (PRINT_RCD * precp);
BOOLEAN         xvt_print_set_page_orient (PRINT_RCD * precp, XVT_PG_ORIENT pgorient);
BOOLEAN         xvt_print_set_page_size  (PRINT_RCD * precp, XVT_PG_SIZE pgsize);
BOOLEAN         xvt_print_start_thread   (XVT_PRINT_FUNCTION func, long data);
```

PRINT_RCD Type

```
typedef struct s_printrcd {...} PRINT_RCD;
```

XVT_PG_ORIENT Enumeration Type

```
typedef enum {
    PG_PORTRIAT,
    PG_LANGSCAPE
} XVT_PG_ORIENT;;
```

XVT_PG_SIZE Type

```
typedef struct {
    double width,
    double height,
    XVT_PG_UNITS pgunits
} XVT_PG_SIZE;
```

XVT_PG_UNITS Enumeration Type

```
typedef enum {
    PG_IN,
    PG_MM,
} XVT_PG_UNITS;
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

*XVT_PRINT_FUNCTION Function Prototype

```
typedef BOOLEAN (* XVT_PRINT_FUNCTION) (long data);
```

use prototype:

```
BOOLEAN XVT_CALLCONV1 xvt_print_function (long data);
```

Rectangle Objects

See Also: Drawing

```
short       xvt_rect_get_height   (RCT * rctp);
PNT *       xvt_rect_get_pos      (RCT * rctp, PNT * pos);
short       xvt_rect_get_width    ( RCT * rctp);
BOOLEAN     xvt_rect_has_point    (RCT * rctp, PNT pnt);
BOOLEAN     xvt_rect_intersect    (RCT * rctp, RCT * rctp1, RCT * rctp2);
BOOLEAN     xvt_rect_is_empty     (RCT * rctp);
void        xvt_rect_offset       (RCT * rctp, short dh, short dv);
void        xvt_rect_set          (RCT * rctp, short left, short top, short right, short bottom);
void        xvt_rect_set_empty    (RCT * rctp);
BOOLEAN     xvt_rect_set_height   (RCT * rctp, short height);
BOOLEAN     xvt_rect_set_pos      (RCT * rctp, PNT pos);
BOOLEAN     xvt_rect_set_width    (RCT * rctp, short width);
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

PNT Type

```
typedef struct s_pnt {                  /* mathematical point */
    short v;                            /* vertical (y) coordinate */
    short h;                            /* horizontal (x) coordinate */
} PNT;
```

Resource Manager

See Also: Controls, Dialogs, Windows

XVT_RES	**xvt_res_add_res** (char * res_name);
void	**xvt_res_free_menu_tree** (MENU_ITEM * mip);
void	**xvt_res_free_win_def** (WIN_DEF * windef);
char *	**xvt_res_get_dlg_data** (int rid, int cid, int tag);
WIN_DEF *	**xvt_res_get_dlg_def** (int rid);
XVT_FNTID	**xvt_res_get_font** (int rid);
XVT_IMAGE	**xvt_res_get_image** (int rid);
char *	**xvt_res_get_image_data** (int rid, int tag);
MENU_ITEM *	**xvt_res_get_menu** (int rid);
char *	**xvt_res_get_menu_data** (int rid, int cid, int tag);
char *	**xvt_res_get_str** (int rid, char * s, int sz_s);
SLIST	**xvt_res_get_str_list** (int rid_first, int rid_last);
char *	**xvt_res_get_win_data** (int rid, int cid, int tag);
WIN_DEF *	**xvt_res_get_win_def** (int rid);
BOOLEAN	**xvt_res_remove_res** (XVT_RES res);
XVT_RES	**xvt_res_use_res** (XVT_RES res);

Resource ID Constants

```
#define DB_ABORT ...
#define DB_ABOUT ...
#define DB_ASK ...
#define DB_ERROR ...
#define DB_FONTSEL ...
#define DB_HELPTEXT ...
#define DB_HELPTOPICS ...
#define DB_NOTE ...
#define DB_OPEN ...
#define DB_RESPONSE ...
#define DB_SAVE ...
#define DB_WARNING ...
#define ICON_RSRC ...
#define MENU_BAR_RID ...                /* ID for default menubar resource */
#define STR_HELPTYPE ...                /* string resource for help file type */
#define XVT_STRING_RES_BASE ...
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

WIN_DEF Type (Full)

```
typedef struct s_win_def {
    WIN_TYPE wtype;
    RCT rct;
    char * text;
    UNIT_TYPE units;
    XVT_COLOR_COMPONENT * ctlcolors;
    union {
        struct {
            short int menu_rid;
            MENU_ITEM * menu_p;
            long flags;
            XVT_FNTID ctl_font_id;
            } win;

        struct {
            long flags;
            XVT_FNTID ctl_font_id;
        } dlg;
        struct {
            short int ctrl_id;
            short int icon_id;
            long flags;
            XVT_FNTID font_id;
        } ctl;
        struct {
            unsigned short attrib;
            XVT_FNTID font_id;
            short int margin;
            short int limit;
            short int tx_id;
        } tx;
    } v;
} WIN_DEF;
```

WIN_TYPE Enumeration Type (Full)

```
typedef enum e_win_type {          /* type of window */
W_NONE,                            /* marker for end of WIN_DEF array */
W_DOC,                             /* document window */
W_PLAIN,                           /* window with plain border */
W_DBL,                             /* window with double border */
W_PRINT,                           /* XVT internal use only */
W_TASK,                            /* task window */
W_SCREEN,                          /* screen window */
W_NO_BORDER,                       /* no border */
W_PIXMAP,                          /* pixmap */
W_MODAL,                           /* modal window */
WD_MODAL,                          /* modal dialog */
WD_MODELESS,                       /* modeless dialog */
WC_PUSHBUTTON,                     /* button control */
WC_RADIOBUTTON,                    /* radio button control */
WC_CHECKBOX,                       /* check box control */
WC_HSCROLL,                        /* horizontal scroll bar control */
WC_VSCROLL,                        /* vertical scroll bar control */
WC_EDIT,                           /* edit control */
WC_TEXT,                           /* static text control */
WC_LBOX,                           /* list box control */
WC_LISTBUTTON,                     /* button with list */
WC_LISTEDIT,                       /* edit field with list */
WC_GROUPBOX,                       /* group box */
WC_TEXTEDIT,                       /* text-edit object */
WC_ICON,                           /* icon control */
WC_NOTEBK,                         /* notebook control */
WC_HTML,                           /* html control */
WC_NUM_WIN_TYPES,                  /* number of WIN_TYPEs */
XVT_ENUM_DUMMY12 = XVT_CC_ENUM_END
} WIN_TYPE;
```

XVT_RES Type

```
typedef long XVT_RES;              /* Resource ID */
```

Screen Window Objects

See Also: Windows, Visible Objects

```
void        xvt_scr_beep (void);
WINDOW      xvt_scr_get_focus_topwin (void);
WINDOW      xvt_scr_get_focus_vobj (void);
void        xvt_scr_hide_cursor (void);
SLIST       xvt_scr_list_wins (void);
void        xvt_scr_set_busy_cursor (void);
void        xvt_scr_set_focus_vobj (WINDOW win);
BOOLEAN     xvt_scr_launch_browser (const char *xrc);
```

Predefined Window Constant (value is WINDOW)

```
#define SCREEN_WIN ...
```

Scrollbar Controls

See Also: Controls, Visible Objects

```
int         xvt_sbar_get_pos (WINDOW ctl_win, SCROLL_TYPE t);
int         xvt_sbar_get_proportion (WINDOW ctl_win, SCROLL_TYPE t);
```

```
void        xvt_sbar_get_range   (WINDOW ctl_win, SCROLL_TYPE t, int * mnp, int * mxp);
void        xvt_sbar_set_pos     (WINDOW ctl_win, SCROLL_TYPE t, int pos);
void        xvt_sbar_set_proportion  (WINDOW ctl_win, SCROLL_TYPE t, int proportion);
void        xvt_sbar_set_range   (WINDOW ctl_win, SCROLL_TYPE t, int mn, int mx);
```

SCROLL_CONTROL Enumeration Type

```
typedef enum e_scroll_ctl {             /* site of scrollbar activity */
    SC_NONE,                            /* nowhere (event should be ignored) */
    SC_LINE_UP,                         /* one line up */
    SC_LINE_DOWN,                       /* one line down */
    SC_PAGE_UP,                         /* previous page */
    SC_PAGE_DOWN,                       /* next page */
    SC_THUMB,                           /* thumb repositioning */
    SC_THUMBTRACK                       /* dynamic thumb tracking */
} SCROLL_CONTROL;
```

SCROLL_TYPE Type

```
typedef enum e_scroll_type {            /* type of scrollbar */
    HSCROLL,                            /* horizontal */
    VSCROLL,                            /* vertical */
    HVSCROLL                            /* either (used for dialog controls) */
} SCROLL_TYPE;
```

SLIST (String List) Objects

```
BOOLEAN     xvt_slist_add_at_elt   (SLIST x, SLIST_ELT e, char * sx, long data);
BOOLEAN     xvt_slist_add_at_pos   (SLIST x, int index, char * sx, long data);
BOOLEAN     xvt_slist_add_sorted   (SLIST x, char * s, long data, BOOLEAN unique,
                BOOLEAN case_sensitive);
int         xvt_slist_count   (SLIST x);
SLIST       xvt_slist_create   (void);
void        xvt_slist_debug   (SLIST x);
void        xvt_slist_destroy   (SLIST x);
char *      xvt_slist_get   (SLIST x, SLIST_ELT e, long * datap);
long *      xvt_slist_get_data   (SLIST_ELT e);
char *      xvt_slist_get_elt   (SLIST x, int index, long * datap);
SLIST_ELT   xvt_slist_get_first   (SLIST x);
SLIST_ELT   xvt_slist_get_next   (SLIST x, SLIST_ELT e);
BOOLEAN     xvt_slist_is_valid   (SLIST x);
BOOLEAN     xvt_slist_rem   (SLIST x, SLIST_ELT e);
```

SLIST Type

```
typedef struct s_slist {...} * SLIST;
```

SLIST_ELT Type

```
typedef struct s_slistelt {...} * SLIST_ELT;        /* element of string list */
```

Standard XVT Dialogs

See: Dialogs

Strings

```
int             xvt_str_collate (const char * mbs1, const char * mbs2);
int             xvt_str_collate_ignoring_case (const char * mbs1, const char * mbs2);
int             xvt_str_compare (const char * mbs1, const char * mbs2);
int             xvt_str_compare_ignoring_case (const char * mbs1, const char * mbs2);
int             xvt_str_compare_n_char (const char * mbs1, const char * mbs2,
                    const size_t n);
int             xvt_str_compare_n_size_ignoring_case( const char * mbs1,
                                const char * mbs2, const size_t n );
int             xvt_str_compare_wchar( const char * mbs1, XVT_WCHAR wc );
char *          xvt_str_concat (char * mbs1, const char * mbs2);
char *          xvt_str_concat_n_char (char * mbs1, const char * mbs2, const size_t n);
int             xvt_str_convert_mb_to_wc (XVT_WCHAR * wc, const char * mbs);
int             xvt_str_convert_mbs_to_wcs (XVT_WCHAR * wcs, const char * mbs,
                    const size_t n);
size_t          xvt_str_convert_to_lower (char * mbs1, const char * mbs2, const size_t n);
size_t          xvt_str_convert_to_upper (char * mbs1, const char * mbs2, const size_t n);
int             xvt_str_convert_wc_to_mb (char * mbs, const XVT_WCHAR wc);
XVT_WCHAR       xvt_str_convert_wchar_to_lower (XVT_WCHAR wc);
XVT_WCHAR       xvt_str_convert_wchar_to_upper (XVT_WCHAR wc);
int             xvt_str_convert_wcs_to_mbs (char * mbs, const XVT_WCHAR * wcs,
                    const size_t n);
char *          xvt_str_copy (char * mbs1, const char * mbs2);
char *          xvt_str_copy_n_char (char * mbs1, const char * mbs2, const size_t n);
size_t          xvt_str_copy_n_size (char * mbs1, const char * mbs2, size_t n);
char *          xvt_str_duplicate (const char * mbs);
char *          xvt_str_find_char_set (const char * mbs, const char * mbset);
char *          xvt_str_find_eol (char * buf, long nbytes, long * lenp, EOL_FORMAT * fp);
char *          xvt_str_find_first_char (const char * mbs, const char * mbc);
char *          xvt_str_find_first_wchar ( const char * mbs, XVT_WCHAR wc );
char *          xvt_str_find_last_char (const char * mbs, const char * mbc);
char *          xvt_str_find_last_wchar ( const char * mbs, XVT_WCHAR wc );
char *          xvt_str_find_not_char_set (const char * mbs, const char * mbset);
char *          xvt_str_find_substring (const char * mbs1, const char * mbs2);
char *          xvt_str_find_token (const char * mbs, const char * delimiter, size_t * n);
char *          xvt_str_get_mnemonic ( const char * label, short * offset,
                                char * buf, const size_t bug_size );
char *          xvt_str_set_mnemonic ( const char * label, const short offset,
                                char * buf, const size_t buf_size );
size_t          xvt_str_get_byte_count (const char * mbs);
size_t          xvt_str_get_char_count (const char * mbs);
int             xvt_str_get_char_size (const char * mbs);
const char *    xvt_str_get_next_char (const char * mbs);
size_t          xvt_str_get_n_char_count (const char * mbs, const size_t n, size_t * used);
size_t          xvt_str_get_n_char_size (const char * mbs, const size_t n);
const char *    xvt_str_get_prev_char (const char * start, const char * mbs);
BOOLEAN         xvt_str_is_alnum (const char * mbs);
BOOLEAN         xvt_str_is_alpha (const char * mbs);
BOOLEAN         xvt_str_is_cntrl ( const char * mbs );
BOOLEAN         xvt_str_is_digit (const char * mbs);
BOOLEAN         xvt_str_is_equal (const char * mbs1, const char * mbs2);
BOOLEAN         xvt_str_is_equal_n_size ( const char * mbs1, const char * mbs2 );
BOOLEAN         xvt_str_is_graph ( const char * mbs );
BOOLEAN         xvt_str_is_invariant (const char * mbs);
```

```
BOOLEAN         xvt_str_is_lower   (const char * mbs);
BOOLEAN         xvt_str_is_print   ( const char * mbs );
BOOLEAN         xvt_str_is_punct   ( const char * mbs );
BOOLEAN         xvt_str_is_space   (const char * mbs);
BOOLEAN         xvt_str_is_upper   (const char * mbs);
BOOLEAN         xvt_str_is_xdigit  (const char * mbs);
BOOLEAN         xvt_str_match      (char * src, char * pat, BOOLEAN case_sensitive);
double          xvt_str_parse_double  (const char * mbs, char ** mbsend);
long            xvt_str_parse_long    (const char * mbs, char ** mbsend, short base);
unsigned long   xvt_str_parse_ulong   (const char * mbs, char ** mbsend, short base);
size_t          xvt_str_sprintf    (char * mbs, const char * format, ...);
size_t          xvt_str_vsprintf   (char * mbs, const char * format, va_list varg);
```

EOL_FORMAT Enumeration Type

```
typedef struct s_pat { ... } *XVT_PATTERN;

typedef const char *(*XVT_FORMAT_HANDLER)( WINDOW win,
        char * instr, int * start, int * end, void * data );

XVT_PATTERN xvt_pattern_create( const char * patstr );
void xvt_pattern_destroy( XVT_PATTERN pat );

BOOLEAN xvt_pattern_match( XVT_PATTERN pat, const char * str,
        const char** endstr );

char * xvt_pattern_format_string( XVT_PATTERN pat, const char * str,
        char * buf, size_t buflen, BOOLEAN complete_string, int * start,
        int * end );

XVT_FORMAT_HANDLER xvt_vobj_get_formatter( WINDOW win );
void * xvt_vobj_get_formatter_data( WINDOW win );

void xvt_vobj_set_formatter( WINDOW win,
        XVT_FORMAT_HANDLER handler, void * data );

void xvt_vobj_set_formatter_data( WINDOW win, void * data );
```

EOL_FORMAT Enumeration Type

```
typedef enum e_eol {                    /* terminator found by xvt_fsys_find_eol function */
    EOL_NORMAL,                         /* normal (or first) line terminator */
    EOL_DIFF,                           /* terminator different from previous */
    EOL_NONE                            /* end of buffer before any terminator */
} EOL_FORMAT;
```

XVT_MAX_MB_SIZE Constant

```
#define XVT_MAX_MB_SIZE ...             /* maximum size of a multibyte character */
```

XVT_WCHAR Type

```
typedef ... XVT_WCHAR;                  /* platform-specific */
```

Text Edit Objects

See Also: Controls, Visible Objects

```
BOOLEAN     xvt_tx_add_par   (TXEDIT tx, T_PNUM pnum, char * s);
BOOLEAN     xvt_tx_append    (TXEDIT tx, T_PNUM pnum, char * s);
BOOLEAN     xvt_tx_clear     (TXEDIT tx);
TXEDIT      xvt_tx_create    (WINDOW win, RCT * lrctp, unsigned attr, XVT_FNTID font_id,
                int margin, int limit);
TXEDIT      xvt_tx_create_def  (WIN_DEF * win_def_p,
                WINDOW parent_win, long app_data);
BOOLEAN     xvt_tx_destroy   (TXEDIT tx);
unsigned    xvt_tx_get_attr  (TXEDIT tx);
int         xvt_tx_get_limit   (TXEDIT tx);
char *      xvt_tx_get_line  (TXEDIT tx, T_PNUM pnum, ACCESS_CMD cmd,
                T_LNUM lnum, unsigned * lenp);
int         xvt_tx_get_margin  (TXEDIT tx);
T_CNUM      xvt_tx_get_num_chars  (TXEDIT tx, T_PNUM pnum, T_LNUM lnum);
T_LNUM      xvt_tx_get_num_lines  (TXEDIT tx);
T_LNUM      xvt_tx_get_num_par_lines  (TXEDIT tx, T_PNUM pnum);
T_PNUM      xvt_tx_get_num_pars  (TXEDIT tx);
void        xvt_tx_get_origin  (TXEDIT tx, T_PNUM * pnump, T_LNUM * lnump,
                T_LNUM * org_linep, T_CPOS * org_offsetp);
void        xvt_tx_get_sel   (TXEDIT tx, T_PNUM * p1, T_LNUM * l1, T_CNUM * c1,
                T_PNUM * p2, T_LNUM * l2, T_CNUM * c2);
T_CNUM      xvt_tx_get_tabstop   (TXEDIT tx);
RCT *       xvt_tx_get_view   (TXEDIT tx, RCT * rct);
BOOLEAN     xvt_tx_rem_par   (TXEDIT tx, T_PNUM pnum);
void        xvt_tx_reset   (TXEDIT tx);
void        xvt_tx_resume   (TXEDIT tx);
void        xvt_tx_scroll_hor   (TXEDIT tx, int pixel_amt);
void        xvt_tx_scroll_vert   (TXEDIT tx, int line_amt);
void        xvt_tx_set_attr   (TXEDIT tx, unsigned attr);
void        xvt_tx_set_limit   (TXEDIT tx, int limit);
void        xvt_tx_set_margin   (TXEDIT tx, int margin);
BOOLEAN     xvt_tx_set_par   (TXEDIT tx, T_PNUM pnum, char * s);
void        xvt_tx_set_scroll_callback   (TXEDIT tx, SCROLL_CALLBACK scroll_callback);
void        xvt_tx_set_sel   (TXEDIT tx, T_PNUM p1, T_LNUM l1, T_CNUM c1,
                T_PNUM p2, T_LNUM l2, T_CNUM c2);
void        xvt_tx_set_tabstop   (TXEDIT tx, T_CNUM tabstop);
void        xvt_tx_suspend   (TXEDIT tx);
```

ACCESS_CMD Enumeration Type

```
typedef enum e_access {
    A_LOCK,
    A_GET,
    A_UNLOCK
} ACCESS_CMD;
```

NULL Text Edit Object

```
#define NULL_TXEDIT ...
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

SCROLL_CALLBACK Function Prototype

```
typedef void (* SCROLL_CALLBACK) (TXEDIT tx, T_LNUM org_line, T_LNUM nlines,
    T_CPOS org_offset);
```

use prototype:

```
void XVT_CALLCONV1 scroll_callback (TXEDIT tx, T_LNUM org_line, T_LNUM nlines,
    T_CPOS org_offset);
```

Text Edit Attributes

```
#define ATTR_R40_TXEDIT_BEHAVIOR ...   /* revert to R4.0 text edit behavior */

#define TX_AUTOHSCROLL ...             /* autoscroll horizontally */
#define TX_AUTOVSCROLL ...             /* autoscroll vertically */
#define TX_BORDER ...                  /* rectangular border */
#define TX_DISABLED ...                /* disabled state */
#define TX_ENABLECLEAR ...             /* leave CLEAR enabled always */
#define TX_INVISIBLE ...               /* visibility state */
#define TX_ONEPAR ...                  /* one paragraph only (no \r) */
#define TX_OVERTYPE ...                /* overtype mode */
#define TX_NOCOPY ...                  /* no copy allowed */
#define TX_NOCUT ...                   /* no cut allowed */
#define TX_NOMENU ...                  /* no edit menu changes */
#define TX_NOPASTE ...                 /* no paste allowed */
#define TX_READONLY ..                 /* text is not editable */
#define TX_WRAP ...                    /* wrap text to margin */
```

T_CNUM Type

```
typedef unsigned short T_CNUM;
```

T_CPOS Type

```
typedef unsigned short T_CPOS;
```

T_LNUM Type

```
typedef unsigned short T_LNUM;
```

T_PNUM Type

```
typedef unsigned short T_PNUM;
```

TXEDIT Type

```
#define TXEDIT WINDOW                  /* defines text edit as a window */
```

Timer Objects

See Also: Events

```
long        xvt_timer_create   (WINDOW win, long interval);
void        xvt_timer_destroy  (long id);
```

Time Error Return Value Constant

```
#define XVT_TIMER_ERROR ...
```

Treeview Controls

See Also: Controls, Visible Objects

```
BOOLEAN                 xvt_treeview_add_child_node(XVT_TREEVIEW_NODE parent_node,
                                        XVT_TREEVIEW_NODE child_node);
void                    xvt_treeview_collapse_node(XVT_TREEVIEW_NODE node,
                                        BOOLEAN recurse);
WINDOW                  xvt_treeview_create(WINDOW parent_win, RCT *  rct_p, char * title,
                                        longctl_flags, long data, int ctl_id,
                                        XVT_IMAGE item_image, XVT_IMAGE collapsed_image,
                                        XVT_IMAGE expanded_image, long attrs, int line_height);
XVT_TREEVIEW_NODE xvt_treeview_create_node(XVT_TREEVIEW_NODE_TYPE type,
                                        XVT_IMAGE  item_image, XVT_IMAGE  collapsed_image,
                                        XVT_IMAGE  expanded_image, char *string,
                                        XVT_TREEVIEW_CALLBACK  callback, long data);
void                    xvt_treeview_destroy_node(XVT_TREEVIEW_NODE node);
void                    xvt_treeview_expand_node(XVT_TREEVIEW_NODE node,
                                        BOOLEAN  recurse);
long                    xvt_treeview_get_attributes(WINDOW ctl_win);
XVT_TREEVIEW_NODE xvt_treeview_get_child_node(XVT_TREEVIEW_NODE parent_node,
                                        int position);
long                    xvt_treeview_get_line_height(WINDOW ctl_win);
XVT_TREEVIEW_CALLBACK  xvt_treeview_get_node_callback( XVT_TREEVIEW_NODE node);
long                    xvt_treeview_get_node_data(XVT_TREEVIEW_NODE node);
XVT_IMAGE               xvt_treeview_get_node_image_collapsed(XVT_TREEVIEW_NODE node);
XVT_IMAGE               xvt_treeview_get_node_image_expanded(XVT_TREEVIEW_NODE node);
XVT_IMAGE               xvt_treeview_get_node_image_item(XVT_TREEVIEW_NODE node);
int                     xvt_treeview_get_node_num_children(XVT_TREEVIEW_NODE node);
int                     xvt_treeview_get_node_num_vis_children(XVT_TREEVIEW_NODE node);
char *                  xvt_treeview_get_node_string(XVT_TREEVIEW_NODE node, char *string,
                                        int sz_string);
XVT_TREEVIEW_NODE_TYPE xvt_treeview_get_node_type(XVT_TREEVIEW_NODE node);
XVT_TREEVIEW_NODE xvt_treeview_get_parent_node(XVT_TREEVIEW_NODE child_node);
XVT_TREEVIEW_NODE xvt_treeview_get_root_node(WINDOW ctl_win);
BOOLEAN                 xvt_treeview_node_selected(XVT_TREEVIEW_NODE node);
BOOLEAN                 xvt_treeview_remove_child_node(XVT_TREEVIEW_NODE child_node);
void                    xvt_treeview_resume(WINDOW ctl_win);
void                    xvt_treeview_set_attributes(WINDOW ctl_win, long attrs);
void                    xvt_treeview_set_line_height(WINDOW ctl_win, long line_height);
void                    xvt_treeview_set_node_callback(XVT_TREEVIEW_NODE  node,
                                        XVT_TREEVIEW_CALLBACK fcn);
void                    xvt_treeview_set_node_data (XVT_TREEVIEW_NODE node, long  data);
void                    xvt_treeview_set_node_image_collapsed(XVT_TREEVIEW_NODE node,
                                        XVT_IMAGE image);void
                           xvt_treeview_set_node_string(XVT_TREEVIEW_NODE node, char *string);

void                    xvt_treeview_set_node_image_expanded(XVT_TREEVIEW_NODE node,
                                        XVT_IMAGE image);
void                    xvt_treeview_set_node_image_item(XVT_TREEVIEW_NODE node,
                                        XVT_IMAGE image);
void                    xvt_treeview_set_node_string(XVT_TREEVIEW_NODE node, char *string);
void                    xvt_treeview_set_node_type(XVT_TREEVIEW_NODE node,
                                        XVT_TREEVIEW_NODE_TYPE type);
void                    xvt_treeview_suspend(WINDOW ctl_win);
```

void **xvt_treeview_update**(WINDOW ctl_win);

Visible Objects

See Also: Colors, Dialogs, Screen Window Objects, Visible Objects, Windows

void **xvt_vobj_destroy** (WINDOW win);
long **xvt_vobj_get_attr** (WINDOW win, int attr);
RCT * **xvt_vobj_get_client_rect** (WINDOW win, RCT * lrctp);
long **xvt_vobj_get_data** (WINDOW win);
long **xvt_vobj_get_flags** (WINDOW win);
RCT * **xvt_vobj_get_outer_rect** (WINDOW win, RCT * rctp);
XVT_PALETTE **xvt_vobj_get_palet** (WINDOW win);
WINDOW **xvt_vobj_get_parent** (WINDOW win);
char * **xvt_vobj_get_title** (WINDOW win, char * title, int sz_title);
WIN_TYPE **xvt_vobj_get_type** (WINDOW win);
BOOLEAN **xvt_vobj_is_focusable** (WINDOW win);
BOOLEAN **xvt_vobj_is_valid** (WINDOW win);
void **xvt_vobj_move** (WINDOW win, RCT * grctp);
void **xvt_vobj_raise** (WINDOW win);
void **xvt_vobj_set_attr** (WINDOW win, int attr, long value);
void **xvt_vobj_set_data** (WINDOW win, long data);
void **xvt_vobj_set_enabled** (WINDOW win, BOOLEAN enable);
void **xvt_vobj_set_palet** (WINDOW win, XVT_PALETTE palette);
void **xvt_vobj_set_title** (WINDOW win, char * title);
void **xvt_vobj_set_visible** (WINDOW win, BOOLEAN show);
void **xvt_vobj_translate_points** (WINDOW from, WINDOW to, PNT * pntp, int npnts);

PNT Type

```
typedef struct s_pnt {                  /* mathematical point */
    short v;                            /* vertical (y) coordinate */
    short h;                            /* horizontal (x) coordinate */
} PNT;
```

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

Application Configuration Attributes
#define ATTR_APP_CTL_COLORS ...
#define ATTR_APP_CTL_FONT_RID ...
#define ATTR_APPL_NAME_RID ...
#define ATTR_BACK_COLOR ...
#define ATTR_COLLATE_HOOK ...
#define ATTR_DEBUG_FILENAME ...
#define ATTR_DEFAULT_PALETTE_TYPE ...
#define ATTR_DISPLAY_TYPE ...
#define ATTR_ERRMSG_FILENAME ...
#define ATTR_HAVE_COLOR ...
#define ATTR_HAVE_MOUSE ...
#define ATTR_MULTIBYTE_AWARE ...
#define ATTR_NUM_TIMERS ...
#define ATTR_PROPAGATE_NAV_CHARS ...
#define ATTR_R40_TXEDIT_BEHAVIOR ...
#define ATTR_RESOURCE_FILENAME ...
#define ATTR_TASKWIN_TITLE_RID ...
#define ATTR_XVT_CONFIG ...

Control Attributes
#define ATTR_APP_CTL_COLORS ...
#define ATTR_APP_CTL_FONT_RID ...
#define ATTR_CTL_BUTTON_HEIGHT ...
#define ATTR_CTL_CHECKBOX_HEIGHT ...
#define ATTR_CTL_EDIT_TEXT_HEIGHT ...
#define ATTR_CTL_HORZ_SBAR_HEIGHT ...
#define ATTR_CTL_RADIOBUTTON_HEIGHT ...
#define ATTR_CTL_STATIC_TEXT_HEIGHT ...
#define ATTR_CTL_VERT_SBAR_WIDTH ...
#define ATTR_CTL_PASSWORD ...

#define ATTR_R40_TXEDIT_BEHAVIOR ...

Debugging Attributes
#define ATTR_DEBUG_FILENAME ...

Error Handling Attributes
#define ATTR_ERRMSG_FILENAME ...
#define ATTR_ERRMSG_HANDLER ...

Event Attributes
#define ATTR_EVENT_HOOK ...
#define ATTR_HELP_HOOK ...
#define ATTR_KEY_HOOK ...
#define ATTR_PROPAGATE_NAV_CHARS ...
#define ATTR_SUPPRESS_UPDATE_CHECK ...

Font Attributes
#define ATTR_FONT_CACHE_SIZE ...
#define ATTR_FONT_DIALOG ...
#define ATTR_FONT_MAPPER ...

Help System Attributes
#define ATTR_HELP_CONTEXT ....
#define ATTR_HELP_HOOK ...

Icon Size Attributes
#define ATTR_ICON_HEIGHT ...
#define ATTR_ICON_WIDTH ...

Localization Attributes
#define ATTR_APPL_NAME_RID ...
#define ATTR_COLLATE_HOOK ...
#define ATTR_ERRMSG_FILENAME ...
#define ATTR_KEY_HOOK ...
#define ATTR_MULTIBYTE_AWARE ...
#define ATTR_PROPAGATE_NAV_CHARS ...
#define ATTR_RESOURCE_FILENAME ...
#define ATTR_TASKWIN_TITLE_RID ...

Memory Attributes
#define ATTR_MEMORY_MANAGER ...

Native Window Attributes
#define ATTR_NATIVE_GRAPHIC_CONTEXT ...
#define ATTR_NATIVE_WINDOW ...

Printer Attributes
#define ATTR_PRINTER_HEIGHT ...
#define ATTR_PRINTER_HRES ...
#define ATTR_PRINTER_VRES ...
#define ATTR_PRINTER_WIDTH ...

Screen Window Attributes
#define ATTR_SCREEN_HEIGHT ...
#define ATTR_SCREEN_HRES ...
#define ATTR_SCREEN_VRES ...
#define ATTR_SCREEN_WIDTH ...
#define ATTR_SCREEN_WINDOW ...

Task Window Attributes
#define ATTR_TASK_WINDOW ...
#define ATTR_TASKWIN_TITLE_RID ...

Window Attributes
#define ATTR_DBLFRAME_HEIGHT ...
#define ATTR_DBLFRAME_WIDTH ...
#define ATTR_DOCFRAME_WIDTH ...
#define ATTR_DOCFRAME_HEIGHT ...
#define ATTR_DOC_STAGGER_HORZ ...
#define ATTR_DOC_STAGGER_VERT ...
#define ATTR_FRAME_HEIGHT ...
#define ATTR_FRAME_WIDTH ...
#define ATTR_MENU_HEIGHT ...
#define ATTR_SCREEN_WINDOW ...
#define ATTR_TASK_WINDOW ...
#define ATTR_TASKWIN_TITLE_RID ...
#define ATTR_TITLE_HEIGHT ...

Windows

See Also: Controls, Dialogs, Events, Fonts, Screen Window Objects, Visible Objects

WINDOW	**xvt_win_create** (WIN_TYPE wtype, RCT * rct_p, char * title, int menu_rid, WINDOW parent_win, long win_flags, EVENT_MASK mask, EVENT_HANDLER eh, long app_data);
WINDOW	**xvt_win_create_def** (WIN_DEF * win_def_p, WINDOW parent_win, EVENT_MASK mask, EVENT_HANDLER eh, long app_data);
WINDOW	**xvt_win_create_res** (int rid, WINDOW parent_win, EVENT_MASK mask, EVENT_HANDLER eh, long app_data);
long	**xvt_win_dispatch_event** (WINDOW win, EVENT * ep);
BOOLEAN	**xvt_win_enum_wins** (WINDOW parent_win, XVT_ENUM_CHILDREN func, long data, unsigned long reserved);
WINDOW	**xvt_win_get_ctl** (WINDOW win, int cid);
XVT_COLOR_COMPONENT*	**xvt_win_get_ctl_colors** (WINDOW win);
XVT_FNTID	**xvt_win_get_ctl_font** (WINDOW win);
CURSOR	**xvt_win_get_cursor** (WINDOW win);
EVENT_MASK	**xvt_win_get_event_mask** (WINDOW win);
EVENT_HANDLER	**xvt_win_get_handler** (WINDOW win);
XVT_NAV	**xvt_win_get_nav** (WINDOW win);
TXEDIT	**xvt_win_get_tx** (WINDOW win, int cid);
BOOLEAN	**xvt_win_has_menu** (WINDOW win);
SLIST	**xvt_win_list_wins** (WINDOW parent_win, unsigned long reserved);
void	**xvt_win_process_modal** (WINDOW win);
void	**xvt_win_release_pointer** (void);
void	**xvt_win_set_ctl_colors** (WINDOW win, XVT_COLOR_COMPONENT * colors, XVT_COLOR_ACTION action);
void	**xvt_win_set_ctl_font** (WINDOW win, XVT_FNTID font_id);
void	**xvt_win_set_caret_pos** (WINDOW win, PNT pos);
void	**xvt_win_set_caret_size** (WINDOW win, int width, int height);
void	**xvt_win_set_caret_visible** (WINDOW win, BOOLEAN visible);
void	**xvt_win_set_cursor** (WINDOW win, CURSOR c);
void	**xvt_win_set_doc_title** (WINDOW win, char * title);
void	**xvt_win_set_event_mask** (WINDOW win, EVENT_MASK mask);
void	**xvt_win_set_handler** (WINDOW win, EVENT_HANDLER eh);
void	**xvt_win_trap_pointer** (WINDOW win);

Maximum Window Rectangle Constant

#define XVT_MAX_WINDOW_RECT ...

NULL Window Constant

#define NULL_WIN ...

Predefined Window Constants (value is WINDOW)

#define SCREEN_WIN ...
#define TASK_WIN ...

RCT Type

```
typedef struct s_rct {                  /* mathematical rectangle */
    short top;                          /* top coordinate */
    short left;                         /* left coordinate */
    short bottom;                       /* bottom coordinate */
    short right;                        /* right coordinate */
} RCT;
```

UNIT_TYPE Type

```
typedef enum e_unit_type {
    U_PIXELS,
    U_CHARS,
    U_SEMICHARS
} UNIT_TYPE;
```

EVENT_HANDLER Function Prototype

```
typedef long (*EVENT_HANDLER) (WINDOW win, EVENT * ep);
```

use prototype:

```
long XVT_CALLCONV1 event_handler (WINDOW win, EVENT * ep);
```

WIN_DEF Type (Partial)

```
typedef struct s_win_def {
    WIN_TYPE wtype;
    RCT rct;
    char * text;
    UNIT_TYPE units;
    XVT_COLOR_COMPONENT * ctlcolors;
    union {
        struct {
            short int menu_rid;
            MENU_ITEM * menu_p;
            long flags;
            XVT_FNTID ctl_font_id;
            } win;
        ...
    } v;
} WIN_DEF;
```

WINDOW Type

```
typedef long WINDOW;                    /* window descriptor */
```

WIN_TYPE Enumeration Type (Partial)

```
typedef enum e_win_type {               /* type of window */
    ...
    W_DOC,                              /* document window */
    W_PLAIN,                            /* window with plain border */
    W_DBL,                              /* window with double border */
    W_PRINT,                            /* XVT internal use only */
    W_TASK,                             /* task window */
    W_SCREEN,                           /* screen window */
    W_NO_BORDER,                        /* no border */
    W_PIXMAP,                           /* pixmap */
    W_MODAL,                            /* modal window */
    ...
} WIN_TYPE;
```

Window Creation Flag Constants

```
#define WSF_CLOSE ...              /* is user closeable */
#define WSF_DECORATED ...          /* sizeable, closeable, and has horz. and vert. scrollbar */
#define WSF_DEFER_MODAL ...        /* modal status deferred (not processed by
                                      xvt_win_create*) */
#define WSF_DISABLED ...           /* is initially disabled */
#define WSF_FLOATING ...           /* is floating */
#define WSF_HSCROLL ...            /* has horizontal scrollbar outside client area */
#define WSF_ICONIZABLE ...         /* may iconize window */
#define WSF_ICONIZED ...           /* is initially iconized */
#define WSF_INVISIBLE ...          /* is initially invisible */
#define WSF_MAXIMIZED ...          /* initially maximized */
#define WSF_NO_MENUBAR ...         /* has no menubar of its own */
#define WSF_NONE ...               /* no flags set */
#define WSF_PLACE_EXACT ...        /* place modal window at specified coordinates */
#define WSF_SIZE ...               /* is user sizeable */
#define WSF_SIZEONLY ...           /* lacks border rectangles (Mac only) */
#define WSF_VSCROLL ...            /* has vertical scrollbar outside client area */
```

*XVT_ENUM_CHILDREN Function Prototype

```
typedef BOOLEAN (* XVT_ENUM_CHILDREN) (WINDOW child, long data);
```

use prototype:

```
BOOLEAN XVT_CALLCONV1 enumerate_children (WINDOW child, long data);
```

XRC Statements

accel *tag symbol* [*modifiers*]

symbol:	Key	symbol:	Key
f1	F1	**kp0**	0 on keypad
...	...	...	...
f24	F24	**kp9**	9 on keypad
tab	Tab	**ins**	Insert
back	Backspace	**del**	Delete
return	Return	**copy**	Copy
esc	Escape	**cut**	Cut
mult	* on keypad	**paste**	Paste
sub	- on keypad	**"A"**	ASCII characters
add	+ on keypad		

modifiers:
- alt
- control
- shift

button *id rect "title"* [*options*] [*userdata*]

options:
- default
- disabled
- invisible *or* hidden
- native_just
- left_just
- center_just
- right_just

checkbox *id rect "title"* [*options*] [*userdata*]

options:
- checked
- disabled
- invisible *or* hidden
- native_just
- left_just
- center_just
- right_just

dialog *id rect "title"* [*type*] [*options*] [*userdata*] ... control statements ...

type:
- modal
- modeless

options:
- disabled
- invisible *or* hidden

edit *id rect "title"* [*options*] [*userdata*]

options:
- disabled
- invisible *or* hidden
- opt1 — Use 9-point Geneva font (Mac only)
- opt2 — Use 9-point Monaco font (Mac only)
- opt3 — Multiline (Mac only)
- opt4 — Wordwrap (Mac only)
- password

font *id "family" size* [*style*] [**map** *"font_descriptor"*]

style:

any	printer	user1
blink	scalable	user2
bold	shadow	user3
inverse	strikeout	user4
italic	underline	user5
outline		

font_map *id "font_descriptor"*

groupbox *id rect "title"* [*options*] [*userdata*]

options:

disabled	
invisible *or* hidden	
native_just (default)	
left_just	
center_just	
right_just	
opt1	Use 9-point Geneva font (Mac only)
opt2	Use 9-point Monaco font (Mac only)

icon *id rect ref-id* [*options*] [*userdata*]

options:

disabled	
invisible *or* hidden	

image *id "filename"* [*options*] [*userdata*]

options:

reference	

listbox *id rect* [*options*] [*userdata*]

options:

disabled	
invisible *or* hidden	
readonly	
multiple	
opt1	Use 9-point Geneva font (Mac only)
opt2	Use 9-point Monaco font (Mac only)

listbutton *id rect* [*options*] [*userdata*]

options:

invisible *or* hidden	
disabled	
native_just (default)	
left_just	
center_just	
right_just	

listedit *id rect* [*options*] [*userdata*]
options:
invisible *or* hidden
disabled
native_just (default)
left_just
center_just
right_just

menu *menu_tag "title"* [*options*] [*userdata*]
submenu *menu_tag "title"* [*options*]
item *menu_tag "title"* [*options*] [*userdata*]
item *menu_tag "title"* [*options*] [*userdata*]
...
separator
...
item *menu_tag "title"* [*options*] [*userdata*]
...
options for the **menu** and **submenu** statements:
disabled
options for the **item** statement:
disabled
checkable
checked

menubar *menu_tag*

radiobutton *id rect "title"* [*options*] [*userdata*]
options:
checked
disabled
invisible *or* hidden
group
native_just (default)
left_just
center_just
right_just

scrollbar *id rect* [*options*] [*userdata*]
options:
disabled
invisible *or* hidden

string *id "string"*

text *id rect "title"* [*options*] [*userdata*]
options:
disabled
invisible *or* hidden
native_just (default)
left_just
center_just
right_just
opt1 Use 9-point Geneva font (Mac only)
opt2 Use 9-point Monaco font (Mac only)

textedit *id rect "title"* [*options*] [*userdata*]

options:

autohscroll	limit integer	nocut
autovscroll	margin integer	nocopy
border	onepar	nopaste
enableclear	nomenu	overtype
readonly	invisible	disabled
wrap	font <fid>	

units *unit*

units:

pixels
chars
semichars

userdata "*string 1*", "*string 2*", ... , "*string n*"

window *id rect "title"* [*type*] [*options*] [*menu_id*] [*userdata*] ...control statements...

type:

doc	dbl_border
plain	no_border
modal	

options:

size	iconizable
close	iconized
hscroll	sizeonly
vscroll	maximized
invisible *or* hidden	no_menubar
disabled	defer_modal
place_exact	

#define *identifier substitution_text*
#define *identifier (identifier_list) substitution _text*

#if *constant_expression*
... text ...
#elif *constant_expression*
... text ...
#elif *constant_expression*
... text ...
#else
... text ...
#endif

#ifdef *identifier*
#ifndef *identifier*

#include <*path_specifcation*>
#include "*path_specification*"

#scan "*path_specification*"

#transparent *sentinel* [*literal*] [*no_include*]
... platform-specific code (ending with sentinel alone on line) ...

#undef *identifier*

Help File Source Statements

Comments '
Start beginning of comment line with a single quote.

HEADER
VERSION *version_number*
APPNAME “application name”

FONT code “*font_descriptor*”

BROWSE *sequence_name*

BODYSTANZA ... help topics ...

HTOPIC *topic_ID* “title” [keywords]

BTOPIC *topic_ID* “title” [keywords]

Formatting Codes

\Ppathname**\P**	Bitmap format code
\F*font_code*	Font Change format code
\Iindentation. <hanging_text><tab><paragraph_text>	Hanging Indentation format code
\Vwidth	Horizontal Line format code
\B*topic_ID*[:“string”]**\B**	Hot Button format code
\L*topic_ID*[:“string”]**\L**	Hyperlink format code
\M*margin_width*.	Margin format code
\W	Word Wrap format code
\N	No Word Wrap format code
\A	Paragraph format code

Logic Statements

#define *identifier substitution_text*
#define *identifier (identifier_list) substitution _text*

#if *constant_expression*
... text ...
#elif *constant_expression*
... text ...
#elif *constant_expression*
... text ...
#else
... text ...
#endif

#ifdef *identifier*
#ifndef *identifier*

#include <*path_specifcation*>
#include "*path_specification*"

#scan "*path_specification*"

#undef *identifier*

Predefined Help IDs

Help Menu Item IDs

#define XVT_TPC_CONTENTS ...
#define XVT_TPC_GLOSSARY ...
#define XVT_TPC_HELPONHELP ...
#define XVT_TPC_INDEX ...
#define XVT_TPC_KEYBOARD ...
#define XVT_TPC_ONVERSION ...
#define XVT_TPC_TUTORIAL ...

XVT Help Dialog IDs

#define XVT_TPC_ASK ...	xvt_dm_post_ask
#define XVT_TPC_ERROR ...	xvt_dm_post_error
#define XVT_TPC_FILE_OPEN ...	xvt_dm_post_file_open
#define XVT_TPC_FILE_SAVE ...	xvt_dm_post_file_save
#define XVT_TPC_FONT_SEL ...	xvt_dm_post_font_sel
#define XVT_TPC_MESSAGE ...	xvt_dm_post_message
#define XVT_TPC_NOTE ...	xvt_dm_post_note
#define XVT_TPC_PAGE_SETUP ...	xvt_dm_post_page_setup
#define XVT_TPC_STRING_PROMPT ...	xvt_dm_post_string_prompt
#define XVT_TPC_WARNING ...	xvt_dm_post_warning