

XBasic

Program Development Environment
(PDE)

GuiDesigner

Programmer Reference

*Revision 0.0015
November 1, 1995
Copyright 1990-2000*

Table of Contents

Standard Grid Properties.....	1
Abbreviations.....	1
Grid Property.....	1
Standard Grid Property.....	1
Custom Grid Property.....	1
General Purpose Grid Properties.....	2
Grab and Poke.....	2
Standard Grid Properties - List.....	3
Standard Grid Properties - Summary.....	4
Standard Grid Properties - Details.....	6
Messages.....	17
Overview.....	17
GraphicsDesigner and GuiDesigner Messages - Details.....	17
Message Processing Functions.....	49
Overview.....	49
Message Function Array - func[].....	49
Individuated Grids.....	49
Not Recommended.....	49
Standard Message Processing Functions.....	50
Standard Message Processing Functions - Details.....	50
Conversion Functions.....	65
Support Functions.....	67
Convenience Functions.....	71

Standard Grid Properties

Abbreviations

The following abbreviations appear throughout this document:

```
p property      - grid attribute like border, color, textString, etc
m message       - message : often to get or set a property
s subroutine    - message processing subroutine
f function      - message processing function
```

For example:

```
p align         - align property
m GetAlign     - GetAlign message - in code it's #GetAlign
m SetAlign     - SetAlign message - in code it's #SetAlign
s GetAlign     - message processing subroutine: SUB GetAlign
s SetAlign     - message processing subroutine: SUB SetAlign
f XuiGetAlign() - message processing function: XuiGetAlign()
f XuiSetAlign() - message processing function: XuiSetAlign()
```

As this example illustrates, property, message, subroutine, and function names are closely related.

To get / set the `align` property of a grid, programs send the grid `GetAlign` / `SetAlign` messages with `XuiSendMessage()`.

When the grid function responsible for the grid receives the message, it either ignores the message, calls a message processing function like `XuiGetAlign()` / `XuiSetAlign()`, calls a message processing subroutine like `SUB GetAlign` or `SUB SetAlign`, or both.

Grid Property

A *grid property* is a number, string or array with a name like `align`, `textArray`, `textString`, etc.

Standard Grid Property

A large set of *standard grid properties* are defined for all grid types. Standard message processing functions of the form `xuiGetProperty()` and `xuiSetProperty()` are built into *GuiDesigner* to fully or partly process most standard properties.

For example, `XuiGetBorder()` and `XuiSetBorder()` get and set the grid border properties. They are the default message processing functions for `GetBorder` and `SetBorder`.

Custom Grid Property

Additional grid properties can be defined for accessory grid types. But any grid type that defines a new grid property has to manage the storing and access of the values assigned to these new properties.

There is rarely need for new grid properties, however, since several *general purpose grid properties* are predefined and can hold strings and arrays for any purpose.

General Purpose Grid Properties

The most important *general purpose grid properties* are:

```
array      - ANY ARRAY for any purpose
textArray  - STRING ARRAY for any purpose
textString - STRING for any purpose
valueArray - XLONG array for any purpose
```

array is the least common general purpose property. It is most flexible however, because it can hold an array of *any* data type, including user-defined composite types. Since *GuiDesigner* has no way of knowing the array data type, however, *GuiDesigner* cannot *copy* the array to process **GetArray** and **SetArray** messages. *GuiDesigner* can grab and poke the original array itself by means of **ATTACH** or **SWAP** statements however, since **ATTACH** and **SWAP** are not data type sensitive. The following standard message processing functions / messages are built into *GuiDesigner*:

```
XuiGrabArray()    - process GrabArray messages
XuiPokeArray()    - process PokeArray messages
```

The **textArray** property holds an arbitrary size string array, the contents of which is grid type specific. The following standard message processing functions / messages are built into *GuiDesigner*:

```
XuiGetTextArray() - process GetTextArray messages
XuiSetTextArray() - process SetTextArray messages
XuiGrabTextArray() - process GrabTextArray messages
XuiPokeTextArray() - process PokeTextArray messages
```

The **textString** property holds an arbitrary size string, the contents of which are grid type specific. The following standard message processing functions / messages are built into *GuiDesigner*:

```
XuiGetTextString() - process GetTextString messages
XuiSetTextString() - process SetTextString messages
XuiGrabTextString() - process GrabTextString messages
XuiPokeTextString() - process PokeTextString messages
```

The **valueArray** property holds an arbitrary size **XLONG** array, the contents of which is grid type specific. The following standard message processing functions / messages are built into *GuiDesigner*:

```
XuiCreateValueArray() - process CreateValueArray messages
XuiGetValueArray()    - process GetValueArray messages
XuiSetValueArray()    - process SetValueArray messages
XuiGrabValueArray()   - process GrabValueArray messages
XuiPokeValueArray()   - process PokeValueArray messages
XuiGetValue()         - process GetValue messages
XuiSetValue()         - process SetValue messages
XuiGetValues()        - process GetValues messages
XuiSetValues()        - process SetValues messages
```

Grab and Poke

Grab and **Poke** messages and functions are unlike **Get** and **Set** in important ways. **Grab** messages remove the original array or string they refer to, leaving an empty array or string in the property until replaced with a subsequent **Poke** or **Set**. Similarly, arrays and strings passed by reference in **Poke** messages are stolen, so the returned argument is an empty array or string. Be careful.

Standard Grid Properties - List

grid	GraphicsDesigner grid number
gridName	name of grid (string)
gridType	grid type - XuiLabel, XuiPushButton, etc.
gridTypeName	name of grid type (string)
gridFunction	function that processes messages for this grid
gridFunctionName	name of grid function (string)
callbackGrid	grid argument in callback messages
callbackFunc	function to call with callback messages
window	window number of window that contains this grid
parent	parent grid
cursor	mouse cursor displayed when cursor enters grid
buffer	image grid to buffer the image of this grid
image	image number to display in grid
imageAlign	reserved for future enhancements
imageIndentX	display image this far from left of grid border
imageIndentY	display image this far below top of grid border
imageStartX	start displaying image this far from image left
imageStartY	start displaying image this far from image top
imageWidth	width of image to display
imageHeight	height of image to display
x	horizontal position of grid in parent grid
y	vertical position of grid in parent grid
width	width of grid
height	height of grid
maxWidth	maximum permissible width of this grid
maxHeight	maximum permissible height of this grid
minWidth	minimum permissible width of this grid
minHeight	minimum permissible height of this grid
backgroundColor	color to clear background of grid
drawingColor	draw and fill color for lines/circles/arcs/boxes
lowlightColor	darker color in 3D effects (shadow/highlight)
highlightColor	bright color in 3D effects (shadow/highlight)
dullColor	color for disabled and deemphasized items
accentColor	color for accented items
lowtextColor	darker color for 3D text (raise/lower/shadow)
hightextColor	bright color for 3D text (raise/lower/shadow)
redrawFlags	what needs redrawing (clear/border/text/etc)
timer	millisecond timer value
align	align text to 1 of 4 corners, 4 sides, center
justify	justify text left, right, center, both/full
texture	text texture (flat/lower/raise/shadow)
indentLeft	indent text from grid border at left
indentTop	indent text from grid border at top
indentRight	indent text from grid border at right
indentBottom	indent text from grid border at bottom
font	name of font typeface (string)
fontNumber	font number of typeface,size,weight,italic,angle
fontSize	size of font (10 * point size) (xlong)
fontWeight	weight/boldness of font (xlong)
fontItalic	italic/tilt of font (xlong)
border	current border style
borderUp	border style for up/primary state
borderDown	border style for down/secondary state
style	general purpose style number or style bits
styleMax	maximum style number
group	group number (to group radio boxes/buttons/etc)
can	can focus,respond,inputTextArray,inputTextString
focusKid	kid number that gets default keyboard focus
inputTextArray	kid number that can input a TextArray
inputTextString	kid number that can input a TextString
state	state (enable/disable keyboard/mouse/redraw)
keyboard	enable/disable keyboard messages
mouse	enable/disable mouse messages
redraw	enable/disable redraw messages
array	general purpose array of any data type
helpString	name of help entry or help string (string)
hintString	hint string (reserved)
kidArray	array of kid numbers (xlong)
messageFuncArray	array of message processing functions
messageSubArray	array of message processing subroutines
textArray	general purpose string array
textString	general purpose string
valueArray	general purpose array (xlong)

Standard Grid Properties - Summary

property	messages	- funcs / subs / comments
grid	Create CreateWindow GridFunction GridFunction	- SUB Create required in all grid functions - SUB CreateWindow required in all grid functions (@grid, #Create, x, y, width, height, window, parent) (@grid, #CreateWindow, xDisp, yDisp, width, height, winType, 0)
gridName	GetGridName SetGridName	- XuiGetGridName() - XuiSetGridName()
gridType	Initialize GetGridType SetGridType	- SUB Initialize required in all grid functions creates gridType - XuiRegisterGridType() in all grid functions creates gridType - XuiGetGridType() - XuiSetGridType()
gridTypeName	GetGridTypeName SetGridTypeName	- XuiGetGridTypeName() - XuiSetGridTypeName()
gridFunction	Initialize Create CreateWindow GetGridFunction SetGridFunction	- SUB Initialize required in all grid functions sets gridFunction - SUB Create required in all grid functions - SUB CreateWindow required in all grid functions - XuiGetGridFunction() - XuiSetGridFunction()
gridFunctionName	GetGridFunctionName SetGridFunctionName	- XuiGetGridFuncName() - XuiSetGridFuncName()
callbackGrid callbackFunc	GetCallback SetCallback	- XuiGetCallback() - XuiSetCallback()
window	Create GetWindow	- SUB Create in grid functions - XuiGetWindow()
parent	Create GetParent	- SUB Create in grid functions - XuiGetParent()
cursor	GetCursor SetCursor	- XuiGetCursor() - mouse cursor - XuiSetCursor() - mouse cursor
buffer	GetBuffer SetBuffer	- XuiGetBuffer() - XuiSetBuffer()
image imageIndentX imageIndentY imageName	GetImage SetImage	- XuiGetImage() - XuiSetImage()
imageStartX imageStartY imageWidth imageHeight	GetImageCoords SetImageCoords	- XuiGetImageCoords() - XuiSetImageCoords()
x y width height	Create CreateWindow Initialize GetSize Resize	- SUB Create in grid functions - SUB CreateWindow in grid functions - SUB Initialize in grid functions - XuiGetSize() - XuiResize()
maxWidth maxHeight minWidth minHeight	Initialize GetMaxMinSize SetMaxMinSize	- SUB Initialize in grid functions - XuiGetMaxMinSize() - XuiSetMaxMinSize()
backgroundColor drawingColor lowlightColor highlightColor	GetColor SetColor	- XuiGetColor() - XuiSetColor()
dullColor accentColor lowtextColor hightextColor	GetColorExtra SetColorExtra	- XuiGetColorExtra() - XuiSetColorExtra()
redrawFlags	GetRedrawFlags SetRedrawFlags RedrawGrid	- XuiGetRedrawFlags() - XuiSetRedrawFlags() - XuiRedrawGrid() - redraws grid based on redrawFlags

timer	GetTimer SetTimer StartTimer	- XuiGetTimer() - XuiSetTimer() - XuiStartTimer()
align	GetAlign SetAlign	- XuiGetAlign() - v1,v2,v3 = justify,indentLeft,indentTop - XuiSetAlign() - v1,v2,v3 > justify,indentLeft,indentTop
justify	GetJustify SetJustify	- XuiGetJustify() - v1,v2,v3 = align,indentLeft,indentTop - XuiSetJustify()
texture	GetTexture SetTexture	- XuiGetTexture() - text texture (flat/lower/raise/shadow) - XuiSetTexture() - text texture (flat/lower/raise/shadow)
indentLeft indentTop indentRight indentBottom	GetIndent SetIndent GetAlign SetAlign	- XuiGetIndent() - XuiSetIndent() - XuiGetAlign() - v2,v3 are indentLeft,indentTop - XuiSetAlign() - v2,v3 are indentLeft,indentTop
font fontSize fontWeight fontItalic fontAngle font\$	GetFontNumber SetFontNumber GetFont SetFont	- XuiGetFontNumber() - XuiSetFontNumber() - XuiGetFont() - XuiSetFont()
border borderUp borderDown	GetBorder SetBorder	- XuiGetBorder() - border width returned in r1 - XuiSetBorder()
style styleMax	GetStyle SetStyle	- XuiGetStyle() - v0,v1 = style,styleMax - XuiSetStyle() - v0,v1 = style,styleMax
group	GetGroup SetGroup	- XuiGetGroup() - XuiSetGroup()
can focusKid inputTextArray inputTextString	Initialize	- SUB Initialize in grid functions can bits for focus,respond,inputTextArray,inputTextString inputTextArray is kid # that can input text array from user inputTextString is kid # that can input text string from user
state keyboard mouse redraw	GetState SetState	- XuiGetState() - XuiSetState() state,keyboard,mouse,redraw are TRUE/FALSE (non-zero/zero)
array	GrabArray PokeArray	- XuiGrabArray() - XuiPokeArray()
helpString	GetHelpString SetHelpString	- XuiGetHelpString() - XuiSetHelpString()
hintString	GetHintString SetHintString	- XuiGetHintString() - XuiSetHintString()
kidArray	GetKidArray GetGridNumber GetParent GetKids	- XuiGetKidArray() - XuiGetGridNumber() - XuiGetParent() - XuiGetKids()
textArray	GetTextArray SetTextArray GrabTextArray PokeTextArray	- XuiGetTextArray() - XuiSetTextArray() - XuiGrabTextArray() - XuiPokeTextArray()
textString	GetTextString SetTextString	- XuiGetTextString() - XuiSetTextString()
valueArray	GetValue SetValue GetValues SetValues GetValueArray SetValueArray GrabValueArray PokeValueArray	- XuiGetValue() - XuiSetValue() - XuiGetValues() - XuiSetValues() - XuiGetValueArray() - XuiSetValueArray() - XuiGrabValueArray() - XuiPokeValueArray()

Standard Grid Properties - Details

grid	<p>grid is the grid number assigned to a grid when it is created. When a grid function is called with a Create or CreateWindow message its Create subroutine calls XuiCreateGrid() to create the grid and get the unique grid number then returned by the grid function.</p> <p>grid is the first argument to most <i>GraphicsDesigner</i> and <i>GuiDesigner</i> functions, including XuiSendMessage (grid, message...).</p> <pre>m #Create - create a grid in a window m #CreateWindow - create a window with a grid filling it m #GetGridNumber - return grid number of arbitrary grid s Create - creates a grid in a window s CreateWindow - creates a window and a grid filling it f XuiCreateGrid() - called by grid function Create subroutine f XuiGetGridNumber() - returns grid number of arbitrary grid</pre>
gridName	<p>gridName is the name of a grid. A poor default grid name is made from the grid type name and the grid number when a grid is created. Be sure to enter a descriptive name in the Appearance window for every grid - a name not repeated elsewhere in your program.</p> <pre>m #GetGridName m #SetGridName f XuiGetGridName() f XuiSetGridName()</pre>
gridType	<p>gridType is the grid type number. The first time a grid function is called, its Initialization subroutine calls XuiRegisterGridType() to establish its gridType and other properties.</p> <p>XuiLabel, XuiPushButton, XuiTextLine are typical grid types.</p> <pre>m #GetGridType m #SetGridType s Initialize f XuiRegisterGridType() f XuiGetGridType() f XuiSetGridType()</pre>
gridTypeName	<p>gridTypeName is the name of a grid type, which <i>must</i> be the same as gridFunctionName. The first time a grid function is called, its Initialization subroutine calls XuiRegisterGridType() to establish its gridTypeName and other properties.</p> <pre>m #GetGridTypeName m #SetGridTypeName s Initialize f XuiRegisterGridType() f XuiGetGridTypeName() f XuiSetGridTypeName()</pre>

gridFunction	<p>gridFunction is the address of the function called by <code>XuiSendMessage()</code> whenever a message is sent to a grid. Every grid is created and controlled by its grid function, and every grid function controls all grids of its grid type. The first time a grid function is called, its <code>Initialize</code> subroutine calls <code>XuiRegisterGridType()</code> to establish its gridFunction and other properties.</p> <pre> m #GetGridFunction m #SetGridFunction s Initialize f XuiRegisterGridType() f XuiGetGridFunction() f XuiSetGridFunction() </pre>
gridFunctionName	<p>gridFunctionName is the name of a grid function, which <i>must</i> be the same as gridTypeName. The first time a grid function is called, its <code>Initialize</code> subroutine calls <code>XuiRegisterGridType()</code> to establish its gridFunctionName and other properties.</p> <pre> m #GetGridFunctionName m #SetGridFunctionName s Initialize f XuiRegisterGridType() f XuiGetGridFunctionName() f XuiSetGridFunctionName() </pre>
callbackGrid callbackFunc	<p>callbackGrid is the grid number put in the grid argument of callback messages, while callbackFunc is the address of the callback function that <code>XuiCallback()</code> calls with callback messages.</p> <pre> m #GetCallback m #SetCallback f XuiGetCallback() f XuiSetCallback() </pre>
window	<p>window is the window number of the window that contains grid. window is established in <code>CreateWindow</code> subroutines.</p> <pre> m #Create m #CreateWindow m #GetWindow s Create s CreateWindow f XuiGetWindow() </pre>
parent	<p>parent is the grid number of a grids parent grid. A parent grid creates and often partially controls one or more "kid" grids. parent is established in the <code>Create</code> subroutine of grid functions.</p> <pre> m #Create m #CreateWindow m #GetParent s Create s CreateWindow f XuiGetParent() </pre>

cursor	<p>cursor is the cursor number of the mouse cursor displayed when the mouse cursor enters a grid.</p> <pre>m #GetCursor m #SetCursor f XuiGetCursor () f XuiSetCursor ()</pre>
buffer	<p>buffer is the grid number of the image grid buffering a grid.</p> <pre>m #GetBuffer m #SetBuffer f XuiGetBuffer () f XuiSetBuffer ()</pre>
imageName image imageAlign imageIndentX imageIndentY imageStartX imageStartY imageWidth imageHeight	<p>imageName is the filename of an image bitmap.</p> <p>image is the image grid assigned to a grid.</p> <p>imageAlign is reserved for future enhancements.</p> <p>imageIndentX is the distance to indent the left edge of the image from the left edge of a grid.</p> <p>imageIndentY is the distance to indent the top edge of the image from the top edge of a grid.</p> <p>imageStartX, imageStartY, imageWidth, imageHeight define the portion of the image grid to be drawn in a grid.</p> <pre>m #GetImage m #SetImage m #GetImageCoords m #SetImageCoords f XuiGetImage () f XuiSetImage () f XuiGetImageCoords () f XuiSetImageCoords ()</pre>
x y width height	<p>x,y are the coordinates of a grids upper-left corner in its parent.</p> <p>width,height is the size of a grid in pixels.</p> <pre>s Initialize m #Create m #CreateWindow m #GetSize m #SetSize m #Resize s Create s CreateWindow f XuiGetSize () f XuiSetSize () f XuiResize ()</pre>
minWidth minHeight maxWidth maxHeight	<p>minWidth, minHeight, maxWidth, maxHeight are the absolute minimum and maximum width and height a grid can be resized to in the most extreme circumstances. In particular circumstances, more restrictive minimums or maximums may apply.</p> <pre>s Initialize m #GetMaxMinSize m #SetMaxMinSize f XuiGetMessageFunc () f XuiSetMessageFunc ()</pre>
backgroundColor	<p>backgroundColor is the color grids are cleared to. The default</p>

drawingColor lowlightColor highlightColor	<p>backgroundColor usually defaults to light gray.</p> <p>drawingColor is the color that text and line graphics is drawn in. drawingColor usually defaults to black.</p> <p>lowlightColor is the color that shadows or downslopes are drawn in to create 3D effects, and usually defaults to black.</p> <p>highlightColor is the color that anti-shadows or upslopes are drawn in to create 3D effects, and usually defaults to white.</p> <pre> m #GetColor m #SetColor f XuiGetColor() f XuiSetColor() f XuiSetColorAll() f XuiGetDefaultColors() f XuiSetDefaultColors() </pre>
dullColor accentColor lowtextColor hightextColor	<p>dullColor is the color that disabled controls are drawn in, as in buttons or entries that cannot currently be selected. dullColor usually defaults to gray or cyan.</p> <p>accentColor is the color accented text or features are drawn in, as in selected text or selected entries. accentColor usually defaults to yellow.</p> <p>lowtextColor is the color drawn as the dark shadow in text with 3D texture, and usually defaults to black.</p> <p>hightextColor is the color drawn for the bright highlight in text with 3D texture, and usually defaults to white.</p> <pre> m #GetColorExtra m #SetColorExtra f XuiGetColorExtra() f XuiSetColorExtra() f XuiSetColorExtraAll() f XuiGetDefaultColors() f XuiSetDefaultColors() </pre>
redrawFlags	<p>redrawFlags tell what aspects of the grid need to be redrawn:</p> <pre> \$\$RedrawClip Clip grid to itself \$\$RedrawClear Clear grid before redrawing contents \$\$RedrawBuffer Refresh grid from image buffer \$\$RedrawImage Draw image into grid as specified \$\$RedrawBorder Draw border of grid \$\$RedrawTexture Draw text in specified texture \$\$RedrawTextArray Draw text array into grid \$\$RedrawTextString Draw text string into grid \$\$RedrawTextAccent Draw text in accent color \$\$RedrawTextDull Draw text in dull color </pre> <pre> m #GetRedrawFlags m #SetRedrawFlags f XuiGetRedrawFlags() f XuiSetRedrawFlags() </pre>
timer	<p>timer is a millisecond timer value that is loaded into the timer in response to <code>StartTimer</code> messages by grids that support a timer.</p> <pre> m #GetTimer m #KillTimer m #SetTimer m #StartTimer </pre>

	<pre>f XuiGetTimer() f XuiKillTimer() f XuiSetTimer() f XuiStartTimer()</pre>
align	<p>align is a text alignment property that determines the location of single or multi-line text. The align properties are:</p> <pre>\$\$AlignUpperLeft - upper left corner \$\$AlignUpperCenter - upper center side \$\$AlignUpperRight - upper right corner \$\$AlignMiddleLeft - middle left side \$\$AlignMiddleCenter - middle center \$\$AlignMiddleRight - middle right side \$\$AlignLowerLeft - lower left corner \$\$AlignLowerCenter - lower center side \$\$AlignLowerRight - lower right corner</pre> <pre>m #GetAlign m #SetAlign f XuiGetAlign() f XuiSetAlign()</pre>
justify	<p>justify determines the way text in a grid is justified:</p> <pre>\$\$JustifyLeft - left justify each line - "ragged right" \$\$JustifyCenter - center each line - "ragged left & right" \$\$JustifyRight - right justify each line - "ragged left" \$\$JustifyBoth - left & right justify text - "full justify"</pre> <pre>m #GetJustify m #SetJustify f XuiGetJustify() f XuiSetJustify()</pre>
texture	<p>texture is the current 3D texture style of text drawn in grids.</p> <p>The valid values of texture are at least :</p> <pre>\$\$TextureFlat \$\$TextureNone \$\$TextureLower4 \$\$TextureLower2 \$\$TextureLower1 \$\$TextureShadow \$\$TextureRaise1 \$\$TextureRaise2 \$\$TextureRaise4</pre> <p>By convention, 3D texture is reserved for text that cannot be changed by users, as on labels, buttons, etc. Also by convention, depressed texture is reserved for buttons and other controls that can be activated or selected, while raised texture is reserved for labels and other grids that cannot be activated or selected.</p> <pre>m #GetTexture m #SetTexture f XuiGetTexture() f XuiSetTexture()</pre>
indentLeft indentTop indentRight indentBottom	<p>indentLeft, indentTop, indentRight, indentBottom determine how far text is indented from the left, top, right, bottom of the border of grids. Not all values control text placement in all cases. For example, indentRight, indentBottom have no effect when the align property is \$\$AlignUpperLeft.</p> <pre>m #GetIndent m #SetIndent m #GetJustify m #GetAlign m #SetAlign</pre>

	<pre>f XuiGetIndent() f XuiSetIndent() f XuiGetJustify() f XuiGetAlign() f XuiSetAlign()</pre>
fontName fontNumber fontSize fontWeight fontItalic fontAngle	<p>fontName is a typeface name like Courier, Times New Roman, etc.</p> <p>fontNumber stands for a specific combination of typeface, size, weight, italic, and angle. 0 = default system font.</p> <p>fontSize is size of the font in .1 point units (10 point = 100).</p> <p>fontWeight is boldness weight from 0 to 1000.</p> <p>fontItalic is italic tilt from 0 to 1000.</p> <p>fontAngle is the angle of the baseline a series of characters will follow in .1 degree units (not always supported).</p> <pre>m #GetFont m #SetFont m #GetFontNumber m #SetFontNumber f XuiGetFont() f XuiSetFont() f XuiGetFontNumber() f XuiSetFontNumber()</pre>
border borderUp borderDown	<p>border, borderUp, borderDown are border style properties that determine the appearance of grid borders. In their normal state grids display themselves with border borders. When activated or selected, some grids alternately display borderUp/borderDown borders. border is the current border style.</p> <pre>\$\$BorderNone width = 0 \$\$BorderFlat1 width = 1 \$\$BorderFlat2 width = 2 \$\$BorderFlat4 width = 4 \$\$BorderHiLine1 width = 1 \$\$BorderHiLine2 width = 2 \$\$BorderHiLine4 width = 4 \$\$BorderLoLine1 width = 1 \$\$BorderLoLine2 width = 2 \$\$BorderLoLine4 width = 4 \$\$BorderRaise1 width = 1 \$\$BorderRaise2 width = 2 \$\$BorderRaise4 width = 4 \$\$BorderLower1 width = 1 \$\$BorderLower2 width = 2 \$\$BorderLower4 width = 4 \$\$BorderFrame width = 4 \$\$BorderDrain width = 4 \$\$BorderRidge width = 2 \$\$BorderValley width = 2 \$\$BorderWide.. ..width = 6 \$\$BorderResize width = 8</pre> <pre>m #GetBorder m #SetBorder f XuiGetBorder() f XuiSetBorder()</pre>
style styleMax	<p>style is a general purpose style designator for certain grids, and styleMax is the highest valid value for style.</p> <p>style=0 designates the default style for all grid types.</p> <pre>m #GetStyle</pre>

	<pre> m #SetStyle f XuiGetStyle() f XuiSetStyle() </pre>
group	<p>group is a group identifier. XuiRadioBox and XuiRadioButton grids send messages to all grids with the same group identifier within the same enclosing grid. group can be used for other related purposes as long as group numbers are unique.</p> <pre> m #GetGroup m #SetGroup s Initialize f XuiGetGroup() f XuiSetGroup() </pre>
can focusKid inputTextArray inputTextString	<p>can contains the following capability flags for a grid:</p> <pre> \$\$Focus - grid can accept keyboard focus \$\$Respond - grid can issue Selection callback messages \$\$InputTextArray - grid can input text array from user \$\$InputTextString - grid can input text string from user </pre> <p>If the \$\$Focus bit is set, the focusKid property is the kid # of the grid to initially give keyboard focus to. If the \$\$InputTextArray bit is set, the inputTextArray property is the kid # of the grid that modal convenience functions should get the textArray property from. If the \$\$InputTextString bit is set, the inputTextString property is the kid # of the grid that modal convenience functions should get the textString property from.</p> <pre> s Initialize m GetCan m SetCan f XuiGetCan() f XuiSetCan() </pre>
state keyboard mouse redraw	<p>state is a master grid enable. Most grids ignore keyboard, mouse, and redraw messages when state=0.</p> <p>keyboard is the master keyboard enable.</p> <p>mouse is the master mouse enable.</p> <p>redraw is the master redraw enable.</p> <pre> m #GetState m #SetState f XuiGetState() f XuiSetState() </pre>
helpString	<p>helpString is a text string that determines the help text displayed when help is requested for a grid. helpString can be one of three forms:</p> <ul style="list-style-type: none"> • filename:entryname • :entryname • help text with newline characters between text lines <p>If left empty, the help string defaults to :gridName.</p> <pre> m #GetHelpString m #SetHelpString f XuiGetHelpString() f XuiSetHelpString() </pre>

hintString	<p>hintString is a reserved text string...</p> <pre> m #GetHintString m #SetHintString f XuiGetHintString() f XuiSetHintString() </pre>
kidArray	<p>Kids are grids located within the boundaries of another grid, and to some extent controlled by or related to the grid. Kids are numbered 0,1,2,3,4... with kid 0 being the grid itself. The kidArray property is an array of grid numbers of the kids of a grid.</p> <pre> m #GetKids m #GetKidArray f XuiGetKids() f XuiGetKidArray() </pre>
messageFuncArray messageSubArray	<p>messageFuncArray contains message functions called in response to each message. For each message number there is a message function address, which is 0 for messages not processed by a message processing function.</p> <p>messageSubArray contains message subroutines called in response to each message. For each message number there is a message subroutine address, which is 0 for messages not processed by a message processing subroutine.</p> <pre> m #GetMessageFunc m #SetMessageFunc m #GetMessageFuncArray m #SetMessageFuncArray m #GetMessageSub m #GetMessageSubArray m #SetMessageSub m #SetMessageSubArray s Initialize f XuiGetMessageFunc() f XuiSetMessageFunc() f XuiGetMessageFuncArray() f XuiSetMessageFuncArray() f XuiGetMessageSub() f XuiSetMessageSub() f XuiGetMessageSubArray() f XuiSetMessageSubArray() </pre>
textArray	<p>textArray is a string array that can contain any number of text strings. The most common use of textArray is to hold lines of text, as in XuiTextArea text or list entries in XuiList.</p> <pre> m #GetTextArray m #SetTextArray m #GrabTextArray m #PokeTextArray f XuiGetTextArray() f XuiSetTextArray() f XuiGrabTextArray() f XuiPokeTextArray() </pre>
textString	<p>textString is a string that usually contains a single text line that typically is displayed by the grid.</p> <pre> m #GetTextString m #SetTextString f XuiGetTextString() f XuiSetTextString() </pre>
valueArray	<p>valueArray contains general purpose XLONG values that have different meanings for different grid types.</p>

```
m #GetValue
m #SetValue
m #GetValues
m #SetValues
m #GetValueArray
m #SetValueArray
m #GrabValueArray
m #PokeValueArray
f XuiGetValue()
f XuiSetValue()
f XuiGetValues()
f XuiSetValues()
f XuiGetValueArray()
f XuiSetValueArray()
f XuiGrabValueArray()
f XuiPokeValueArray()
```


Messages

Overview

Messages and message processing functions are often closely related. For example, `#GetBorder` and `#SetBorder` are messages programs send to grids to get and set their `border`, `borderUp`, `borderDown` properties, and `XuiGetBorder()` and `XuiSetBorder()` are the standard message processing functions provided to process these messages.

Programs call `XuiSendMessage()` to send messages to grids, as in:

```
XuiSendMessage (grid, #SetBorder, $$BorderRaise2, -1, -1, 0, kid, 0)
```

Grid functions can control themselves and their kids in the same way. But grid functions can also call message processing functions directly to control their own properties and behavior. A grid function might call `XuiSetBorder()` directly to control its own border. For example, the `MouseDown` message processing subroutine in `XuiPushButton` calls `XuiGetBorder()` and `XuiSetBorder()` directly to set the `border` property equal to the `borderDown` property, something like the following:

```
...
XuiGetBorder (grid, #GetBorder, @border, @up, @down, 0, 0, 0)
XuiSetBorder (grid, #SetBorder, down, up, down, 0, 0, 0)
...
```

GraphicsDesigner and GuiDesigner Messages - Details

Callback	<pre>(grid, message, v0, v1, v2, v3, @r0, r1)</pre> <p>Grid functions call <code>XuiCallback()</code> directly to send a callback message. If a callback function exists for the <code>grid</code>, <code>XuiCallback()</code> calls the callback function and passes it the arguments, except that the original <code>message</code> argument is transferred to <code>r1</code> and replaced by <code>#Callback</code>. <code>XuiCallback()</code> passes <code>r0</code> reference so callback functions potentially can cancel operations by returning <code>-1</code> in <code>r0</code>.</p> <pre>XuiCallback (grid, message, v0, v1, v2, v3, @r0, r1)</pre> <p><code>grid</code> - grid sending callback message <code>message</code> - message: callback message <code>v0,v1,v2,v3,r1</code> - original callback arguments <code>r0</code> - callback argument and possible cancel indicator</p>
Create SUB Create	<pre>(@grid, #Create, x, y, width, height, window, parent)</pre> <p>Call a grid function directly with a <code>create</code> message to create one grid of its grid type. Grid functions return the grid number of the created grid, or 0 if the grid could not be created for some reason. <code>create</code> messages must be processed in message processing subroutines within grid functions, so there are no standard message functions to handle <code>create</code> messages.</p> <p>Programs can call <code>XuiSendMessage()</code> to send all other messages, but must call grid functions directly to send <code>create</code> and <code>CreateWindow</code>:</p> <pre>wrong: XuiSendMessage (@grid, #Create, x, y, w, y, window, parent)</pre>

	<pre>right: XuiTextLine (@grid, #Create, x, y, w, h, window, parent) GridFunc (@grid, #Create, x, y, width, height, window, parent) grid - grid number of grid created by grid function #Create - message: create a grid x, y - x,y coordinates of upper-left corner in parent width, height - width,height size of grid in pixels window - window number of window the grid is put in parent - parent grid of the created grid (enclosing grid)</pre>
<p>CreateWindow</p> <p>SUB CreateWindow</p>	<pre>(@grid, #CreateWindow, xDisp, yDisp, width, height, winType, display\$)</pre> <p>Call a grid function directly with a <code>CreateWindow</code> message to create one grid of their grid type in a window just big enough to contain the grid. Grid functions return the grid number of the created grid and window, or 0 if they could not be created for some reason. <code>CreateWindow</code> messages must be processed in message processing subroutines within grid functions, so no standard message functions are provided for <code>CreateWindow</code>.</p> <p>Programs can call <code>XuiSendMessage()</code> to send all other messages, but must call grid functions directly to send <code>Create</code> and <code>CreateWindow</code>:</p> <pre>wrong: XuiSendMessage (@grid, #CreateWindow, x, y, w, y, wt, 0) right: XuiDialog3B (@grid, #CreateWindow, x, y, w, h, wt, 0) GridFunc (@grid, #CreateWindow, xDisp, yDisp, width, height, winType, disp\$) grid - grid number of grid created by grid function #CreateWindow - message: create a window and grid xDisp, yDisp - x,y display coordinates of upper-left corner of grid width, height - width,height size of grid in pixels winType - window type to create and parent window disp\$ - display name to contain window winType - window type bits OR parent window - window type bits are in high 16-bits - parent window is in low 16-bits - parent window is usually 0 except menu pulldown lists winType constant - meaning (in xgr.dec file : IMPORT "xgr") \$\$WindowTopMost - stays above other windows \$\$WindowNoSelect - window is not selected by mouse button events \$\$WindowNoFrame - window has no resize frame \$\$WindowResizeFrame - window has a resize frame \$\$WindowTitleBar - window has a title bar \$\$WindowSystemMenu - window has a system menu button \$\$WindowMinimizeBox - window has a minimize button \$\$WindowMaximizeBox - window has a maximize button</pre>
Destroy	<pre>(grid, #Destroy, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>Destroy</code> to a grid to destroy it. If the grid is attached to a window, that is, it was created by <code>CreateWindow</code>, the window containing the grid is also destroyed and removed from the display.</p> <p>Do not send messages to grids after they are destroyed. Their grid numbers are reassigned to subsequently created grids, and the messages are erroneously sent to the new grids.</p> <pre>XuiSendMessage (grid, #Destroy, 0, 0, 0, 0, kid, 0) XuiDestroy (grid, #Destroy, 0, 0, 0, 0, 0, 0) grid,kid - target grid #Destroy - message: destroy grids</pre>
Destroyed	<pre>(grid, #Destroyed, 0, 0, 0, 0, kid, 0)</pre> <p>When a user destroys a window manually, a <code>WindowDestroyed</code> message is sent to the window function assigned to the destroyed window.</p>

	<p>The window function sends a <code>Destroyed</code> message to every parentless grid in the destroyed window. Grids propagate <code>Destroyed</code> messages to their kids.</p> <pre>XuiSendMessage (grid, #Destroyed, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#Destroyed</code> - message: notify grids they've been destroyed</p>
DestroyWindow	<pre>(grid, #DestroyWindow, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>DestroyWindow</code> to any grid in a window to destroy the window and all kids in the window.</p> <pre>XuiSendMessage (grid, #DestroyWindow, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid (any grid in the window) <code>#DestroyWindow</code> - message: destroy the window containing the grid</p>
Disable	<pre>(grid, #Disable, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>Disable</code> to a grid to have <i>GraphicsDesigner</i> stop sending keyboard and mouse messages to the grid. Grids propagate <code>Disable</code> messages to their kids.</p> <pre>XuiSendMessage (grid, #Disable, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#Disable</code> - message: disable grid</p>
DisplayWindow	<pre>(grid, #DisplayWindow, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>DisplayWindow</code> to any grid in a window to display the window.</p> <pre>XuiSendMessage (grid, #DisplayWindow, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid (any grid in the window) <code>#DisplayWindow</code> - message: display the window containing the grid</p>
Enable	<pre>(grid, #Enable, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>Enable</code> to a grid to have <i>GraphicsDesigner</i> resume sending keyboard and mouse messages to the grid. Grids propagate <code>Enable</code> messages to their kids.</p> <pre>XuiSendMessage (grid, #Enable, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#Enable</code> - message: enable grids</p>
GetBorder	<pre>(grid, #GetBorder, @border, @borderUp, @borderDown, 0, kid, @width)</pre> <p>Send <code>GetBorder</code> to a grid to get its current border style, two alternate border styles, and border width of the current border style.</p> <pre>XuiSendMessage (grid, GetBorder, @border, @borderUp, @borderDown, 0, kid, @width)</pre> <p><code>grid,kid</code> - target grid <code>#GetBorder</code> - message: get grid border styles, etc... <code>border</code> - the current border style <code>borderUp</code> - alternate border style <code>borderDown</code> - alternate border style <code>width</code> - width of current border style</p>
GetCallback	<pre>(grid, #GetCallback, @callGrid, @callFunc, @v2, @v3, @r0, 0)</pre> <p>Send <code>GetCallback</code> to a grid to get its callback information, which includes a callback grid, a callback function address, and callback indices.</p> <pre>XuiSendMessage (grid, #GetCallback, @callGrid, @callFunc, @v2, @v3, @r0, 0)</pre>

	<pre> grid - target grid #GetCallback - message: get the callback grid/function/indices callGrid - grid argument in callback message callFunc - address of function callback messages are sent to v2,v3,r0 - values sent back with callback messages (if not -1) </pre>
GetCallbackArgs	<pre>(@g, #GetCallbackArgs, @v0, @v1, @v2, @v3, @r0, @r1)</pre> <p>Call <code>XuiCallback()</code> directly with a <code>GetCallbackArgs</code> message to retrieve the most recently sent callback arguments, some of which may have been replaced with values registered when the callback function was set.</p> <p>When a callback function is set, values can be registered to be sent in <code>v2,v3,r0,r1</code> whenever the grid sends a callback message. When the grid calls <code>XuiCallback()</code> to send a callback message, any of the <code>v2,v3,r0,r1</code> arguments that was not <code>-1</code> in the <code>SetCallback</code> call is sent with the registered callback value.</p> <p>Calling <code>XuiCallback()</code> with a <code>GetCallbackArgs</code> message retrieves the original <code>v2,v3,r0,r1</code> the grid sent.</p> <pre> XuiCallback (@g, #GetCallbackArgs, @v0, @v1, @v2, @v3, @r0, @r1) #GetCallbackArgs - message: get original callback arguments g,v0,v1,v2,v3,r0,r1 - original callback arguments sent by grid </pre>
GetColor	<pre>(grid, #GetColor, @back, @draw, @low, @high, kid, 0)</pre> <p>Send <code>GetColor</code> to a grid to get its background, drawing, highlight, and lowlight colors.</p> <pre> XuiSendMessage (grid, #GetColor, @back, @draw, @low, @high, kid, 0) grid,kid - target grid #GetColor - message: get main grid colors back - background color draw - drawing color low - lowlight color (3D down slope color) high - highlight color (3D up slope color) </pre>
GetColorExtra	<pre>(grid, #GetColorExtra, @dull, @accent, @loText, @hiText, kid, 0)</pre> <p>Send <code>GetColorExtra</code> to a grid to get its dull, accent, lowtext, and hightext colors.</p> <pre> XuiSendMessage (grid, #GetColorExtra, @dull, @acc, @lt, @ht, kid, 0) grid,kid - target grid #GetColorExtra - message: get auxiliary grid colors dull - dull color (disabled features, etc...) accent - accent color (selected text, etc...) loText - lowText color (for 3D text texture) hiText - accent color (for 3D text texture) </pre>
GetCursor	<pre>(grid, #GetCursor, @cursor, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetCursor</code> to a grid to get the cursor number of the cursor that displays when the mouse cursor enters grid.</p> <pre> grid,kid - target grid #GetCursor - message: get cursor number (0 = default cursor) cursor - cursor number </pre>
GetDisplay	<pre>(grid, #GetDisplay, 0, 0, 0, 0, kid, @display\$)</pre> <p>Send <code>GetDisplay</code> to any grid in any window in a display to get the name of the display. The default display is an empty string "" .</p>

	<pre>XuiSendMessage (grid, #GetDisplay, 0, 0, 0, 0, kid, @display\$)</pre> <p>grid,kid - target grid #GetDisplay - message: get display containing grid,kid display\$ - name of display</p>
GetEnclosedGrids	<pre>(grid, #GetEnclosedGrids, 0, 0, 0, kid, @grid[])</pre> <p>Send <code>GetEnclosedGrids</code> to a grid to get an array containing the grid numbers of all grids whose grid-boxes are completely contained within the grid. Coincident boundaries are considered enclosed. Some part of a grid must fall outside to be classified as not enclosed.</p> <p>grid,kid - target grid #GetEnclosedGrids - message: get array of enclosed grid numbers grid[] - array of enclosed grid numbers (empty if none)</p>
GetEnclosingGrid	<pre>(grid, #GetEnclosingGrid, @enclosingGrid, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetEnclosingGrid</code> to a grid to get the grid number of the smallest grid that completely encloses the grid. Coincident boundaries are considered enclosed. In other words, if no part of a grid extends outside another, it is enclosed by that enclosing grid.</p> <p>grid,kid - target grid #GetEnclosingGrid - message: get smallest enclosing grid number enclosingGrid - grid number of smallest enclosing grid</p>
GetFont	<pre>(grid, #GetFont, @size, @weight, @italic, @angle, kid, @font\$)</pre> <p>Send <code>GetFont</code> to a grid to get its font name and characteristics.</p> <p><code>XgrCreateFont()</code> assigns a unique font number for each combination of typeface, size, weight, italic, and angle.</p> <p>grid,kid - target grid #GetFont - message: get font assigned to grid size - font size in 1/20 point units weight - font boldness from 0 to 1000 - 400 is normal italic - font italic tilt from 0 to 1000 - 0 is no italic angle - font baseline tilt in .1 degree units</p>
GetFontNumber	<pre>(grid, #GetFontNumber, @font, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetFontNumber</code> to a grid to get its font number.</p> <p><code>XgrCreateFont()</code> assigns a unique font number for each combination of typeface, size, weight, italic, and angle.</p> <p>grid,kid - target grid #GetFontNumber - message: get font number assigned to grid font - font number</p>
GetGridFunction	<pre>(grid, #GetGridFunction, @func, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetGridFunction</code> to a grid to get the address of its grid function.</p> <pre>XuiSendMessage (grid, #GetGridFunction, @g, @func, 0, 0, kid, 0)</pre> <p>grid,kid - target grid #GetGridFunction - message: get grid number and grid function func - address of grid function assigned to target grid</p>
GetGridFunctionName	<pre>(grid, #GetGridFunctionName, @func, 0, 0, 0, kid, func\$)</pre> <p>Send <code>GetGridFunctionName</code> to a grid to get the name of its grid function. The address of the grid function is also available.</p> <pre>XuiSendMessage (grid, #GetGridFunction, @func, 0, 0, 0, kid, func\$)</pre> <p>grid,kid - target grid</p>

	<pre>#GetGridFunctionName - message: get grid function name func - address of grid function assigned to grid func\$ - name of grid function assigned to grid</pre>
GetGridName	<pre>(grid, #GetGridName, @gridNumber, 0, 0, 0, kid, @gridName\$)</pre> <p>Send <code>GetGridName</code> to a grid to get its grid name. Default grid names are assigned to grids when they are created, and you should assign descriptive grid names during the design process.</p> <pre>XuiSendMessage (grid, #GetGridName, @gridNumber, 0, 0, 0, kid, @gridName\$)</pre> <pre>grid,kid - target grid #GetGridName - message: get grid name gridNumber - grid number of grid,kid gridName\$ - grid name of grid,kid</pre>
GetGridNumber	<pre>(grid, #GetGridNumber, @g, @kid1, @kid2, @kid3, kid, @parent)</pre> <p>Send <code>GetGridNumber</code> to a grid to get its grid number. The grid numbers of its first three kids and its parent are also returned.</p> <pre>XuiSendMessage (grid, #GetGridNumber, @g, @kid1, @kid2, @kid3, kid, @parent)</pre> <pre>grid,kid - target grid #GetGridNumber - message to get grid number of a grid g - grid number of grid,kid kid1,kid2,kid3 - grid number of kid #1, kid #2, kid #3 of grid,kid parent - grid number of parent of grid,kid</pre>
GetGridType	<pre>(grid, #GetGridType, @gridType, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetGridType</code> to a grid to get its grid type.</p> <pre>XuiSendMessage (grid, #GetGridType, @gridType, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetGridType - message: get grid type gridType - grid type of grid</pre>
GetGridTypeName	<pre>(grid, #GetGridTypeName, @gridType, 0, 0, 0, kid, gridType\$)</pre> <p>Send <code>GetGridTypeName</code> to a grid to get the name of its grid function. The <code>gridType</code> number is also available.</p> <pre>XuiSendMessage (grid, #GetGridTypeName, @func, 0, 0, 0, kid, func\$)</pre> <pre>grid,kid - target grid #GetGridTypeName - message: get grid type name gridType - grid type number gridType\$ - grid type name</pre>
GetHelp	<pre>(grid, #GetHelp, x, y, state, time, @kid, @help\$)</pre> <p><code>GetHelp</code> is sent to a grid by the <code>InstantHelp</code> window when the user presses the right mouse button over the grid. <code>GetHelp</code> can also be send to a grid to achieve the same effect. The <code>InstantHelp</code> window sends the <code>x,y,state,time</code> arguments from the mouse event, so programs can send all zeros to distinguish itself from mouse events.</p> <p><code>XuiGetHelp()</code>, the standard message processing function for the <code>GetHelp</code> message, does the following:</p> <p>Sends a <code>Help</code> callback message on behalf of the grid. The callback function can display customized help by sending <code>SetHelp</code> to <code>XuiHelp()</code>. The callback function can return <code>r0=-1</code> to cancel the normal help request response. If <code>XuiGetHelp()</code> detects <code>r0=-1</code>, it returns this cancel notice and an empty help string. If the help string for the grid is not empty, <code>XuiGetHelp()</code> returns the help string in <code>help\$</code>. If the help string is empty, <code>XuiGetHelp()</code> returns ("<code>:</code>" + <code>gridName\$</code>) in <code>help\$</code>.</p> <pre>grid,kid - target grid #GetHelp - message: get help string to display</pre>

	<pre> x,y - mouse coordinates if initiated by MouseDown state,time - mouse information if initiated by MouseDown @kid - return as -1 to cancel the help response help\$ - the help string (file:entry, :entry, help\$, or "") </pre>
GetHelpFile	<pre> (grid, #GetHelpFile, 0, 0, 0, 0, kid, @helpfile\$) grid,kid - target grid (any valid grid) #GetHelpFile - message: get current default help file helpFile\$ - help file name </pre>
GetHelpString	<pre> (grid, #GetHelpString, 0, 0, 0, 0, kid, @help\$) </pre> <p>Send <code>GetHelpString</code> to a grid to get its current help string. This can be an empty string, "filename:entryname", ":entryname", or the help text itself. Also see the <code>GetHelp</code> message.</p> <pre> XuiSendMessage (grid, #GetHelpString, 0, 0, 0, 0, kid, @help\$) grid,kid - target grid #GetHelpString - message: get help string from grid help\$ - help string from grid </pre>
GetHelpStrings	<pre> (grid, #GetHelpStrings, 0, 0, 0, 0, kid, @help\$[]) </pre> <p>Send <code>GetHelpStrings</code> to a grid to get its help string and the help strings from all its kids. Each help string can be an empty string, "filename:entryname", ":entryname", or the help text itself.</p> <pre> XuiSendMessage (grid, #GetHelpStrings, 0, 0, 0, 0, kid, @help\$[]) grid,kid - target grid #GetHelpStrings - message: get help string from grid and its kids help\$[] - help string from grid and its kids </pre>
GetImage	<pre> (grid, #GetImage, @image, 0, @indentX, @indentY, kid, 0) </pre> <p>Send <code>GetImage</code> to a grid to get the grid number of the image grid assigned to the grid, as well as the indent from the left and top of the grid that the image should be drawn. Not all grids display images.</p> <pre> XuiSendMessage (grid, #GetImage, @image, 0, @indentX, @indentY, kid, 0) grid,kid - target grid #GetImage - message: get grid number of image indentX - indent from left of grid to left of image indentY - indent from top of grid to top of image </pre>
GetImageCoords	<pre> (grid, #GetImageCoords, @offsetX, @offsetY, @width, @height, kid, 0) </pre> <p>Send <code>GetImageCoords</code> to a grid to get the coordinates of the portion of the image grid that are to drawn into the grid. These are 0,0 relative coordinates in the image, so they determine what portion of the image is drawn in the grid, not where the image is drawn in the grid. See <code>indentX,indentY</code> in <code>GetImage</code>.</p> <pre> XuiSendMessage (grid, #GetImageCoords, @offsetX, @offsetY, @width, @height, kid, 0) grid,kid - target grid #GetImageCoords - message: get coordinate of portion of image to draw offsetX - offset from image left to left of portion to draw offsetY - offset from image top to top of portion to draw width - width of portion of image to draw height - height of portion of image to draw </pre>
GetJustify	<pre> (grid, #GetJustify, @justify, @align, @inL, @inT, kid, bw) </pre> <p>Send <code>GetJustify</code> to a grid to get its text justify, align, <code>indentLeft</code>, <code>indentTop</code> properties, and its border width.</p> <pre> XuiSendMessage (grid, #GetJustify, @justify, @align, @inL, @inT, kid, @bw) grid,kid - target grid </pre>

	<pre>#GetJustify - message: get justify, align, indent, border width justify - text justify property align - text align property inL,inT - text indentLeft,indentTop property bw - width of the border style</pre>
GetKeyboardFocus	<pre>(grid, #GetKeyboardFocus, @focusGrid, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetKeyboardFocus</code> to any grid to get the grid that currently has keyboard focus. The grid with keyboard focus may not be in the same window.</p> <pre>XuiSendMessage (grid, #GetKeyboardFocus, @focusGrid, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetKeyboardFocus - message: get current keyboard focus grid number focusGrid - grid number of grid that has keyboard focus</pre>
GetKeyboardFocusGrid	<pre>(grid, #GetKeyboardFocusGrid, @focusGrid, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetKeyboardFocusGrid</code> to any grid in a window to get the grid number of the grid that will get keyboard focus the next time the window is selected. If the window is currently selected, <code>focusGrid</code> currently has keyboard focus.</p> <pre>XuiSendMessage (grid, #GetKeyboardFocusGrid, @focusGrid, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetKeyboardFocusGrid - message: get keyboard focus grid in window focusGrid - grid that will get keyboard focus in window</pre>
GetKidArray	<pre>(grid, #GetKidArray, 0, 0, 0, 0, kid, @kid[])</pre> <p>Send <code>GetKidArray</code> to a grid to get an array containing the grid numbers of the grid kids. <code>kid[0]</code> is the grid number of the grid.</p> <pre>XuiSendMessage (grid, GetKidArray, 0, 0, 0, 0, kid, @kid[])</pre> <pre>grid,kid - target grid #GetKidArray - message: get array of kid grid numbers kid[] - grid numbers of grid,kid and its kids</pre>
GetKids	<pre>(grid, #GetKids, @g, @k1, @k2, @k3, kid, @k4)</pre> <p>Send <code>GetKids</code> to a grid to get an array containing the grid number of the grid and its kids. <code>g</code> is the grid number of the target grid.</p> <pre>XuiSendMessage (grid, GetKids, @g, @k1, @k2, @k3, kid, @k4)</pre> <pre>grid,kid - target grid #GetKids - message: get array of kid grid numbers g - grid number of grid,kid k1,k2,k3,k4 - grid numbers of 1st 4 kids of grid,kid</pre>
GetMaxMinSize	<pre>(grid, #GetMaxMinSize, @maxW, @maxH, @minW, @minH, kid, 0)</pre> <p>Send <code>GetMaxMinSize</code> to a grid to get is maximum and minimum width,height under any circumstances. See <code>GetSmallestSize</code>.</p> <pre>XuiSendMessage (grid, #GetMaxMinSize, @maxW, @maxH, @minW, @minH, kid, 0)</pre> <pre>grid,kid - target grid #GetMaxMinSize - message: get max/min width/height of grid,kid maxW,maxH - maximum possible width and height of grid,kid minW,minH - minimum possible width and height of grid,kid</pre>
GetMessageFunc	<pre>(grid, #GetMessageFunc, message, @func, 0, 0, kid, 0)</pre> <p>Send <code>GetMessageFunc</code> to a grid to get the address of the function that handles message number <code>message</code>.</p> <pre>XuiSendMessage (grid, #GetMessageFunc, message, @func, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid</pre>

	<pre>#GetMessageFunc - message: get message function address for a message message - message number to get function address for func - address of function executed in response to msg</pre>
GetMessageFuncArray	<pre>(grid, #GetMessageFuncArray, 0, 0, 0, 0, kid, @func[])</pre> <p>Send <code>GetMessageFuncArray</code> to a grid to get the addresses of the functions that process messages for the grid.</p> <pre>XuiSendMessage (grid, #GetMessageFuncArray, 0,0,0,0, kid, @func[])</pre> <pre>grid,kid - target grid #GetMessageFuncArray - message: get message function addresses func[] - message function addresses</pre>
GetMessageSub	<pre>(grid, #GetMessageSub, message, @sub, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetMessageSub - message: get message subroutine address message - message to get subroutine address for sub - subroutine address</pre>
GetMessageSubArray	<pre>(grid, #GetMessageSubArray, 0, 0, 0, 0, kid, @sub[])</pre> <pre>grid,kid - target grid #GetMessageSubArray - message: get array of message subroutines sub[] - array of message subroutine addresses</pre>
GetModalInfo	<pre>(grid, #GetModalInfo, @v0, @v1, @v2, @v3, kid, @r1)</pre> <pre>grid,kid - target grid #GetModalInfo - message: display modal window, accept response v0,v1,v2,v3,r1 - arguments returned by response callback</pre>
GetParent	<pre>(grid, #GetParent, @parent, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetParent</code> to a grid to get its grid number and the grid number of its parent grid.</p> <pre>XuiSendMessage (grid, #GetParent, @parent, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetParent - message to get grid number of parent grid parent - grid number of parent of grid,kid</pre>
GetSize	<pre>(grid, #GetSize, @x, @y, @width, @height, kid, 0)</pre> <p>Send <code>GetSize</code> to a grid to get the position of its upper-left corner in window coordinates, and its size in pixels.</p> <pre>XuiSendMessage (grid, #GetSize, @x, @y, @width, @height, kid, 0)</pre> <pre>grid,kid - target grid #GetSize - message: get grid position in window and width,height x,y - upper-left corner of grid,kid in parent width,height - width and height of grid,kid in pixels</pre>
GetSmallestSize	<pre>(grid, #GetSmallestSize, 0, 0, @width, @height, kid, 0)</pre> <p>Send <code>GetSmallestSize</code> to a grid to get the smallest size it can be resized to and still reasonably contain its contents. Not every grid type supports this message, so routines should be prepared to have the <code>width,height</code> arguments returned unchanged, or possibly with the current grid size returned by <code>GetSize</code>, or possibly filled with the grid minimums returned by <code>GetMaxMinSize</code>.</p> <pre>XuiSendMessage (grid, #GetSmallestSize, 0, 0, @width, @height, kid, 0)</pre> <pre>grid,kid - target grid #GetSmallestSize - message: get smallest size that fits contents width,height - smallest width and height</pre>
GetState	<pre>(grid, #GetState, @state, @keyboard, @mouse, @redraw, kid, 0)</pre>

	<p>Send <code>GetState</code> to a grid to get its master disabled/enabled state, as well as disabled/enabled states for keyboard messages, mouse messages, and redraw messages.</p> <p>See the <code>Disable</code> and <code>Enable</code> messages for related information.</p> <pre>XuiSendMessage (grid, #GetState, @state, @keyboard, @mouse, @redraw, kid, 0)</pre> <p>grid,kid - target grid #GetState - message: get disable/enable state (FALSE / TRUE) state - FALSE/TRUE = disabled/enabled (everything) keyboard - FALSE/TRUE = disabled/enabled keyboard mouse - FALSE/TRUE = disabled/enabled mouse redraw - FALSE/TRUE = disabled/enabled redraw</p>
GetStateFlags	<pre>(grid, #GetStateFlags, @state, @key, @mouse, @redraw, kid, @enable)</pre> <p>grid,kid - target grid #GetStateFlags - message: get state flags state - overall keyboard/mouse/redraw disable/enable key - keyboard disable/enable mouse - mouse disable/enable redraw - redraw disable/enable enable - GraphicsDesigner master disable/enable state - #Disable/#Enable messages & XuiDisable()/XuiEnable()</p>
GetStyle	<pre>(grid, #GetStyle, @style, styleMax, 0, 0, kid, 0)</pre> <p>grid,kid - target grid #GetStyle - message: get style property style - style property maxStyle - maximum valid value for style property</p>
GetTextArray	<pre>(grid, #GetTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p>Send <code>GetTextArray</code> to a grid to get its text array.</p> <pre>XuiSendMessage (grid, #GetTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p>grid,kid - target grid #GetTextArray - message: get text array text\$[] - text array from grid,kid</p>
GetTextArrayLine	<pre>(grid, #GetTextArrayLine, line, 0, 0, @upper, kid, @text\$)</pre> <p>Send <code>GetTextArrayLine</code> to a grid to get one text string from its text array. The lower bound of the text array is 0.</p> <pre>XuiSendMessage (grid, #GetTextArrayLine, line, 0, 0, @upper, kid, @text\$)</pre> <p>grid,kid - target grid #GetTextArrayLine - message: get one line from text array line - line number to get (0 to UBOUND(text\$[])) upper - UBOUND(text\$[]) text\$ - text line from text array in grid,kid</p>
GetTextArrayLines	<pre>(grid, #GetTextArrayLines, first, count, 0, @upper, kid, @text\$[])</pre> <p>Send <code>GetTextArrayLines</code> to a grid to get consecutive lines from its text array. The lower bound of the text array is 0.</p> <pre>XuiSendMessage (grid, #GetTextArrayLines, first, count, 0, @upper, kid, @text\$[])</pre> <p>grid,kid - target grid #GetTextArrayLines - message: get consecutive lines from text array first - first line number to get (starts at 0) count - number of lines to get upper - upper bound of text\$[] text\$[] - text array with lines from text array</p>
GetTextString	<pre>(grid, #GetTextString, 0, 0, 0, 0, kid, @text\$)</pre> <p>Send <code>GetTextString</code> to a grid to get its text string.</p> <pre>XuiSendMessage (grid, #GetTextString, 0, 0, 0, 0, kid, @text\$)</pre>

	<pre>grid,kid - target grid #GetTextString - message: get text string text\$ - text string from grid,kid</pre>
GetTextStrings	<pre>(grid, #GetTextStrings, 0, 0, 0, 0, kid, @text\$[])</pre> <p>Send <code>GetTextStrings</code> to a grid to get the text strings from the grid and its kids.</p> <pre>XuiSendMessage (grid, #GetTextStrings, 0, 0, 0, 0, kid, @text\$[])</pre> <pre>grid,kid - target grid #GetTextStrings - message: get text string from grid,kid and its kids text\$[] - text strings from grid,kid and its kids</pre>
GetTexture	<pre>(grid, #GetTexture, @texture, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetTexture - message: get text texture (Flat/Lower/Raise/Shadow) texture - texture property</pre>
GetValueArray	<pre>(grid, #GetValueArray, 0, 0, 0, 0, kid, @value[])</pre> <pre>grid,kid - target grid #GetValueArray - message: get copy of XLONG value array property value[] - XLONG value array property</pre>
GetValue	<pre>(grid, #GetValue, @v0, 0, 0, 0, kid, index)</pre> <pre>grid,kid - target grid #GetValue - message: get one element of XLONG value array v0 - XLONG value from value array - value[index] index - element in value array to return</pre>
GetValues	<pre>(grid, #GetValues, @v0, @v1, @v2, @v3, kid, index)</pre> <p>Send <code>GetValues</code> to a grid to get four elements from its <code>XLONG value[]</code> array starting at <code>value[r1]</code>.</p> <pre>XuiSendMessage (grid, GetValues, @v0, @v1, @v2, @v3, kid, @v4)</pre> <pre>grid,kid - target grid #GetValues - message: return four sequential values from value[] v0,v1,v2,v3 - value[r1+0] to value[r1+3] r1 - index into value[]</pre>
GetWindow	<pre>(grid, #GetWindow, @window, @winType, @func, @windowGrid, kid, @focusGrid)</pre> <p>Send <code>GetWindow</code> to a grid to get the window number of the window containing it. The <code>windowType</code>, <code>windowFunction</code>, <code>windowGrid</code>, and <code>keyboardFocusGrid</code> are also returned.</p> <pre>XuiSendMessage (grid, #GetWindow, @window, @winType, @func, @windowGrid, kid, @focusGrid)</pre> <pre>grid,kid - target grid #GetWindow - message: get window containing grid,kid window - window number of window containing grid,kid winType - window type assigned to window at create time func - window function assigned to window windowGrid - grid attached to window (so they resize together) focusGrid - grid given keyboard focus when window is selected</pre>
GetWindowFunction	<pre>(grid, #GetWindowFunction, @func, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetWindowFunction</code> to a grid to get the window function of the window that contains the grid.</p> <pre>XuiSendMessage (grid, #GetWindowFunction, @func, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #GetWindowFunction - message: get window containing the grid func - address of window function of window</pre>
GetWindowGrid	<pre>(grid, #GetWindowGrid, @windowGrid, 0, 0, 0, kid, 0)</pre> <p>Send <code>GetWindowGrid</code> to a grid to get the window grid assigned to the</p>

	<p>window that contains grid. A window grid is a grid attached to a window so that the window resizes if the grid is resized. Grid functions establish this connection when their <code>CreateWindow</code> subroutines send <code>RegisterGrid</code> messages to <code>XuiWindow()</code>.</p> <pre>XuiSendMessage (grid, #GetWindowGrid, @windowGrid, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#GetWindowGrid</code> - message: get window containing the grid <code>windowGrid</code> - grid number of grid attached to window</p>
GetWindowIcon	<pre>(grid, #GetWindowIcon, @icon, 0, @indentX, @indentY, kid, 0)</pre> <p>Send <code>GetWindowIcon</code> to a grid to get the icon number of the icon assigned to the window that contains the grid.</p> <pre>XuiSendMessage (grid, #GetWindowIcon, @icon, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#GetWindowIcon</code> - message: get icon number <code>icon</code> - icon number of window containing <code>grid,kid</code></p>
GetWindowSize	<pre>(grid, #GetWindowSize, @xDisp, @yDisp, @width, @height, kid, 0)</pre> <p>Send <code>GetWindowSize</code> to a grid to get the position and size of the window that contains the grid.</p> <pre>XuiSendMessage (grid, #GetWindowSize, @xDisp, @yDisp, @width, @height, kid, 0,)</pre> <p><code>grid,kid</code> - target grid <code>#GetWindowSize</code> - message: get position and size of window <code>xDisp,yDisp</code> - display coordinates of upper left corner of window <code>width,height</code> - width and height of window in pixels</p>
GetWindowTitle	<pre>(grid, #GetWindowTitle, @window, 0, 0, 0, kid, @title\$)</pre> <p>Send <code>GetWindowTitle</code> to a grid to get the title string displayed on the window containing the grid.</p> <pre>XuiSendMessage (grid, #GetWindowTitle, @window, 0, 0, 0, kid, @title\$)</pre> <p><code>grid,kid</code> - target grid <code>#GetWindowTitle</code> - message: get window title <code>title\$</code> - title of window containing grid</p>
GotKeyboardFocus	<pre>(grid, #GotKeyboardFocus, 0, 0, 0, 0, kid, 0)</pre> <p><code>GotKeyboardFocus</code> is sent to a grid by its window function when the grid is given keyboard focus, which means subsequent keyboard messages will be sent to the grid. Text grids normally display their text cursor when they receive <code>GotKeyboardFocus</code> and erase it when they receive <code>LostKeyboardFocus</code>.</p> <pre>XuiSendMessage (grid, #GotKeyboardFocus, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#GotKeyboardFocus</code> - message: got keyboard focus</p>
GrabTextArray	<pre>(grid, #GrabTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p>Send <code>GrabTextArray</code> to a grid to grab its text array.</p> <p>Unlike <code>GetTextArray</code>, the original text array is returned, so the grid is left with an empty text array until it is reinstated or replaced by <code>SetTextArray</code> Or <code>PokeTextArray</code>.</p> <p><code>GrabTextArray</code> is provided for cases where a <code>TextArea</code> grid contains a large amount of text and the overhead of copying the entire text array</p>

	<p>with <code>GetTextArray</code> involves excessive overhead.</p> <pre>XuiSendMessage (grid, #GrabTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p><code>grid,kid</code> - target grid <code>#GrabTextArray</code> - message: Grab text array from grid <code>text\$[]</code> - original text array stolen from grid</p>
GrabValueArray	<pre>(grid, #GrabValueArray, 0, 0, 0, 0, kid, @array[])</pre> <p>Send <code>GrabValueArray</code> to a grid to grab its <code>values[]</code> array. The original array is returned, so the grid is left with an empty <code>value[]</code> array until it is replaced by <code>PokeValueArray</code>.</p> <p><code>GrabValueArray</code> is provided for cases where a grid contains more data than is conveniently accessible with <code>GetValues</code>.</p> <pre>XuiSendMessage (grid, #GrabValueArray, 0, 0, 0, 0, kid, @array[])</pre> <p><code>grid,kid</code> - target grid <code>#GrabValueArray</code> - message: Grab value array from grid <code>array[]</code> - original <code>values[]</code> array removed from grid</p>
Help	<pre>(grid, #Help, x, y, state, time, @r0, 0)</pre> <p>The <code>Help</code> message is sent to callback functions by <code>XuiGetHelp()</code> in response to <code>GetHelp</code> messages. Normally callback functions ignore <code>Help</code> messages and let the automatic help system function normally.</p> <p>If a callback function wants to suppress help, it can set <code>r0</code> to <code>-1</code> and return. If a callback function wants to provide special purpose help, it can send <code>setHelp</code> and a help string to <code>XuiHelp()</code> for display in the <code>HelpWindow</code>, then set <code>r0=-1</code> to suppress the normal help response (which would overwrite the custom help).</p> <p>The most common use of special purpose help is to provide more selective help information. If the help request originated in a <code>TextArea</code> grid, for example, a program could learn from the <code>MouseDown</code> coordinates in the <code>x,y</code> arguments what word the user clicked on, and provide special help for that word.</p> <pre>XuiSendMessage (grid, #Help, x, y, state, time, @r0, 0)</pre> <p><code>grid</code> - target grid <code>#Help</code> - message: Help callback <code>x,y</code> - window coordinates of <code>MouseDown</code> that initiated request <code>state,time</code> - mouse information that initiated request <code>r0</code> - set to <code>-1</code> to suppress displaying of help</p>
HideWindow	<pre>(grid, #HideWindow, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>HideWindow</code> to a grid to remove the window that contains the grid from the display.</p> <pre>XuiSendMessage (grid, #HideWindow, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#HideWindow</code> - message: hide the window that contains the grid</p>
KeyDown	<pre>(grid, #KeyDown, x, y, state, time, kid, 0)</pre> <p><code>KeyDown</code> is sent to a grid by a window function when it receives a <code>WindowKeyDown</code> message and the grid has keyboard focus. The arguments tell the position of the mouse cursor at the time the key down event was detected, the key that was depressed, other keyboard</p>

	<p>status information, and the system millisecond time that the key event was detected. Most programs respond to <code>KeyDown</code> and ignore <code>KeyUp</code> messages.</p> <pre>XuiSendMessage (grid, #KeyDown, x, y, state, time, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#KeyDown</code> - message: keyboard key was pressed <code>x,y</code> - <code>x,y</code> coordinates of mouse cursor when key went down <code>state</code> - key that went down and other keyboard status information <code>time</code> - free running system millisecond time when key went down</p>
KeyUp	<pre>(grid, KeyUp, x, y, state, time, kid, 0)</pre> <p><code>KeyUp</code> is sent to a grid by a window function when it receives a <code>WindowKeyUp</code> message and the grid has keyboard focus. The arguments tell the position of the mouse cursor at the time the key up event was detected, the key that was released, other keyboard status information, and the system millisecond time that the key up event was detected. Most programs respond to <code>KeyDown</code> and ignore <code>KeyUp</code>.</p> <pre>XuiSendMessage (grid, #KeyUp, x, y, state, time, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#KeyUp</code> - message: keyboard key was released <code>x,y</code> - <code>x,y</code> coordinates of mouse cursor when key went up <code>state</code> - key that went down and other keyboard status information <code>time</code> - free running system millisecond time when key went up</p>
LostKeyboardFocus	<pre>(grid, #LostKeyboardFocus, 0, 0, 0, 0, kid, 0)</pre> <p><code>LostKeyboardFocus</code> is sent to a grid by window functions when the keyboard focus is taken from the grid and assigned to another. Subsequent keyboard messages will be sent to the new grid. Text grids normally display their text cursor when they receive a <code>GotKeyboardFocus</code> message, and erase it when they receive a <code>LostKeyboardFocus</code> message.</p> <pre>XuiSendMessage (grid, #LostKeyboardFocus, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#LostKeyboardFocus</code> - message: grid,kid lost keyboard focus</p>
MaximizeWindow	<pre>(grid, #MaximizeWindow, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>MaximizeWindow</code> to a grid to display the window that contains the grid as large as possible up to the size of the display.</p> <pre>XuiSendMessage (grid, #MaximizeWindow, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#MaximizeWindow</code> - message: maximize the window that contains grid</p>
MinimizeWindow	<pre>(grid, #MinimizeWindow, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>MinimizeWindow</code> to a grid to remove the window that contains the grid from the display, and replace it with a small icon that can be double-clicked to restore the window.</p> <pre>XuiSendMessage (grid, #MinimizeWindow, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#MinimizeWindow</code> - message: minimize the window that contains grid</p>
MonitorContext	<pre>(grid, #MonitorContext, callGrid, callFunc, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#MonitorContext</code> - message: monitor context of window <code>callGrid</code> - grid argument in <code>ContextChange</code> callback message <code>callFunc</code> - function called with <code>ContextChange</code> message</p>

	<p>reported position. If any mouse buttons are down, <code>MouseDown</code> is sent instead of <code>MouseMove</code>.</p> <pre>XuiSendMessage (grid, #MouseMove, x, y, state, time, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#MouseMove</code> - message: mouse moved with no buttons down <code>x,y</code> - coordinates of mouse cursor <code>state</code> - state of mouse buttons and keyboard <code>time</code> - system millisecond timer when move was detected</p>
MouseUp	<pre>(grid, #MouseUp, x, y, state, time, kid, 0)</pre> <p><code>MouseUp</code> is sent to a grid when a mouse button is released while the mouse cursor is within grid.</p> <pre>XuiSendMessage (grid, #MouseUp, x, y, state, time, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#MouseUp</code> - message: mouse button released in grid <code>x,y</code> - coordinates of mouse cursor <code>state</code> - state of mouse buttons and keyboard <code>time</code> - system millisecond timer when move was detected</p>
PokeTextArray	<pre>(grid, #PokeTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p>Send <code>PokeTextArray</code> to a grid to set its text array without changing the values of its text cursor.</p> <pre>XuiSendMessage (grid, #PokeTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p><code>grid,kid</code> - target grid <code>#PokeTextArray</code> - message: set text array without changing text cursor <code>text\$[]</code> - text array to set</p>
PokeValueArray	<pre>(grid, #PokeValueArray, 0, 0, 0, 0, kid, @array[])</pre> <p>Send <code>PokeValueArray</code> to install its <code>value[]</code> array. The original array is installed in the grid, so an empty <code>value[]</code> array is returned.</p> <p><code>PokeValueArray</code> is provided for cases where a grid contains more data than is accessible with <code>SetValues</code>.</p> <pre>XuiSendMessage (grid, #PokeValueArray, 0, 0, 0, 0, kid, @array[])</pre> <p><code>grid,kid</code> - target grid <code>#PokeValueArray</code> - message: Poke value array into grid <code>array[]</code> - <code>value[]</code> array to install in grid (stolen)</p>
Redraw	<pre>(grid, #Redraw, x, y, width, height, kid, 0)</pre> <p>Send <code>Redraw</code> to a grid to make it redraw itself and its kids. Grids first redraw themselves, then send redraw messages to their kids. If the redraw is in response to a <code>WindowExpose</code> message, the window coordinates of the rectangle that was exposed is supplied. Grids can return without redrawing themselves if they do not extend into the area delimited by <code>x,y,width,height</code>.</p> <p>width or height ≤ 0 means redraw is required.</p> <pre>XuiSendMessage (grid, #Redraw, x, y, width, height, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#Redraw</code> - message: redraw grid and kids <code>x,y</code> - coordinates of upper-left of exposed area <code>width,height</code> - size of exposed area (≤ 0 means always redraw)</p>
RedrawGrid	<pre>(grid, #RedrawGrid, x, y, width, height, kid, 0)</pre> <p>Send <code>RedrawGrid</code> to a grid to make it redraw itself, but not its kids. If</p>

	<p>the redraw is in response to a <code>WindowExpose</code> message, the window coordinates of the rectangle that was exposed is supplied. Grids can return without redrawing themselves if they do not extend into the area delimited by <code>x,y,width,height</code>.</p> <p><code>width <= 0</code> OR <code>height <= 0</code> means redraw is required.</p> <pre>XuiSendMessage (grid, #Redraw, x, y, width, height, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#RedrawGrid</code> - message: redraw grid, but not kids <code>x,y</code> - coordinates of upper-left of exposed area <code>width,height</code> - size of exposed area (<code><= 0</code> means always redraw)</p>
RedrawWindow	<pre>(grid, #RedrawWindow, xWin, yWin, width, height, kid, 0)</pre> <p>Send <code>RedrawWindow</code> to a grid to redraw all or part of the window that contains the grid. A <code>Redraw</code> message with the <code>xWin,yWin,width,height</code> arguments is sent to every parentless grid in the window. To force the entire window to be redrawn, set <code>width</code> or <code>height</code> to 0.</p> <pre>XuiSendMessage (grid, #RedrawWindow, xWin, yWin, width, height, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#RedrawWindow</code> - message: redraw the window that contains grid <code>xWin,yWin</code> - window coords of upper-left of rectangle to redraw <code>width,height</code> - size of rectangle to be redrawn</p>
Resize	<pre>(grid, #Resize, x, y, width, height, kid, 0)</pre> <p>Send <code>Resize</code> to a grid to change its position and/or size in its window. Most grids comply to reasonable resize requests, but some grids do not resize, or do not resize below or above certain limits. Therefore it may be necessary to check the size of a grid after a resize request.</p> <pre>XuiSendMessage (grid, #Resize, x, y, width, height, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#Resize</code> - message: resize a grid <code>x,y</code> - coords of upper-left corner of grid in parent <code>width,height</code> - size of grid</p>
ResizeNot	<pre>(grid, #Resize, x, y, width, height, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#Resize</code> - message: resize (<code>#ResizeNot</code> also acceptable) <code>x,y</code> - coords of new upper-left corner of grid <code>width,height</code> - ignored</p>
ResizeWindow	<pre>(grid, #ResizeWindow, xDisp, yDisp, width, height, kid, 0)</pre> <p>Send <code>ResizeWindow</code> to a grid to reposition and/or resize the window that contains the grid.</p> <pre>XuiSendMessage (grid, #ResizeWindow, xDisp, yDisp, width, height, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#ResizeWindow</code> - message: resize the window that contains grid <code>xDisp,yDisp</code> - display coordinates of upper-left corner of window <code>width,height</code> - size of window</p>
ResizeWindowToGrid	<pre>(grid, #ResizeWindowToGrid, 0, 0, 0, 0, kid, 0)</pre> <p><code>grid,kid</code> - target grid <code>#ResizeWindowToGrid</code> - message: resize window to fit window grid</p>
SelectWindow	<pre>(grid, #SelectWindow, 0, 0, 0, 0, kid, 0)</pre> <p>Send <code>selectWindow</code> to any grid in a window to select the window. The window is displayed and given keyboard focus.</p>

	<pre>XuiSendMessage (grid, #SelectWindow, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - target grid #SelectWindow - message: display and give keyboard focus to window</p>
Selection	<pre>(grid, #Selection, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - target grid #Selection - message: selection callback message</p>
SetBorder	<pre>(grid, #SetBorder, border, borderUp, borderDown, 0, kid, 0)</pre> <p>Send <code>setBorder</code> to a grid to set its current border style, <code>borderUp</code> style, and/or <code>borderDown</code> style. Send <code>-1</code> in any argument to leave the current value unchanged.</p> <pre>XuiSendMessage (grid, #SetBorder, border, borderUp, borderDown, 0, kid, 0)</pre> <p>grid,kid - target grid #SetBorder - message: set grid border styles and/or text indent border - the current border style borderUp - the borderUp style borderDown - the borderDown style</p>
SetCallback	<pre>(grid, #SetCallback, callGrid, callFunc, v2, v3, r0, r1)</pre> <p>Send <code>setCallback</code> to a grid to set its callback information, which includes a callback grid, callback function, and callback indices. To cancel a callback, send <code>0</code> in <code>callGrid</code> and <code>callFunc</code>.</p> <p>Whenever the grid sends a callback message, the function whose address is <code>callFunc</code> is called and passed arguments <code>callGrid,#Callback,v0,v1,v2,v3,r0,message</code>.</p> <p><code>message,v0,v1</code> are values from the original callback message, and <code>callGrid</code> is from the <code>setCallback</code> call.</p> <p><code>v2,v3,r0</code> are values from the original callback message if <code>v2,v3,r0</code> in the <code>setCallback</code> message were <code>-1</code>. Arguments that were not <code>-1</code> contain their <code>setCallback</code> value.</p> <p>Functions called with callback messages can get the original callback message arguments by calling <code>xuiCallback()</code> directly with a <code>GetCallbackArgs</code> message.</p> <pre>XuiSendMessage (grid, #SetCallback, callGrid, callFunc, v2, v3, r0, r1)</pre> <p>grid - target grid #SetCallback - message: set the callback function for grid callGrid - becomes grid argument in callback message callFunc - address of function to send callback messages to v2,v3,r0,r1 - values to send back with callback messages</p>
SetColor	<pre>(grid, #SetColor, back, draw, low, high, kid, 0)</pre> <p>Send <code>setColor</code> to a grid to set its background, drawing, highlight, and/or lowlight colors. Send <code>-1</code> in any argument to leave the current value unchanged.</p> <pre>XuiSendMessage (grid, #SetColor, back, draw, low, high, kid, 0)</pre> <p>grid,kid - target grid #SetColor - message: set main grid colors back - background color draw - drawing color low - lowlight color (3D down slope color) high - highlight color (3D up slope color)</p>

SetColorExtra	<pre>(grid, #SetColorExtra, dull, accent, lowText, highText, 0, kid, 0)</pre> <p>Send <code>SetColorExtra</code> to a grid to set its dull, accent, lowText, and highText colors. Send -1 in any argument to leave the current value unchanged.</p> <pre>XuiSendMessage (grid, #SetColorExtra, dull, accent, lowText, highText, kid, 0)</pre> <pre>grid,kid - target grid #SetColorExtra - message: set auxiliary grid colors dull - dull color (disabled features, etc...) accent - accent color (selected text, etc...) loText - text 3D texture color (darker) hiText - text 3D texture color (lighter)</pre>
SetCursor	<pre>(grid, #SetCursor, cursor, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetCursor - message: set mouse cursor number cursor - mouse cursor number</pre>
SetFont	<pre>(grid, #SetFont, size, weight, italic, angle, kid, @typeface\$)</pre> <pre>grid,kid - target grid #SetFont - message: set font to specified characteristics size - point size in .1 point units weight - boldness from 0 to 1000 (extra thin to extra heavy) italic - italic tilt (T/F) angle - angle of text from horizontal in .1 degree units typeface\$ - name of font typeface (Courier New, Times New Roman...)</pre>
SetFontNumber	<pre>(grid, #SetFontNumber, font, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetFontNumber - message: set font number property font - font number</pre>
SetGridFunction	<pre>(grid, #SetGridFunction, func, 0, 0, 0, kid, 0)</pre> <p>Send <code>setGridFunction</code> to a grid to set the address of its grid function.</p> <pre>XuiSendMessage (grid, #SetGridFunction, func, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetGridFunction - message: set grid function func - address of grid function assigned to grid,kid</pre>
SetGridFunctionName	<pre>(grid, #SetGridFunctionName, 0, 0, 0, 0, kid, @name\$)</pre> <pre>grid,kid - target grid #SetGridFunctionName - message: set grid function name property name\$ - grid function name</pre>
SetGridName	<pre>(grid, #SetGridName, 0, 0, 0, 0, kid, @gridName\$)</pre> <p>Send <code>setGridName</code> to a grid to set its grid name. If you never assign a name to a grid, a default name based on the grid type and grid number is assigned.</p> <pre>XuiSendMessage (grid, #SetGridName, 0, 0, 0, 0, kid, @gridName\$)</pre> <pre>grid,kid - target grid #SetGridName - message: set grid name gridName\$ - grid name of grid,kid</pre>
SetGridType	<pre>(grid, #SetGridType, gridType, 0, 0, 0, kid, 0)</pre> <p>Send <code>setGridType</code> to a grid to set its grid type.</p> <pre>XuiSendMessage (grid, #SetGridType, @g, gridType, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetGridFunction - message: set grid type g - grid number of grid,kid</pre>

	<pre>gridType - grid type of grid,kid</pre>
SetGridTypeName	<pre>(grid, #SetGridTypeName, 0, 0, 0, 0, kid, @name\$)</pre> <pre>grid,kid - target grid</pre> <pre>#SetGridTypeName - message: set grid type name property</pre> <pre>name\$ - grid type name</pre>
SetHelp	<pre>(grid, #SetHelp, 0, 0, 0, 0, kid, help\$)</pre> <pre>grid,kid - target grid</pre> <pre>#SetHelp - message: set string in InstantHelp window</pre> <pre>help\$ - string to display in InstantHelp window</pre>
SetHelpFile	<pre>(grid, #SetHelpFile, 0, 0, 0, 0, kid, filename\$)</pre> <pre>grid,kid - target grid (any valid grid)</pre> <pre>#SetHelpFile - message: set default help file name</pre> <pre>filename\$ - default help file name</pre>
SetHelpString	<pre>(grid, #SetHelpString, wild, 0, 0, 0, kid, @help\$)</pre> <p>Send <code>setHelpString</code> to a grid to set its current help string. This can be an empty string, "filename:entryname", ":entryname", or help text.</p> <pre>XuiSendMessage (grid, #SetHelpString, wild, 0, 0, 0, kid, @help\$)</pre> <pre>grid,kid - target grid</pre> <pre>#SetHelpString - message: set help string</pre> <pre>wild - message also sent to kids of grid,kid if wild = -1</pre> <pre>help\$ - help string</pre>
SetHelpStrings	<pre>(grid, #SetHelpStrings, 0, 0, 0, 0, kid, @help\$[])</pre> <p>Send <code>setHelpStrings</code> to a grid to set its help string and the help strings of all its kids. Each help string can be an empty string, "filename:entryname", ":entryname", or the help text itself.</p> <pre>XuiSendMessage (grid, #SetHelpStrings, 0, 0, 0, 0, kid, @help\$[])</pre> <pre>grid,kid - target grid</pre> <pre>#SetHelpStrings - message: set help string for grid and its kids</pre> <pre>help\$[] - help string for grid and its kids</pre>
SetImage	<pre>(grid, #SetImage, image, align, inX, inY, kid, image\$)</pre> <pre>grid,kid - target grid</pre> <pre>#SetImage - message: set image grid property</pre> <pre>image - optional image grid number (if image\$ is empty)</pre> <pre>align - image alignment in grid (may not be implemented)</pre> <pre>inX - x-axis indent from grid border</pre> <pre>inY - y-axis indent from grid border</pre> <pre>image\$ - filename of bitmap image</pre>
SetImageCoords	<pre>(grid, #SetImageCoords, offsetX, offsetY, width, height, kid, 0)</pre> <pre>grid,kid - target grid</pre> <pre>#SetImageCoords - message: set active area of image grid properties</pre> <pre>offsetX,offsetY - x,y offset in image grid to begin display</pre> <pre>width,height - width,height of image grid to display</pre>
SetJustify	<pre>(grid, #SetJustify, justX, justY, inX, inY, kid, 0)</pre> <p>Send <code>setJustify</code> to a grid to set its x and y text justify numbers, and/or its x and y text indent. Send <code>-1</code> in any argument to leave the current value unchanged.</p> <pre>XuiSendMessage (grid, #SetJustify, justX, justY, inX, inY, kid, 0)</pre> <pre>grid,kid - target grid</pre> <pre>#SetJustify - message: Set xy justify and/or xy text indent</pre> <pre>justX,justY - the x,y justify numbers</pre> <pre>inX,inY - x,y text indent in pixels (unless centered)</pre> <pre>width - width of the up border style</pre>

	<pre>justX,justY = 0 - left,top justify justX,justY = 1 - center,middle justify justX,justY = 2 - right,bottom justify</pre>
SetKeyboardFocus	<pre>(grid, #SetKeyboardFocus, focusGrid, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetKeyboardFocus - message: set keyboard focus focusGrid - keyboard focus grid if non-zero else target</pre>
SetKeyboardFocusGrid	<pre>(grid, #SetKeyboardFocusGrid, focusGrid, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetKeyboardFocusGrid - message: set keyboard focus grid for window focusGrid - keyboard focus grid if non-zero else target</pre>
SetMaxMinSize	<pre>(grid, #SetMaxMinSize, maxW, maxH, minW, minH, kid, 0)</pre> <p>Send <code>setMaxMinSize</code> to a grid to set its maximum and minimum width,height under any circumstances. -1 in any min/max argument means don't change the current value.</p> <pre>XuiSendMessage (grid, #SetMaxMinSize, maxW, maxH, minW, minH, kid, 0)</pre> <pre>grid,kid - target grid #SetMaxMinSize - message: set max/min width/height of grid,kid maxW,maxH - maximum possible width and height of grid,kid minW,minH - minimum possible width and height of grid,kid</pre>
SetMessageFunc	<pre>(grid, #SetMessageFunc, msg, func, 0, 0, kid, 0)</pre> <p>Send <code>setMessageFunc</code> to a grid to set the address of the function that processes message number <code>msg</code>.</p> <pre>XuiSendMessage (grid, #SetMessageFunc, msg, func, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetMessageFunc - message: set message function address for a message msg - message number to set function address for func - address of function executed in response to msg</pre>
SetMessageFuncArray	<pre>(grid, #SetMessageFuncArray, 0, 0, 0, 0, kid, @func[])</pre> <p>Send <code>setMessageFuncArray</code> to a grid to set the addresses of the functions that process messages for the grid.</p> <pre>XuiSendMessage (grid, #SetMessageFuncs, 0, 0, 0, 0, kid, @func[])</pre> <pre>grid,kid - target grid #SetMessageFuncs - message: set message function addresses func[] - addresses of message functions</pre>
SetPosition	<pre>(grid, #SetPosition, zero, low, high, highest, kid, 0)</pre> <pre>grid,kid - target grid #SetPosition - message: set scale/scroll position zero - lowest possible value, usually 0 low - low end of range (scroll-bar slider) high - high end of current range (scroll-bar slider) highest - highest possible value, usually max line #</pre>
SetSize	<pre>(grid, #SetSize, x, y, width, height, kid, 0)</pre> <p>Send <code>setSize</code> to a grid to set the position of its upper-left corner in window coordinates, and its size in pixels.</p> <pre>XuiSendMessage (grid, #SetSize, x, y, width, height, kid, 0)</pre> <pre>grid,kid - target grid #SetSize - message: set position in window and width,height x,y - upper-left corner of grid,kid in parent width,height - width and height of grid,kid in pixels</pre>
SetState	<pre>(grid, #SetState, state, keyboard, mouse, redraw, kid, 0)</pre>

	<p>Send <code>setState</code> to a grid to set its master, keyboard, mouse, and redraw disabled/enabled states. See the <code>Disable</code> and <code>Enable</code> messages for related information.</p> <pre>XuiSendMessage (grid, #SetState, state, keyboard, mouse, redraw, kid, 0)</pre> <p>grid,kid - target grid #SetState - message: set disable/enable state (FALSE / TRUE) state - master disable/enable state (keyboard,mouse,redraw) keyboard - keyboard disable/enable state mouse - mouse disable/enable state redraw - redraw disable/enable state</p>
SetStateFlags	<pre>(grid, #SetStateFlags, state, keyboard, mouse, redraw, kid, 0)</pre> <p>Send <code>setStateFlags</code> to a grid to set its master, keyboard, mouse, and redraw disabled/enabled state flags.</p> <pre>XuiSendMessage (grid, #SetState, state, keyboard, mouse, redraw, kid, 0)</pre> <p>grid,kid - target grid #SetStateFlags - message: set disable/enable state (FALSE / TRUE) state - master disable/enable state flags keyboard - keyboard disable/enable state flags mouse - mouse disable/enable state flags redraw - redraw disable/enable state flags</p>
SetStyle	<pre>(grid, #SetStyle, style, styleMax, 0, 0, kid, 0)</pre> <p>grid,kid - target grid #SetStyle - message: set style property style - style property styleMax - maximum valid value of style property</p>
SetTextArray	<pre>(grid, #SetTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p>Send <code>SetTextArray</code> to a grid to set its text array.</p> <pre>XuiSendMessage (grid, #SetTextArray, 0, 0, 0, 0, kid, @text\$[])</pre> <p>grid,kid - target grid #SetTextArray - message: set text array text\$[] - text array</p>
SetTextArrayLine	<pre>(grid, #SetTextArrayLine, line, 0, 0, @upper, kid, @text\$)</pre> <p>Send <code>SetTextArrayLine</code> to a grid to get one text string from its text array. The lower bound of the text array is 0.</p> <pre>XuiSendMessage (grid, #SetTextArrayLine, line, 0, 0, @upper, kid, @text\$)</pre> <p>grid,kid - target grid #SetTextArrayLine - message: set one line in text array line - line number to set (0 to UBOUND(text\$[])) upper - upper bound of text\$[] text\$ - text line to put in text array</p>
SetTextString	<pre>(grid, #SetTextString, 0, 0, 0, 0, kid, @text\$)</pre> <p>Send <code>setTextString</code> to a grid to set its text string.</p> <pre>XuiSendMessage (grid, #SetTextString, 0, 0, 0, 0, kid, @text\$)</pre> <p>grid,kid - target grid #SetTextString - message: set text string text\$ - text string to put in grid</p>
SetTextStrings	<pre>(grid, #SetTextStrings, 0, 0, 0, 0, kid, @text\$[])</pre> <p>Send <code>setTextStrings</code> to a grid to set text strings for the grid and its kids. If the last element in <code>text\$[]</code> is reached before the last kid, the text strings in the rest of the kids are not changed.</p> <pre>XuiSendMessage (grid, #SetTextStrings, 0, 0, 0, 0, kid, @text\$[])</pre>

	<pre>grid,kid - target grid #SetTextStrings - message: set text strings in grid and its kids text\$[] - text strings for grid and its kids</pre>
SetTexture	<pre>(grid, #SetTexture, texture, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetTexture - message: set text texture property texture - text texture (Flat/Lower/Raise/Shadow)</pre>
SetValue	<pre>(grid, #SetValue, v0, v1, v2, v3, kid, r1)</pre> <p>Send <code>setValue</code> to a grid to set the value of <code>value[r1]</code>.</p> <pre>XuiSendMessage (grid, #SetValue, v0, 0, 0, 0, kid, r1)</pre> <pre>grid,kid - target grid #SetValue - message: set value[r1] to v0 v0 - value to assign to value[r1] r1 - index into value[]</pre>
SetValues	<pre>(grid, #SetValues, v0, v1, v2, v3, kid, r1)</pre> <p>Send <code>setValues</code> to a grid to set the values of up to four elements in its <code>XLONG value[]</code> array starting with <code>value[r1]</code>.</p> <pre>XuiSendMessage (grid, #SetValues, v0, v1, v2, v3, kid, r1)</pre> <pre>grid,kid - target grid #SetValues - message: set value[r1+0] thru value[r1+3] for grid v0,v1,v2,v3, - value[r1+0] to value[r1+3] r1 - starting index (value[r1+0], value[r1+1] ...)</pre>
SetValueArray	<pre>(grid, #SetValueArray, 0, 0, 0, 0, kid, @value[])</pre> <pre>grid,kid - target grid #SetValueArray - message: set XLONG value array property value[] - XLONG value array</pre>
SetWindowFunction	<pre>(grid, #SetWindowFunction, func, 0, 0, 0, kid, 0)</pre> <p>Send <code>setWindowFunction</code> to a grid to set the address of the window function of the window that contains the grid.</p> <pre>XuiSendMessage (grid, #SetWindowFunction, func, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #SetWindowFunction - message: set window function address func - address of window function for window</pre>
SetWindowIcon	<pre>(grid, #SetWindowIcon, icon, 0, 0, 0, kid, @icon\$)</pre> <pre>grid,kid - target grid #SetWindowIcon - message: set window icon icon - icon number if icon\$ is an empty string icon\$ - name of icon</pre>
SetWindowTitle	<pre>(grid, #SetWindowTitle, @window, 0, 0, 0, kid, @title\$)</pre> <p>Send <code>setWindowTitle</code> to a grid to set the title string displayed on the window that contains the grid.</p> <pre>XuiSendMessage (grid, #SetWindowTitle, 0, 0, 0, 0, kid, @title\$)</pre> <pre>grid,kid - target grid #SetWindowTitle - message: set window title title\$ - title of window</pre>
ShowWindow	<pre>(grid, #ShowWindow, 0, 0, 0, 0, kid, 0)</pre> <pre>grid,kid - target grid #ShowWindow - message: show window but don't select it</pre>
TimeOut	<pre>(grid, #TimeOut, msTime, 0, 0, 0, kid, 0)</pre> <p>A <code>TimeOut</code> is sent to a grid by <i>GraphicsDesigner</i> when its grid timer</p>

	<p>has timed out.</p> <pre>XuiSendMessage (grid, #TimeOut, msTime, 0, 0, 0, kid, 0)</pre> <p>grid,kid - target grid #TimeOut - message: grid timer has timed out msTime - free-running system millisecond time</p>
WindowCreate	<pre>(@grid, #WindowCreate, xDisp, yDisp, width, height, winType, display\$)</pre> <p>grid - grid number of created window grid #WindowCreate - message: create window filled with a window grid xDisp,yDisp - display coords of upper-left corner of window innards width,height - width,height in pixels of window innards winType - windowType OR parentWindow display\$ - display name to contain window</p>
WindowDeselected	<pre>(window, #WindowDeselected, 0, 0, 0, 0, 0, window)</pre> <p>WindowDeselected is sent to a window by <i>GraphicsDesigner</i> after the window is deselected because another window was selected. WindowDeselected is sent before WindowSelected.</p> <pre>XuiSendMessage (window, #WindowDeselected, 0, 0, 0, 0, 0, window)</pre> <p>window - window number #WindowDeselected - message: window deselected (another was selected) window - dup of 1st argument sent by GraphicsDesigner</p>
WindowDestroy	<pre>(window, #WindowDestroy, 0, 0, 0, 0, 0, 0)</pre> <p>Send WindowDestroy to a window to destroy the window and all grids in the window. The window sends Destroyed messages to all parentless grids in the window before it destroys itself and removes itself from the display.</p> <p>The first argument must be a window number, not a grid number. DestroyWindow is a grid message that accomplishes the same thing, but takes a grid number (of any grid in the window).</p> <pre>XuiSendMessage (window, #WindowDestroy, 0, 0, 0, 0, 0, 0)</pre> <p>window - window number #WindowDestroy - message: destroy window and all grids in window</p>
WindowDestroyed	<pre>(window, #WindowDestroyed, 0, 0, 0, 0, 0, window)</pre> <p>WindowDestroyed is sent to a window by <i>GraphicsDesigner</i> when the window has been destroyed by the user or program.</p> <pre>XuiSendMessage (window, #WindowDestroyed, 0, 0, 0, 0, 0, window)</pre> <p>window - window number #WindowDestroyed - message: window has been destroyed window - duplicate of 1st argument sent by GraphicsDesigner</p>
WindowDisplay	<pre>(window, #WindowDisplay, 0, 0, 0, 0, 0, 0)</pre> <p>Send WindowDisplay to a window to display and select the window.</p> <p>The first argument must be a grid number, not a window number. DisplayWindow is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window).</p> <pre>XuiSendMessage (window, #WindowDisplay, 0, 0, 0, 0, 0, 0)</pre> <p>window - window number #WindowDisplay - message: display window</p>
WindowDisplayed	<pre>(window, #WindowDisplayed, 0, 0, 0, 0, 0, window)</pre>

	<p>WindowDisplayed is sent to a window by <i>GraphicsDesigner</i> when the window state changes from hidden or iconified to displayed, whether by the user or program.</p> <pre>XuiSendMessage (window, #WindowDisplayed, 0, 0, 0, 0, 0, window)</pre> <p> window - window number #WindowDisplayed - message: window has been displayed window - duplicate of 1st argument sent by GraphicsDesigner </p>
WindowGetDisplay	<pre>(window, #WindowGetDisplay, 0, 0, 0, 0, 0, @display\$)</pre> <p> window - window number #WindowGetDisplay - message: get display associated with window display\$ - name of display - empty string = default display </p>
WindowGetKeyboardFocusGrid	<pre>(window, #WindowGetKeyboardFocusGrid, @focusGrid, 0, 0, 0, 0, 0)</pre> <p> window - window number #WindowGetKeyboardFocusGrid - message: get window keyboard focus grid focusGrid - grid with most recent focus in window </p>
WindowGetKidArray	<pre>(window, #WindowGetKidArray, 0, 0, 0, 0, 0, @kid[])</pre> <p> window - window number #WindowGetKidArray - message: get array of kid window numbers kid[] - array of kid window numbers </p>
WindowGetSelectedWindow	<pre>(window, #WindowGetSelectedWindow, @selectedWindow, 0, 0, 0, 0, 0)</pre> <p> window - window number #WindowGetSelectedWindow - message: get selected window (0 = none) selectedWindow - currently selected window </p>
WindowGetSize	<pre>(window, #WindowGetSize, @xDisp, @yDisp, @width, @height, 0, 0)</pre> <p> window - window number #WindowGetSize - message: get display position and size of window xDisp,yDisp - x,y position of window innards on display width,height - width,height in pixels of window innards </p>
WindowGetTitle	<pre>(window, #WindowGetTitle, 0, 0, 0, 0, 0, @title\$)</pre> <p> Send WindowGetTitle to a window to get its title string. The first argument must be a window number, not a grid number. GetWindowTitle is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window). </p> <pre>XuiSendMessage (window, #WindowGetTitle, 0, 0, 0, 0, 0, @title\$)</pre> <p> window - window number #WindowGetTitle - message: get window title string title\$ - title string </p>
WindowHide	<pre>(window, #WindowHide, 0, 0, 0, 0, 0, 0)</pre> <p> Send WindowHide to a window to remove it from the display. The first argument must be a window number, not a grid number. HideWindow is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window). </p> <pre>XuiSendMessage (window, #WindowHide, 0, 0, 0, 0, 0, 0)</pre> <p> window - window number #WindowHide - message: hide the window </p>
WindowHidden	<pre>(window, #WindowHidden, 0, 0, 0, 0, 0, window)</pre> <p> WindowHidden is sent to a window by <i>GraphicsDesigner</i> when the window is hidden by the user or program. </p> <pre>XuiSendMessage (window, #WindowHidden, 0, 0, 0, 0, 0, window)</pre> <p> window - window number </p>

	<pre>#WindowHidden - message: window has been hidden window - duplicate of 1st argument sent by GraphicsDesigner</pre>
WindowKeyDown	<pre>(window, #WindowKeyDown, xWin, yWin, state, time, 0, 0) window - window number #WindowKeyDown - message: key down event in selected window xWin,yWin - window coords of current mouse cursor state - keyboard key and mouse button state information time - free-running system millisecond timer</pre>
WindowKeyUp	<pre>(window, #WindowKeyUp, xWin, yWin, state, time, 0, 0) window - window number #WindowKeyUp - message key up event in selected window xWin,yWin - window coords of current mouse cursor state - keyboard key and mouse button state information time - free-running system millisecond timer</pre>
WindowMaximize	<pre>(window, #WindowMaximize, 0, 0, 0, 0, 0, 0)</pre> <p>Send <code>WindowMaximize</code> to a window to make it as large as possible, up to the size of the display.</p> <p>The first argument must be a window number, not a grid number. <code>MaximizeWindow</code> is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window).</p> <pre>XuiSendMessage (window, #WindowMaximize, 0, 0, 0, 0, 0, 0) window - window number #WindowMaximize - message: maximize the window</pre>
WindowMaximized	<pre>(window, #WindowMaximized, 0, 0, 0, 0, 0, window)</pre> <p><code>WindowMaximized</code> is sent to a window when the window is maximized by the user.</p> <pre>XuiSendMessage (window, #WindowMaximized, 0, 0, 0, 0, 0, window) window - window number #WindowMaximized - message: window has been maximized window - duplicate of 1st argument</pre>
WindowMinimize	<pre>(window, #WindowMinimize, 0, 0, 0, 0, 0, 0)</pre> <p>Send <code>WindowMinimize</code> to a window to remove it from the display and replace it with a small icon, usually along the bottom edge of the display.</p> <p>The first argument must be a window number, not a grid number. <code>MinimizeWindow</code> is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window).</p> <pre>XuiSendMessage (window, #WindowMinimize, 0, 0, 0, 0, 0, 0) window - window number #WindowMinimize - message: minimize the window</pre>
WindowMinimized	<pre>(window, #WindowMinimized, 0, 0, 0, 0, 0, window)</pre> <p><code>WindowMinimized</code> is sent to a window when the window is minimized by the user or program.</p> <pre>XuiSendMessage (window, #WindowMinimized, 0, 0, 0, 0, 0, window) window - window number #WindowMinimized - message: window has been minimized window - duplicate of 1st argument</pre>

WindowMonitorContext	<pre>(window, #WindowMonitorContext, @callGrid, @callFunc, 0, 0, 0, 0)</pre> <p> window - window number #WindowMonitorContext - message: monitor window changes callGrid - grid argument in ContextChange callbacks callFunc - callback function </p>
WindowMonitorKeyboard	<pre>(window, #WindowMonitorKeyboard, @callGrid, @callFunc, 0, 0, 0, 0)</pre> <p> window - window number #WindowMonitorKeyboard - message: monitor keyboard events callGrid - grid argument in keyboard callbacks callFunc - callback function </p>
WindowMonitorMouse	<pre>(window, #WindowMonitorMouse, @callGrid, @callFunc, 0, 0, 0, 0)</pre> <p> window - window number #WindowMonitorMouse - message: monitor mouse events callGrid - grid argument in mouse callbacks callFunc - callback function </p>
WindowRedraw	<pre>(window, #WindowRedraw, xWin, yWin, width, height, 0, window)</pre> <p> WindowRedraw is sent to a window by <i>GraphicsDesigner</i> when all or part of the window needs to be redrawn because its was uncovered by another window, or because its was displayed from a hidden or iconified state. If <code>width <= 0</code> OR <code>height <= 0</code>, the whole window needs to be redrawn, otherwise the part of the window in the rectangle described by <code>xWin,yWin,width,height</code> needs to be redrawn. </p> <p> In response to <code>WindowRedraw</code> messages, window functions send a <code>Redraw</code> message to every parentless grid in the window, along with the <code>x,y,width,height</code> arguments. </p> <pre>XuiSendMessage (window, #WindowRedraw, xWin, yWin, width, height, 0, window)</pre> <p> window - window number #WindowRedraw - message: redraw all or part of a window window - duplicate of 1st argument sent by <i>GraphicsDesigner</i> </p>
WindowRegister	<pre>(window, #WindowRegister, winGrid, -1, width, height, @win, title\$)</pre> <p> window - window number #WindowRegister - message: register window grid, title, etc winGrid - window grid (grid to fill window) width,height - width,height in pixels of window innards win - window number returned title\$ - window title </p>
WindowResize	<pre>(window, #WindowResize, xDisp, yDisp, width, height, 0, 0)</pre> <p> Send <code>WindowResize</code> to a window to reposition it on the display and/or resize it. <code>-1</code> can be sent in <code>xDisp</code> and/or <code>yDisp</code> to accept a default window position on that axis. A value <code><=0</code> can be supplied in <code>width</code> and/or <code>height</code> to tell the window to resize to its window grid. </p> <p> The first argument must be a window number, not a grid number. <code>Resize</code> is a grid message that accomplishes the same thing if sent to a grid created by <code>CreateWindow</code>, but takes a grid number of any grid in the window. </p> <pre>XuiSendMessage (window, #WindowResize, 0, 0, 0, 0, 0, window)</pre> <p> window - window number #WindowResize - message: Resize the window xDisp,yDisp - display coords of upper-left corner of window width,height - size of window in pixels </p>
WindowResized	<pre>(window, #WindowResized, xDisp, yDisp, width, height, 0, window)</pre>

	<p><code>WindowResized</code> is sent to a <code>window</code> by <i>GraphicsDesigner</i> after its has been resized.</p> <pre>XuiSendMessage (window, #WindowResized, xDisp, yDisp, width, height, 0, window)</pre> <p><code>window</code> - window number <code>#WindowResized</code> - message: Window has been resized by user</p>
WindowSelect	<pre>(window, #WindowSelect, 0, 0, 0, 0, 0, 0)</pre> <p>Send <code>WindowSelect</code> to a <code>window</code> to display and select the window.</p> <p>The first argument must be a window number, not a grid number. <code>SelectWindow</code> is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window).</p> <pre>XuiSendMessage (window, #WindowSelect, 0, 0, 0, 0, 0, 0)</pre> <p><code>window</code> - window number <code>#WindowSelect</code> - message: display and select window</p>
WindowSelected	<pre>(window, #WindowSelected, 0, 0, 0, 0, 0, window)</pre> <pre>XuiSendMessage (window, #WindowSelected, 0, 0, 0, 0, 0, window)</pre> <p><code>window</code> - window number <code>#WindowSelected</code> - message: window has been selected <code>window</code> - duplicate of 1st argument sent by <i>GraphicsDesigner</i></p> <p><code>WindowSelected</code> is sent to a <code>window</code> by <i>GraphicsDesigner</i> when the window state changes from not selected to selected, whether the change is caused by the user or the program. The selected window is the window with keyboard focus, which means that keyboard messages are sent to that window.</p>
WindowSetDisplay	<pre>(window, #WindowSetDisplay, 0, 0, 0, 0, 0, @display\$)</pre> <p><code>window</code> - window number <code>#WindowSetDisplay</code> - message: set window display <code>display\$</code> - display name (empty string = default display)</p>
WindowSetKeyboardFocusGrid	<pre>(window, #WindowSetKeyboardFocusGrid, focusGrid, 0, 0, 0, 0, 0)</pre> <p><code>window</code> - window number <code>#WindowSetKeyboardFocusGrid</code> - message: set window keyboard focus grid <code>focusGrid</code> - keyboard focus grid for window</p>
WindowSetTitle	<pre>(window, #WindowSetTitle, 0, 0, 0, 0, 0, @title\$)</pre> <p>Send <code>WindowSetTitle</code> to a <code>window</code> to set its title string.</p> <p>The first argument must be a window number, not a grid number. <code>SetWindowTitle</code> is a grid message that accomplishes the same thing but takes a grid number (of any grid in the window).</p> <pre>XuiSendMessage (window, #WindowSetTitle, 0, 0, 0, 0, 0, @title\$)</pre> <p><code>window</code> - window number <code>#WindowSetTitle</code> - message: set window title string <code>title\$</code> - title string</p>
WindowShow	<pre>(window, #WindowShow, 0, 0, 0, 0, 0, 0)</pre> <p>Send <code>WindowShow</code> to a <code>window</code> to display the window without selecting it. If the window cannot be displayed without selecting it, it is displayed and selected.</p> <pre>XuiSendMessage (window, #WindowShow, 0, 0, 0, 0, 0, 0)</pre> <p><code>window</code> - window number</p>

```
#WindowShow - message: show window ( display without selecting )
```

Message Processing Functions

Overview

Whenever a grid function is called, it takes whatever action is appropriate for its grid type and the message it receives.

`Create` and `CreateWindow` messages are always processed within grid functions, but most other messages are simply passed on to a *message processing function* for processing. Grid functions simply ignore messages they don't recognize.

Message Function Array - func []

Every grid type has a *message function array* that contains a function address to call for every message. When a grid function receives a message, it calls the appropriate function in its message function array. If there's no function in the message function array for a given message, the grid function processes the message internally in a message processing subroutine or ignores it and returns.

Individuated Grids

All grids of a given grid type act the same, so there's only one message function array for each grid type. A program can, however, modify the behavior of a particular grid by adding or replacing one or more message function addresses for the grid.

A message function is changed by sending a `SetMessageFunc` message to the grid along with the new message function address to replace the old one. The first time a new message function is assigned to a grid the message function array from the grid type is duplicated, updated with the new entry, and assigned to the individual grid.

Henceforth the customized message function array is accessed to dispatch messages for that grid. The grid still responds to most messages like other grids of the same grid type, but it now responds differently to messages whose entries have been changed in its message function array. For example, the `#Redraw` entry in the message function array for most grid types contains the address of the standard `XuiRedraw()` message processing function. This can be changed to another function with `#SetMessageFunc`.

Not Recommended

The practice of individuating grids is not recommended, though in extreme cases it may be justified to modify the behavior of a complex grid you have no source access to. In such a case, considerable testing is in order, to assure existing functionality is not broken or disturbed in undesirable ways.

Standard Message Processing Functions

Standard message processing functions are provided with *GuiDesigner* to process common messages. Most messages for which no standard message processing function exists are special purpose messages recognized by only one or a few grid types.

Grid functions call standard message processing functions to handle most standard messages they recognize, and usually ignore the rest. When no existing message processing function handles a message in the desired manner, grid functions call a custom message processing function or a message processing subroutine within the grid function.

A description of the built-in standard message functions follows:

Standard Message Processing Functions - Details

XuiCallback	<pre>(grid, message, v0, v1, v2, v3, @r0, r1)</pre> <p>grid - grid sending the callback message (or appearing to) message - message: send this message to the callback function v0,v1,v2,v3 - variables whose meanings depend on message r0 - variable whose meaning depends on message r1 - variable whose meaning depends on message</p> <p>Call <code>XuiCallback()</code> directly to send a callback message to the callback function registered for <code>grid</code>. If no callback function is registered for <code>grid</code>, <code>XuiCallback()</code> simply returns without error.</p>
XuiCreateValueArray	<pre>(grid, #CreateValueArray, upperBound, 0, 0, 0, kid, 0)</pre> <p>grid,kid - target grid upperBound - upper bound of XLONG value array (value[])</p>
XuiDestroy	<pre>(grid, #Destroy, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - identifies target grid #Destroy - message: destroy target grid</p>
XuiDestroyWindow	<pre>(grid, #DestroyWindow, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - identifies target grid #DestroyWindow - message: destroy window containing target grid</p>
XuiDisable	<pre>(grid, #Disable, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - identifies target grid #Disable - message: disable mouse/keyboard/redraw messages</p>
XuiDisplayWindow	<pre>(grid, #DisplayWindow, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - identifies target grid #DisplayWindow - message: display window that contains target grid</p>
XuiEnable	<pre>(grid, #Enable, 0, 0, 0, 0, kid, 0)</pre> <p>grid,kid - identifies target grid #Enable - message: enable mouse/keyboard/redraw messages</p>
XuiGetAlign	<pre>(grid, #GetAlign, @align, @justify, @inLeft, @inTop, kid, @bw)</pre> <p>grid,kid - identifies target grid #GetAlign - message: get align property align - align property; 9 alignments (upper-left to lower-right) justify - justify property - left,center,right,both inLeft - indent text from left grid border inTop - indent text from top grid border</p>
XuiGetBorder	<pre>(grid, #GetBorder, @border, @borderUp, @borderDown, 0, kid, @bw)</pre> <p>grid,kid - target grid #GetBorder - message: get border properties</p>

	border - current border property borderUp - primary, normal, or only border property borderDown - secondary border property (as in button down) bw - border width in pixels calculated for know border styles
XuiGetCallback	(grid, #GetCallback, @callGrid, @callFunc, @i2, @i3, kid, 0) grid - target grid #GetCallback - message: get callback arguments callGrid - grid argument in callback messages callFunc - callback function registered by XuiSetCallback() i2,i3 - index values registered by XuiSetCallback() -1 = none kid - kid # of grid
XuiGetColor	(grid, #GetColor, @back, @draw, @lowlight, @highlight, kid, 0) grid,kid - target grid #GetColor - message: get color properties (4 of 8) back - background color draw - drawing color lowlight - lowlight color highlight - highlight color
XuiGetColorExtra	(grid, #GetColorExtra, @dull, @accent, @lowtext, @hightext, kid, 0) grid,kid - target grid #GetColorExtra - message: get color properties (4 of 8) dull - dull color accent - accent color lowtext - lowtext color hightext - hightext color
XuiGetCursor	(grid, #GetCursor, @cursor, 0, 0, 0, 0, kid, 0) grid,kid - target grid #GetCursor - message: get mouse cursor number cursor - mouse cursor number
XuiGetDisplay	(grid, #GetDisplay, 0, 0, 0, 0, 0, kid, @display\$) grid,kid - target grid #GetDisplay - message: get display screen name display\$ - display screen name
XuiGetEnclosedGrids	(grid, #GetEnclosedGrids, 0, 0, 0, 0, 0, kid, grid[]) grid,kid - target grid #GetEnclosedGrids - message: get grids surrounding target grid[] - array of grid numbers enclosed by grid,kid
XuiGetEnclosingGrid	(grid, #GetEnclosingGrid, @enclosingGrid, 0, 0, 0, 0, kid, 0) grid,kid - target grid #GetEnclosingGrid - message: get grid surrounding target grid enclosingGrid - grid number of grid enclosing grid,kid
XuiGetFont	(grid, #GetFont, @size, @weight, @italic, @angle, kid, @typeface\$) grid,kid - target grid #GetFont - message: get font information size - size of font in .1 point units weight - boldness of font from 0 to 1000 italic - italic tilt angle - angle of text baseline in .1 degree units typeface\$ - name of font typeface (Courier, Times New Roman, etc)
XuiGetFontNumber	(grid, #GetFontNumber, @font, 0, 0, 0, 0, kid, 0) grid,kid - target grid #GetFontNumber - message: grid font number font - font number
XuiGetGridFunction	(grid, #GetGridFunction, @func, 0, 0, 0, 0, kid, 0) grid,kid - target grid #GetGridFunction - message: get grid function address func - grid function address
XuiGetGridFunctionName	(grid, #GetGridFunctionName, @func, 0, 0, 0, 0, kid, gridFunc\$) grid,kid - target grid

	<pre>#GetGridFunctionName - message: get grid function address func - grid function address gridFunc\$ - grid function name</pre>
XuiGetGridName	<pre>(grid, #GetGridName, @gridNumber, 0, 0, 0, kid, @grid\$) grid,kid - target grid #GetGridName - message: get grid name gridNumber - grid number of target grid grid\$ - grid name</pre>
XuiGetGridNumber	<pre>(grid, #GetGridNumber, @g, @kid1, @kid2, @kid3, kid, @parent) grid,kid - target grid #GetGridNumber - message: get grid number g - grid number of target grid kid1,kid2,kid3 - grid numbers of kid1, kid2, kid3 parent - grid number of parent grid</pre>
XuiGetGridType	<pre>(grid, #GetGridType, @gridType, 0, 0, 0, kid, 0) grid,kid - target grid #GetGridType - message: get grid type gridType - grid type</pre>
XuiGetGridTypeName	<pre>(grid, #GetGridTypeName, @gridType, 0, 0, 0, kid, @gridType\$) grid,kid - target grid #GetGridTypeName - message: get grid type name gridType - grid type gridType\$ - grid type name</pre>
XuiGetGroup	<pre>(grid, #GetGroup, @group, 0, 0, 0, kid, 0) grid,kid - target grid #GetGroup - message: get group property group - group number</pre>
XuiGetHelp	<pre>(grid, #GetHelp, x, y, state, time, kid, @help\$) grid,kid - target grid #GetHelp - message: get help information due to help button event x,y - mouse cursor position (else < 0) state - optional mouse button state time - optional free-running millisecond system time</pre>
XuiGetHelpFile	<pre>(grid, #GetHelpFile, 0, 0, 0, 0, kid, @helpFile\$) grid,kid - target grid #GetHelpFile - message: get current help file name helpFile\$ - current help file name</pre>
XuiGetHelpString	<pre>(grid, #GetHelpString, 0, 0, 0, 0, kid, @helpString\$) grid,kid - target grid #GetHelpString - message: get help string property helpString\$ - help string property</pre>
XuiGetHelpStrings	<pre>(grid, #GetHelpStrings, 0, 0, 0, 0, kid, @helpString\$[]) grid,kid - target grid #GetHelpStrings - message: get help string of target and its kids helpString\$[] - help strings from target and its kids</pre>
XuiGetHintString	<pre>(grid, #GetHintString, 0, 0, 0, 0, kid, @hintString\$) grid,kid - target grid #GetHintString - message: get hint string hintString\$ - hint string property</pre>
XuiGetImage	<pre>(grid, #GetImage, @image, @align, @indentX, @indentY, kid, @image\$) grid,kid - target grid #GetImage - message: get image grid property and image x,y indent image - image grid align - image alignment (reserved - one of nine possibilities) indentX - indent image from border on left indentY - indent image from border on top image\$ - filename of image</pre>
XuiGetImageCoords	<pre>(grid, #GetImageCoords, @offsetX, @offsetY, @width, @height, kid, 0)</pre>

	<pre> grid,kid - target grid #GetImageCoords - message: get image coordinates (coords in image) offsetX - leftmost pixel in image to display (0 = leftmost) offsetY - topmost pixel in image to display (0 = topmost) width - width in pixels of image to display (0 = all) height - height in pixels of image to display (0 = all) </pre>
XuiGetIndent	<pre> (grid, #GetIndent, @inLeft, @inTop, @inRight, @inBottom, kid, @bw) grid,kid - target grid #GetIndent - message: get text indent properties (all in pixels) inLeft - indent text from border on left inTop - indent text from border on top inRight - indent text from border on right inBottom - indent text from border on bottom </pre>
XuiGetJustify	<pre> (grid, #GetJustify, @justify, @align, @inLeft, @inTop, kid, @bw) grid,kid - target grid #GetJustify - message: get text justify property justify - text justify property (left,center,right,both) align - text align property inLeft - text indent from border on left inTop - text indent from border on top </pre>
XuiGetKeyboardFocus	<pre> (grid, #GetKeyboardFocus, @focusGrid, 0, 0, 0, kid, 0) grid,kid - target grid #GetKeyboardFocus - message: get grid with keyboard focus focusGrid - grid that currently has keyboard focus (else 0) </pre>
XuiGetKeyboardFocusGrid	<pre> (grid, #GetKeyboardFocusGrid, @focusGrid, 0, 0, 0, kid, 0) grid,kid - target grid #GetKeyboardFocusGrid - message: get window keyboard focus grid focusGrid - most recent keyboard focus grid in window </pre>
XuiGetKidArray	<pre> (grid, #GetKidArray, @g, @parent, 0, 0, kid, @kid[]) grid,kid - target grid #GetKidArray - message: get array of kid numbers of target kids kid[] - grid numbers of target grid kids </pre>
XuiGetKidNumber	<pre> (grid, #GetKidNumber, @kidNumber, 0, 0, 0, kid, 0) grid,kid - target grid #GetKidNumber - message: get kid number of target relative to parent kidNumber - kid number of target relative to its parent grid </pre>
XuiGetKids	<pre> (grid, #GetKids, @g, @k1, @k2, @k3, kid, @k4) grid,kid - target grid #GetKids - message: get grid numbers of target and 1st 4 kids g - target grid k1,k2,k3,k4 - grid numbers of target grids first four kid grids </pre>
XuiGetMaxMinSize	<pre> (grid, #GetMaxMinSize, @maxW, @maxH, @minW, @minH, kid, 0) grid,kid - target grid #GetMaxMinSize - message: get grid max/min width/height properties maxW,maxH - maximum width and height of target minW,minH - minimum width and height of target </pre>
XuiGetMessageFunc	<pre> (grid, #GetMessageFunc, message, @func, 0, 0, kid, 0) grid,kid - target grid #GetMessageFunc - message: get a message processing function message - message the message processing function processes func - message processing function address for message </pre>
XuiGetMessageFuncArray	<pre> (grid, #GetMessageFuncArray, 0, 0, 0, 0, kid, @func[]) grid,kid - target grid #GetMessageFuncArray - message: get all message processing functions func[] - message processing function addresses </pre>
XuiGetMessageSub	<pre> (grid, #GetMessageSub, message, @sub, 0, 0, kid, 0) grid,kid - target grid #GetMessageSub - message: get message processing subroutine </pre>

	<pre> message - message the message processing subroutine processes sub - message processing subroutine address for message </pre>
XuiGetMessageSubArray	<pre> (grid, #GetMessageSubArray, 0, 0, 0, 0, kid, @sub[]) grid,kid - target grid #GetMessageSubArray - message: get message processing subroutines sub[] - message processing subroutine addresses </pre>
XuiGetModalInfo	<pre> (grid, #GetModalInfo, @v0, @v1, @v2, @v3, kid, @r1) grid,kid - target grid #GetModalInfo - message: display modal window and get response v0,v1,v2,v3,r1 - arguments from user generated callback </pre>
XuiGetMouseFocus	<pre> (grid, #GetMouseFocus, @mouseFocusGrid, 0, 0, 0, kid, 0) grid,kid - target grid #GetMouseFocus - grid number of grid with mouse focus </pre>
XuiGetParent	<pre> (grid, #GetParent, @parent, @g, @k1, @k2, kid, @k3) grid,kid - target grid #GetParent - message: get grid number of parent of target parent - grid number of parent of target grid </pre>
XuiGetRedrawFlags	<pre> (grid, #GetRedrawFlags, @redrawFlags, 0, 0, 0, kid, 0) grid,kid - target grid #GetRedrawFlags - message: get redraw flags property redrawFlags - redraw flags property </pre>
XuiGetSize	<pre> (grid, #GetSize, @x, @y, @width, @height, kid, 0) grid,kid - target grid #GetSize - message: get position and size in window coordinates x,y - coordinates of upper left corner of target width - width of grid in pixels height - height of grid in pixels </pre>
XuiGetSmallestSize	<pre> (grid, #GetSmallestSize, 0, 0, @width, @height, kid, 0) grid,kid - target grid #GetSmallestSize - message: get smallest acceptable size of grid width,height - smallest acceptable width and height in pixels </pre>
XuiGetState	<pre> (grid, #GetState, @state, @keyboard, @mouse, @redraw, kid, @enable) grid,kid - target grid #GetState - message: get state of target grid state - state property (0 means ignore keyboard,mouse,redraw) keyboard - keyboard property (0 means ignore keyboard) mouse - mouse property (0 means ignore mouse) redraw - redraw property (0 means ignore redraw) enable - enable property (0 means disabled by GraphicsDesigner) </pre>
XuiGetStyle	<pre> (grid, #GetStyle, @style, @styleMax, 0, 0, kid, 0) grid,kid - target grid #GetStyle - message: get style property style - style property styleMax - maximum valid style property </pre>
XuiGetTextArray	<pre> (grid, #GetTextArray, 0, 0, 0, 0, kid, @text\$[]) grid,kid - target grid #GetTextArray - message: get text array property text\$[] - text array property </pre>
XuiGetTextArrayLine	<pre> (grid, #GetTextArrayLine, line, 0, 0, @upper, kid, @text\$) grid,kid - target grid #GetTextArrayLine - message: get line from text array property line - line number in text array property (0 = 1st line) upper - upper bound of text array property text\$ - text string from text array property </pre>
XuiGetTextArrayLines	<pre> (grid, #GetTextArrayLines, line, count, 0, @upper, kid, @text\$[]) grid,kid - target grid #GetTextArrayLines - message: get lines from text array property </pre>

	<pre> line - first line number in text array property count - number of lines to return from text array upper - upper bound of text array property text\$[] - text lines from text array property </pre>
XuiGetTextString	<pre> (grid, #GetTextString, 0, 0, 0, 0, kid, @text\$) grid,kid - target grid #GetTextString - message: get text string property text\$ - text string property </pre>
XuiGetTextStrings	<pre> (grid, #GetTextStrings, 0, 0, 0, 0, kid, @text\$[]) grid,kid - target grid #GetTextStrings - message: get text string from target and kids text\$[] - text string property from target and kids </pre>
XuiGetTexture	<pre> (grid, #GetTexture, @texture, 0, 0, 0, kid, 0) grid,kid - target grid #GetTexture - message: get text texture property texture - text texture property </pre>
XuiGetTimer	<pre> (grid, #GetTimer, @timer, 0, 0, 0, kid, 0) grid,kid - target grid #GetTimer - message: get timer property timer - timer property (milliseconds) </pre>
XuiGetValue	<pre> (grid, #GetValue, @v0, 0, 0, 0, kid, index) grid,kid - target grid #GetValue - message: get XLONG value from value array property v0 - value[index] (element of XLONG value array property) index - element in XLONG value array property </pre>
XuiGetValues	<pre> (grid, #GetValues, @v0, @v1, @v2, @v3, kid, index) grid,kid - target grid #GetValues - message: get XLONG values from value array property v0,v1,v2,v3 - value[index+0], value[index+1], value[index+2] ... index - starting element in XLONG value array property </pre>
XuiGetValueArray	<pre> (grid, #GetValueArray, 0, 0, 0, 0, kid, @value[]) grid,kid - target grid #GetValueArray - message: get XLONG value array property value[] - copy of XLONG value array property </pre>
XuiGetWindow	<pre> (grid, #GetWindow, @window, @wtwp, @wf, @wg, kid, @kfg) grid,kid - target grid #GetWindow - message: get window containing target window - window number of window that contains target grid wtwp - (windowType OR windowParent) of window wf - window function address of window wg - window grid number (grid that fills window) kfg - most recent keyboard focus grid in window </pre>
XuiGetWindowFunction	<pre> (grid, #GetWindowFunction, @func, 0, 0, 0, kid, 0) grid,kid - target grid #GetWindowFunction - message: get window function address func - window function address </pre>
XuiGetWindowGrid	<pre> (grid, #GetWindowGrid, @g, 0, 0, 0, kid, 0) grid,kid - target grid #GetWindowGrid - message: get grid that fills window g - grid number of grid that fills window </pre>
XuiGetWindowIcon	<pre> (grid, #GetWindowIcon, @icon, 0, 0, 0, kid, @icon\$) grid,kid - target grid #GetWindowIcon - message: get icon number of minimized window icon icon - minimized window icon number icon\$ - minimized window icon name </pre>
XuiGetWindowSize	<pre> (grid, #GetWindowSize, @xDisp, @yDisp, @width, @height, kid, 0) grid,kid - target grid </pre>

	<pre>#GetWindowSize - message: get window position and size on display xDisp,yDisp - x,y position of upper left corner of window innards width,height - width,height of window innards</pre>
XuiGetWindowTitle	<pre>(grid, #GetWindowTitle, 0, 0, 0, 0, kid, @title\$) grid,kid - target grid #GetWindowTitle - message: get window title title\$ - window title</pre>
XuiGotKeyboardFocus	<pre>(grid, #GotKeyboardFocus, 0, 0, 0, 0, kid, 0) grid,kid - target grid #GotKeyboardFocus - message: target just given keyboard focus</pre>
XuiGrabArray	<pre>(grid, #GrabArray, 0, 0, 0, 0, kid, @array[]) grid,kid - target grid #GrabArray - message: grab array property array[] - array property (removed - leaves empty array)</pre>
XuiGrabTextArray	<pre>(grid, #GrabTextArray, 0, 0, 0, 0, kid, @text\$[]) grid,kid - target grid #GrabTextArray - message: grab text array property text\$[] - textArray property (removed - leaves empty array)</pre>
XuiGrabTextString	<pre>(grid, #GrabTextString, 0, 0, 0, 0, kid, @text\$) grid,kid - target grid #GrabTextString - message: grab text string property text\$ - textString property (removed - leaves empty string)</pre>
XuiGrabValueArray	<pre>(grid, #GrabValueArray, 0, 0, 0, 0, kid, @value[]) grid,kid - target grid #GrabValueArray - message: grab value array property value[] - value array property (removed - leaves empty array)</pre>
XuiHideWindow	<pre>(grid, #HideWindow, 0, 0, 0, 0, kid, 0) grid,kid - target grid #HideWindow - message: hide window containing target</pre>
XuiInitialize	<pre>(grid, #Initialize, 0, 0, 0, 0, kid, 0) grid,kid - target grid #Initialize - message: initialize</pre>
XuiKeyboardFocusBackward	<pre>(grid, #KeyboardFocusBackward, 0, 0, 0, 0, kid, 0) grid,kid - target grid #KeyboardFocusBackward - message: move keyboard focus backward</pre>
XuiKeyboardFocusForward	<pre>(grid, #KeyboardFocusForward, 0, 0, 0, 0, kid, 0) grid,kid - target grid #KeyboardFocusForward - message: move keyboard focus forward</pre>
XuiLostKeyboardFocus	<pre>(grid, #LostKeyboardFocus, 0, 0, 0, 0, kid, 0) grid,kid - target grid #LostKeyboardFocus - message: keyboard focus just removed from target</pre>
XuiMaximizeWindow	<pre>(grid, #MaximizeWindow, 0, 0, 0, 0, kid, 0) grid,kid - target grid #MaximizeWindow - message: maximize window</pre>
XuiMinimizeWindow	<pre>(grid, #MinimizeWindow, 0, 0, 0, 0, kid, 0) grid,kid - target grid #MinimizeWindow - message: minimize window</pre>
XuiMonitorContext	<pre>(grid, #MonitorContext, g, f, 0, 0, kid, action) grid,kid - target grid #MonitorContext - message: monitor window context changes g - grid argument in ContextChange callbacks f - function to call with ContextChange callbacks action - IFZ action THEN delete ELSE install</pre>

XuiMonitorHelp	(grid, #MonitorHelp, g, f, 0, 0, kid, action) grid,kid - target grid #MonitorHelp - message: monitor help requests g - grid argument in mouse help button callbacks f - function to call with help button messages action - IFZ action THEN delete ELSE install
XuiMonitorKeyboard	(grid, #MonitorKeyboard, g, f, 0, 0, kid, action) grid,kid - target grid #MonitorKeyboard - message: monitor keyboard messages g - grid argument in keyboard message callbacks f - function to call with keyboard messages
XuiMonitorMouse	(grid, #MonitorMouse, g, f, 0, 0, kid, action) grid,kid - target grid #MonitorMouse - message: monitor mouse messages g - grid argument in mouse message callbacks f - function to call with mouse messages
XuiMouseEnter	(grid, #MouseEnter, 0, 0, 0, 0, kid, 0) grid,kid - target grid #MouseEnter - message: mouse cursor entered grid
XuiPokeArray	(grid, #PokeArray, 0, 0, 0, 0, kid, @array[]) grid,kid - target grid #PokeArray - message: poke array property array[] - array property
XuiPokeTextArray	(grid, #PokeTextArray, 0, 0, 0, 0, kid, @text\$[]) grid,kid - target grid #PokeTextArray - message: poke text array property text\$[] - text array property
XuiPokeTextString	(grid, #PokeTextString, 0, 0, 0, 0, kid, @text\$) grid,kid - target grid #PokeTextString - message: poke text string property text\$ - text string property
XuiPokeValueArray	(grid, #PokeValueArray, 0, 0, 0, 0, kid, @value[]) grid,kid - target grid #PokeValueArray - message: poke XLONG value array property value[] - XLONG value array property
XuiRedraw	(grid, #Redraw, x, y, width, height, kid, 0) grid,kid - target grid #Redraw - message: redraw grid and its kids x,y - upper-left corner of grid that needs redraw width,height - width,height of window area that needs redraw
XuiRedrawGrid	(grid, #RedrawGrid, x, y, width, height, kid, 0) grid,kid - target grid #RedrawGrid - message: redraw grid (but not its kids) x,y - upper-left corner of grid that needs redraw width,height - width,height of window area that needs redraw
XuiRedrawWindow	(grid, #RedrawWindow, xWin, yWin, width, height, kid, 0) grid,kid - target grid #RedrawWindow - message: redraw grids in window xWin,yWin - upper-left corner of window area that needs redraw width,height - width,height of window area that needs redraw
XuiResize	(grid, #Resize, x, y, width, height, kid, 0) grid,kid - target grid #Resize - message: resize grid (and kids presumably) x,y - upper-left corner of grid in parent coords width,height - width,height of grid in window coords
XuiResizeNot	(grid, #ResizeNot, x, y, 0, 0, kid, 0)

	<pre> grid,kid - target grid #ResizeNot - message: don't resize grid (but maybe reposition) x,y - upper-left corner of grid in parent coords </pre>
XuiResizeWindow	<pre> (grid, #ResizeWindow, xDisp, yDisp, width, height, kid, 0) grid,kid - target grid #ResizeWindow - message: resize grid (and kids presumably) xDisp,yDisp - upper-left corner of window innards on display width,height - width,height of window innards </pre>
XuiResizeWindowToGrid	<pre> (grid, #ResizeWindowToGrid, 0, 0, 0, 0, kid, 0) grid,kid - target grid #ResizeWindowToGrid - message: resize window to window grid </pre>
XuiSelectWindow	<pre> (grid, #SelectWindow, 0, 0, 0, 0, kid, 0) grid,kid - target grid #SelectWindow - message: select window (give keyboard focus) </pre>
XuiSetAlign	<pre> (grid, #SetAlign, align, justify, inLeft, inTop, kid, 0) grid,kid - target grid #SetAlign - message: set text align,justify,indent properties align - align property (-1 = don't change) justify - justify property (-1 = don't change) inLeft - indent from border at left (-1 = don't change) inTop - indent from border at top (-1 = don't change) </pre>
XuiSetBorder	<pre> (grid, #SetBorder, border, borderUp, borderDown, 0, kid, 0) grid,kid - target grid #SetBorder - message: set border properties border - current border property (-1 = no change) borderUp - primary border property (-1 = no change) borderDown - secondary border property (-1 = no change) </pre>
XuiSetCallback	<pre> (grid, #SetCallback, callGrid, callFunc, i2, i3, kid, 0) grid,kid - target grid #SetCallback - message: set callback properties callGrid - grid argument in callback messages callFunc - address of function to call with callback messages i2,i3 - callback indices (-1 for none/pass-through) </pre>
XuiSetColor	<pre> (grid, #SetColor, back, draw, lowlight, highlight, kid, 0) grid,kid - target grid #SetColor - message: set color properties (4 of 8) back - background color (-1 = no change) draw - drawing color (-1 = no change) lowlight - lowlight color (-1 = no change) highlight - highlight color (-1 = no change) </pre>
XuiSetColorAll	<pre> (grid, #SetColor, 0, 0, 0, 0, kid, 0) grid,kid - target grid #SetColor - message: set color to target and all its kids back - background color (-1 = no change) draw - drawing color (-1 = no change) lowlight - lowlight color (-1 = no change) highlight - highlight color (-1 = no change) </pre>
XuiSetColorExtra	<pre> (grid, #SetColorExtra, dull, accent, lowtext, hightext, kid, 0) grid,kid - target grid #SetColorExtra - message: set color properties (4 of 8) dull - dull color property (-1 = no change) accent - accent color property (-1 = no change) lowtext - lowtext color property (-1 = no change) hightext - hightext color property (-1 = no change) </pre>
XuiSetColorExtraAll	<pre> (grid, #SetColorExtra, dull, accent, lowtext, hightext, kid, 0) grid,kid - target grid #SetColorExtra - message: set color to target and its kids dull - dull color property (-1 = no change) accent - accent color property (-1 = no change) lowtext - lowtext color property (-1 = no change) </pre>

	<p>hightext - hightext color property (-1 = no change)</p>
XuiSetCursor	<p>(grid, #SetCursor, cursor, 0, 0, 0, kid, 0)</p> <p>grid,kid - target grid #SetCursor - message: set cursor number cursor - mouse cursor number property</p>
XuiSetFont	<p>(grid, #SetFont, size, weight, italic, angle, kid, @typeface\$)</p> <p>grid,kid - target grid #SetFont - message: set font property size - size of font in .1 point units weight - boldness of font (0 to 1000) italic - italic tilt of font (true/false) angle - tilt of font baseline in .1 degree units (0 - 360) typeface\$ - name of typeface (Courier New, Times New Roman, etc)</p>
XuiSetFontNumber	<p>(grid, #SetFontNumber, font, 0, 0, 0, kid, 0)</p> <p>grid,kid - target grid #SetFontNumber - message: set font number property font - font number</p>
XuiSetGridFunction	<p>(grid, #SetGridFunction, func, 0, 0, 0, kid, 0)</p> <p>grid,kid - target grid #SetGridFunction - message: set grid function of target grid func - grid function address</p>
XuiSetGridFunctionName	<p>(grid, #SetGridFunctionName, 0, 0, 0, 0, kid, @gridFunc\$)</p> <p>grid,kid - target grid #SetGridFunctionName - message: set grid function name gridFunc\$ - grid function name property</p>
XuiSetGridName	<p>(grid, #SetGridName, 0, 0, 0, 0, kid, @grid\$)</p> <p>grid,kid - target grid #SetGridName - message: set grid name property grid\$ - grid name property (valid symbol name)</p>
XuiSetGridType	<p>(grid, #SetGridType, gridType, 0, 0, 0, kid, 0)</p> <p>grid,kid - target grid #SetGridType - message: set grid type property gridType - grid type property</p>
XuiSetGridTypeName	<p>(grid, #SetGridTypeName, 0, 0, 0, 0, kid, @gridType\$)</p> <p>grid,kid - target grid #SetGridTypeName - message: set grid type name gridType\$ - grid type name</p>
XuiSetGroup	<p>(grid, #SetGroup, group, 0, 0, 0, kid, 0)</p> <p>grid,kid - target grid #SetGroup - message: set group property</p>
XuiSetHelp	<p>(grid, #SetHelp, 0, 0, 0, 0, kid, @help\$)</p> <p>grid,kid - target grid #SetHelp - message: display string in InstantHelp window help\$ - string to display in InstantHelp window</p>
XuiSetHelpFile	<p>(grid, #SetHelpFile, 0, 0, 0, 0, kid, @helpFile\$)</p> <p>grid,kid - target grid #SetHelpFile - message: set current help file name (for program) helpFile\$ - name of active help file</p>
XuiSetHelpString	<p>(grid, #SetHelpString, 0, 0, 0, 0, kid, @help\$)</p> <p>grid,kid - target grid #SetHelpString - message: set help string property help\$ - help string property</p>
XuiSetHelpStrings	<p>(grid, #SetHelpStrings, 0, 0, 0, 0, kid, @help\$[])</p> <p>grid,kid - target grid #SetHelpStrings - message: set help string of target and kids</p>

	<pre>help\$[] - help string property for target and kids</pre>
XuiSetHintString	<pre>(grid, #SetHintString, 0, 0, 0, 0, kid, @hint\$) grid,kid - target grid #SetHintString - message: set hint string property hint\$ - hint string property</pre>
XuiSetImage	<pre>(grid, #SetImage, image, imageAlign, inX, inY, kid, @image\$) grid,kid - target grid #SetImage - message: set image properties image - grid number of image grid (alternate to image\$) imageAlign - image alignment property (reserved) inX,inY - indent image from left/top border of target</pre>
XuiSetImageCoords	<pre>(grid, #SetImageCoords, startX, startY, width, height, kid, 0) grid,kid - target grid #SetImageCoords - message: set image coordinates startX,startY - start displaying image x,y pixels from left/top width,height - width,height of image to display in pixels</pre>
XuiSetIndent	<pre>(grid, #SetIndent, inLeft, inTop, inRight, inBottom, kid, 0) grid,kid - target grid #SetIndent - message: set text indent properties inLeft - indent text from left border (-1 = no change) inTop - indent text from top border (-1 = no change) inRight - indent text from right border (-1 = no change) inBottom - indent text from bottom border (-1 = no change)</pre>
XuiSetJustify	<pre>(grid, #SetJustify, justify, 0, 0, 0, kid, 0) grid,kid - target grid #SetJustify - message: set text justify property justify - text justify property (-1 = no change)</pre>
XuiSetKeyboardFocus	<pre>(grid, #SetKeyboardFocus, g, 0, 0, 0, kid, 0) grid,kid - target grid #SetKeyboardFocus - message: set keyboard focus g - grid to receive keyboard focus (0 for target)</pre>
XuiSetKeyboardFocusGrid	<pre>(grid, #SetKeyboardFocusGrid, g, 0, 0, 0, kid, 0) grid,kid - target grid #SetKeyboardFocusGrid - message: set window keyboard focus grid g - grid to receive keyboard focus in window (0 = target)</pre>
XuiSetMaxMinSize	<pre>(grid, #SetMaxMinSize, maxW, maxH, minW, minH, kid, 0) grid,kid - target grid #SetMaxMinSize - message: set max/min size properties maxW,maxH - maximum width,height of target (-1 = no change) minW,minH - minimum width,height of target (-1 = no change)</pre>
XuiSetMessageFunc	<pre>(grid, #SetMessageFunc, message, func, 0, 0, kid, 0) grid,kid - target grid #SetMessageFunc - message: set message processing function message - set message processing function for this message func - message processing function address</pre>
XuiSetMessageFuncArray	<pre>(grid, #SetMessageFuncArray, 0, 0, 0, 0, kid, @func[]) grid,kid - target grid #SetMessageFuncArray - message: set message processing function array func[] - array of message processing function addresses</pre>
XuiSetMessageSub	<pre>(grid, #SetMessageSub, message, sub, 0, 0, kid, 0) grid,kid - target grid #SetMessageSub - message: set message processing subroutine message - set message processing subroutine for this message sub - message processing subroutine address</pre>
XuiSetMessageSubArray	<pre>(grid, #SetMessageSubArray, 0, 0, 0, 0, kid, @sub[]) grid,kid - target grid</pre>

	<pre>#SetMessageSubArray - message: set message processing subroutines sub[] - message processing subroutine array</pre>
XuiSetMouseFocus	<pre>(grid, #SetMouseFocus, g, 0, 0, 0, kid, 0) grid,kid - target grid #SetMouseFocus - message: set mouse focus to target g - grid to receive mouse focus (target if g = 0)</pre>
XuiSetSize	<pre>(grid, #SetSize, x, y, width, height, kid, 0) grid,kid - target grid #SetSize - message: set position and size of target x,y - coordinates of upper-left corner of target width,height - width,height of target in pixels</pre>
XuiSetRedrawFlags	<pre>(grid, #SetRedrawFlags, redrawFlags, 0, 0, 0, kid, 0) grid,kid - target grid #SetRedrawFlags - message: set redraw flags property redrawFlags - redraw flags property</pre>
XuiSetState	<pre>(grid, #SetState, state, keyboard, mouse, redraw, kid, 0) grid,kid - target grid #SetState - message: set state properties state - state property keyboard - keyboard property mouse - mouse property redraw - redraw property</pre>
XuiSetStyle	<pre>(grid, #SetStyle, style, styleMax, 0, 0, kid, 0) grid,kid - target grid #SetStyle - message: set style properties style - style property styleMax - maximum valid style property</pre>
XuiSetTextArray	<pre>(grid, #SetTextArray, 0, 0, 0, 0, kid, @text\$[]) grid,kid - target grid #SetTextArray - message: set text array property text\$[] - text array property</pre>
XuiSetTextArrayLine	<pre>(grid, #SetTextArrayLine, line, 0, 0, 0, kid, @text\$) grid,kid - target grid #SetTextArrayLine - message: set text line in text array property line - line to replace in text array property text\$ - text to replace text array line</pre>
XuiSetTextArrayLines	<pre>(grid, #SetTextArrayLines, line, count, first, @up, kid, @text\$[]) grid,kid - target grid #SetTextArrayLines - message: replace lines in text array property line - lowest line to replace (0 relative) count - number of lines to replace first - first source text line in text\$[] (0 relative) up - upper bound of text array property (result) text\$[] - source text lines</pre>
XuiSetTextString	<pre>(grid, #SetTextString, 0, 0, 0, 0, kid, @text\$) grid,kid - target grid #SetTextString - message: set text string property text\$ - text string property</pre>
XuiSetTextStrings	<pre>(grid, #SetTextStrings, 0, 0, 0, 0, kid, @text\$[]) grid,kid - target grid #SetTextStrings - message: set text string to target and kids text\$[] - text strings to set to target and kids</pre>
XuiSetTexture	<pre>(grid, #SetTexture, texture, 0, 0, 0, kid, 0) grid,kid - target grid #SetTexture - message: set text texture property texture - text texture property</pre>
XuiSetTimer	<pre>(grid, #SetTimer, timer, 0, 0, 0, kid, 0)</pre>

	<pre> grid,kid - target grid #SetTimer - message: set grid timer timer - milliseconds </pre>
XuiSetValue	<pre> (grid, #SetValue, v0, 0, 0, 0, kid, index) grid,kid - target grid #SetValue - message: set element of XLONG value array v0 - value to put in value[index] index - element in value array </pre>
XuiSetValues	<pre> (grid, #SetValues, v0, v1, v2, v3, kid, index) grid,kid - target grid #SetValues - message: set elements of XLONG value array v0,v1,v2,v3 - values to assign to value[index+0] - value[index+3] index - starting element in XLONG value array </pre>
XuiSetValueArray	<pre> (grid, #SetValueArray, 0, 0, 0, 0, kid, @value[]) grid,kid - target grid #SetValueArray - message: set XLONG value array value[] - value array property (copied) </pre>
XuiSetWindowFunction	<pre> (grid, #SetWindowFunction, func, 0, 0, 0, kid, 0) grid,kid - target grid #SetWindowFunction - message: set window function address func - window function address </pre>
XuiSetWindowIcon	<pre> (grid, #SetWindowIcon, icon, 0, 0, 0, kid, @icon\$) grid,kid - target grid #SetWindowIcon - message: set window icon icon - icon number (if icon\$ is an empty string) icon\$ - icon name (if empty string, see icon argument) </pre>
XuiSetWindowTitle	<pre> (grid, #SetWindowTitle, 0, 0, 0, 0, kid, @title\$) grid,kid - target grid #SetWindowTitle - message: set window title title\$ - window title </pre>
XuiShowWindow	<pre> (grid, #ShowWindow, 0, 0, 0, 0, kid, 0) grid,kid - target grid #ShowWindow - message: show window without selecting </pre>
XuiStartTimer	<pre> (grid, #StartTimer, msec, 0, 0, 0, kid, 0) grid,kid - target grid #StartTimer - message: start grid timer msec - milliseconds (if msec <= 0, msec = timer property) </pre>

Conversion Functions

XuiAlignNameToNumber	(@name\$, @number) name\$ - align property name number - align property number
XuiBorderNameToNumber	(@name\$, @number) name\$ - border property name number - border property number
XuiCanNameToNumber	(@name\$, @number) name\$ - can property name number - can property number
XuiCanStringToNumber	(@name\$, @number) name\$ - can property string number - can property number
XuiJustifyNameToNumber	(@name\$, @number) name\$ - justify property name number - justify property number
XuiTextureNameToNumber	(@name\$, @number) name\$ - texture property name number - texture property number

XuiAlignNumberToName	(number, @name\$) number - align property number name\$ - align property name
XuiBorderNumberToName	(number, @name\$) number - border property number name\$ - border property name
XuiCanNumberToName	(number, @name\$) number - can property number name\$ - can property name
XuiCanNumberToString	(number, @name\$) number - can property number name\$ - can property string
XuiJustifyNumberToName	(number, @name\$) number - justify property number name\$ - justify property name
XuiTextureNumberToName	(number, @name\$) number - align property number name\$ - align property name

XuiBorderNumberToWidth	(number, @width) number - border property number width - width of border in pixels
XuiGridFuncNameToAddr	(@func\$, @addr) func\$ - grid function name addr - grid function address
XuiGridTypeToGridFunc	(gridType, @gridFunc) gridType - grid type number gridFunc - grid func address

Support Functions

Xui	()
XuiCreateGrid	(@grid, gridType, x, y, width, height, window, parent, func) grid - grid number of created grid gridType - grid type of grid to create x,y - coords to position upper left corner of grid in parent width,height - width and height in pixels of grid to create window - window to create grid in parent - parent grid (0 = no parent = window grid) func - grid function address
XuiGetDefaultColors	(@back, @draw, @low, @high, @dull, @accent, @lowtext, @hightext) back - default background color draw - default drawing color low - default lowlight color high - default highlight color dull - default dull color accent - default accent color lowtext - default lowtext color hightext - default hightext color
XuiGetDefaultCursor	(@cursor) cursor - default mouse cursor
XuiGetDefaultMessageFuncArray	(@func[]) func[] - array of default message processing functions
XuiGetGridRegion	(grid, x, y, @region, @cursor, @defaultCursor) grid - grid number x,y - coordinates of test point region - region of grid cursor - nominal resize cursor number for region defaultCursor - default cursor
XuiGetGridTypeMessageFuncArray	(gridType, @func[]) gridType - grid type func[] - message processing functions for grid type
XuiGetGridTypeMessageSubArray	(gridType, SUBADDR @sub[]) gridType - grid type sub[] - message processing subroutines for grid type

XuiGetGridTypeValue	(gridType, @property\$, @ANY) gridType - grid type property\$ - property name ("align", "border", "textString", etc) ANY - XLONG or STRING value or array (per property\$)
XuiGetGridValue	(grid, @property\$, @ANY) grid - grid number property\$ - property name ANY - XLONG or STRING value or array (per property\$)
XuiGetHelpWindowGrid	(@window, @grid) window - InstantHelp window number grid - InstantHelp grid number
XuiGetTextArraySize	(@text\$[], font, @w, @h, @width, @height, extraX, extraY) text\$[] - text array font - font number w,h - width,height in pixels of text if horizontal width,height - width,height in pixels of text at font angle extraX,extraY - add extra pixels in X,Y for each line (texture, etc)
XuiGetTextSelectionGrid	(@grid) grid - text selection grid
XuiGridKid	(grid, checkGrid) grid - grid number checkGrid - check grid number (is this a kid of grid)
XuiGridOverlapsWin	(grid, @overlap, xWin, yWin, width, height) grid - grid number overlap - true if grid and rectangle overlap xWin,yWin - window coords of upper-left corner of rectangle width,height - width,height of rectangle
XuiPassOn	(grid, message, @v0, @v1, @v2, @v3, r0, @ANY) grid - grid number message - message number v0,v1,v2,v3 - arguments depend on message r0,ANY - arguments depend on message
XuiPlaceWindow	(window, mode, @xDisp, @yDisp, @width, @height) window - window number mode - placement mode xDisp,yDisp - upper-left corner of window innards in display coords width,height - width,height in pixels of window innards
XuiPositionGrid	(grid, @x, @y, @width, @height) grid - grid number x,y - coords of upper-left corner of grid in parent width,height - width,height in pixels of grid
XuiProcessMessage	(grid, message, @v0, @v1, @v2, @v3, r0, @ANY, gridType) grid - grid number message - message number v0,v1,v2,v3 - arguments depend on message r0,ANY - arguments depend on message gridType - grid type of grid function

XuiRegisterGridType	(@gridType, @gridType\$, gridFunc, @messFunc[], SUBADDR @messSub[]) gridType - grid type number gridType\$ - grid type name gridFunc - grid function address messFunc[] - message processing function addresses messSub[] - message processing subroutine addresses
XuiSendMessage	(grid, message, @v0, @v1, @v2, @v3, r0, @ANY) grid - grid number message - message number v0,v1,v2,v3 - arguments depend on message r0,ANY - arguments depend on message
XuiSendToKid	(grid, message, @v0, @v1, @v2, @v3, r0, @ANY) grid - grid number message - message number v0,v1,v2,v3 - arguments depend on message r0,ANY - arguments depend on message
XuiSetDefaultColors	(back, draw, low, high, dull, accent, lowtext, hightext) back - default background color draw - default drawing color low - default lowlight color high - default highlight color dull - default dull color accent - default accent color lowtext - default lowtext color hightext - default hightext color
XuiSetDefaultCursor	(cursor) cursor - cursor number
XuiSetGridTypeValue	(gridType, @property\$, ANY) gridType - grid type number property\$ - property name ANY - XLONG or STRING value or array (per property name)
XuiSetGridValue	(grid, @property\$, ANY) grid - grid number property\$ - property name ANY - XLONG or STRING value or array (per property name)
XuiSetTextSelectionGrid	(grid) grid - grid number
XuiVersion\$	() return value - version of GuiDesigner as string, as in "3.1204"
XuiWindow	(wingrid, message, @v0, @v1, @v2, @v3, r0, @ANY) wingrid - window number or grid number (per message) message - message number v0,v1,v2,v3 - arguments depend on message r0,ANY - arguments depend on message

Convenience Functions

XuiCreateWindow	<p>(@grid, gridType\$, xDisp, yDisp, width, height, winType, display\$)</p> <p>grid - grid created inside window gridType\$ - name of grid type to put inside created window xDisp,yDisp - x,y position of window in display coords width,height - width,height in pixels of window winType - window type OR parent window number display\$ - display name to contain window</p> <p>If a zero is passed in winType, a default window type window is created, and its window type is returned in winType.</p>
XuiDialog	<p>(@label\$, @default\$, @kid, @reply\$)</p> <p>label\$ - text to display on XuiDialog2B label grid default\$ - default response (put in XuiTextLine grid) kid - selected kid reply\$ - final contents of XuiTextLine grid</p>
XuiGetReply	<p>(grid, @title\$, @label\$, @grids\$, @v0, @v1, @kid, @reply\$)</p> <p>grid - grid number title\$ - window title label\$ - text to display on first label kid grids\$ - text to display on subsequent grids v0,v1 - callback arguments kid - selected kid reply\$ - text from inputTextString or inputTextArray grid</p>
XuiGetResponse	<p>(@gridType\$, @title\$, @label\$, @grids\$, @v0, @v1, @kid, @reply\$)</p> <p>gridType\$ - grid type name title\$ - window title label\$ - text to display on first label kid grids\$ - text to display on subsequent grids v0,v1 - callback arguments kid - selected kid reply\$ - text from inputTextString or inputTextArray grid</p>
XuiMessage	<p>(@label\$)</p> <p>label\$ - text to display on XuiMessage1B label grid</p>
XuiSendStringMessage	<p>(grid, message\$, v0, v1, v2, v3, kid, ANY)</p> <p>grid - grid number message\$ - message name v0,v1,v2,v3 - arguments depend on message\$ kid - kid # of target grid,kid ANY - argument depends on message\$</p>
XuiQueueCallbacks	<p>(grid, message, v0, v1, v2, v3, kid, r1)</p> <p>grid - grid number message - message number v0,v1,v2,v3 - arguments depend on message kid - kid # in callback message r1 - argument depends on message</p>
XuiGetNextCallback	<p>(@grid, @message\$, @v0, @v1, @v2, @v3, @kid, @r1\$)</p> <p>grid - grid number message\$ - message name string v0,v1,v2,v3 - arguments depend on message\$ kid - kid # in callback message r1\$ - grid name of grid that initiated the callback</p>

#Callback, 17
 #Create, 18
 #CreateWindow, 18
 #Destroy, 19
 #Destroyed, 19
 #DestroyWindow, 19
 #Disable, 19
 #DisplayWindow, 19
 #Enable, 20
 #GetBorder, 20
 #GetCallback, 20
 #GetCallbackArgs, 20
 #GetColor, 21
 #GetColorExtra, 21
 #GetCursor, 21
 #GetDisplay, 21
 #GetEnclosedGrids, 21
 #GetEnclosingGrid, 21
 #GetFont, 22
 #GetFontNumber, 22
 #GetGridFunction, 22
 #GetGridFunctionName, 22
 #GetGridName, 22
 #GetGridType, 23
 #GetGridTypeName, 23
 #GetHelp, 23
 #GetHelpFile, 23
 #GetHelpString, 24
 #GetHelpStrings, 24
 #GetImage, 24
 #GetImageCoords, 24
 #GetJustify, 24
 #GetKeyboardFocus, 25
 #GetKeyboardFocusGrid, 25
 #GetKidArray, 25
 #GetKids, 25
 #GetMaxMinSize, 25
 #GetMessageFunc, 25
 #GetMessageFuncArray, 26
 #GetMessageSub, 26
 #GetMessageSubArray, 26
 #GetModallInfo, 26
 #GetParent, 26
 #GetSize, 26
 #GetSmallestSize, 26
 #GetState, 27
 #GetStateFlags, 27
 #GetStyle, 27
 #GetTextArray, 27
 #GetTextArrayLine, 27
 #GetTextArrayLines, 27
 #GetTextString, 28
 #GetTextStrings, 28
 #GetTexture, 28
 #GetValue, 28
 #GetValueArray, 28
 #GetValues, 28
 #GetWindow, 28
 #GetWindowFunction, 29
 #GetWindowGrid, 29
 #GetWindowIcon, 29
 #GetWindowSize, 29
 #GetWindowTitle, 29
 #GotKeyboardFocus, 30
 #GrabTextArray, 30
 #GrabValueArray, 30
 #Help, 31
 #HideWindow, 31
 #KeyDown, 31
 #KeyUp, 32
 #LostKeyboardFocus, 32
 #MaximizeWindow, 32
 #MinimizeWindow, 32
 #MonitorContext, 32
 #MonitorKeyboard, 32
 #MonitorMouse, 33
 #MouseDown, 33
 #MouseDown, 33
 #MouseDrag, 33
 #MouseEnter, 33
 #MouseExit, 33
 #MouseMove, 34
 #MouseUp, 34
 #PokeTextArray, 34
 #PokeValueArray, 34
 #Redraw, 35
 #RedrawGrid, 35
 #RedrawWindow, 35
 #Resize, 35, 36
 #ResizeNot, 36
 #ResizeWindow, 36
 #ResizeWindowToGrid, 36
 #Selection, 36
 #SelectWindow, 36
 #SetBorder, 36
 #SetCallback, 37
 #SetColor, 37
 #SetColorExtra, 37
 #SetCursor, 37
 #SetFont, 38
 #SetFontNumber, 38
 #SetGridFunction, 38
 #SetGridFunctionName, 38
 #SetGridName, 38

#SetGridTypeName, 38
 #SetHelp, 38
 #SetHelpFile, 38
 #SetHelpString, 39
 #SetHelpStrings, 39
 #SetImage, 39
 #SetImageCoords, 39
 #SetJustify, 39
 #SetKeyboardFocus, 39
 #SetKeyboardFocusGrid, 39
 #SetMaxMinSize, 40
 #SetMessageFunc, 40
 #SetMessageFuncs, 40
 #SetPosition, 40
 #SetSize, 40
 #SetState, 40
 #SetStateFlags, 41
 #SetStyle, 41
 #SetTextArray, 41
 #SetTextArrayLine, 41
 #SetTextString, 41
 #SetTextStrings, 41
 #SetTexture, 41
 #SetValue, 42
 #SetValueArray, 42
 #SetValues, 42
 #SetWindowFunction, 42
 #SetWindowIcon, 42
 #SetWindowTitle, 42
 #ShowWindow, 42
 #Timeout, 42
 #WindowCreate, 43
 #WindowDeselected, 43
 #WindowDestroy, 43
 #WindowDestroyed, 43
 #WindowDisplay, 43
 #WindowDisplayed, 44
 #WindowGetDisplay, 44
 #WindowGetKeyboardFocusGrid, 44
 #WindowGetKidArray, 44
 #WindowGetSelectedWindow, 44
 #WindowGetSize, 44
 #WindowGetTitle, 44
 #WindowHidden, 45
 #WindowHide, 44
 #WindowKeyDown, 45
 #WindowKeyUp, 45
 #WindowMaximize, 45
 #WindowMaximized, 45
 #WindowMinimize, 45
 #WindowMinimized, 46
 #WindowMonitorContext, 46
 #WindowMonitorKeyboard, 46
 #WindowMonitorMouse, 46
 #WindowRedraw, 46
 #WindowRegister, 46
 #WindowResize, 47
 #WindowResized, 47
 #WindowSelect, 47
 #WindowSelected, 47
 #WindowSetDisplay, 47
 #WindowSetKeyboardFocusGrid, 48
 #WindowSetTitle, 48
 #WindowShow, 48

 abbreviations, 1
 accentColor, 3, 4, 9
 align, 3, 5, 10
 array, 2, 3, 5
 attribute, 1

 backgroundColor, 3, 4, 9
 border, 3, 5, 12
 borderDown, 3, 5, 12
 borderUp, 3, 5, 12
 buffer, 3, 4, 8

 callbackFunc, 3, 4
 callbackGrid, 3, 4, 7
 can, 3, 5, 13
 Create, 4
 CreateValueArray, 2
 CreateWindow, 4
 cursor, 3, 4, 8

 drawingColor, 3, 4, 9
 dullColor, 3, 4, 9

 f XuiCreateGrid(), 6
 f XuiGetAlign(), 10, 11
 f XuiGetBorder(), 12
 f XuiGetBuffer(), 8
 f XuiGetCallback(), 7
 f XuiGetCan(), 13
 f XuiGetColor(), 9
 f XuiGetColorExtra(), 9
 f XuiGetCursor(), 8
 f XuiGetDefaultColors(), 9
 f XuiGetFont(), 12
 f XuiGetFontNumber(), 12
 f XuiGetGridFunction(), 7
 f XuiGetGridFunctionName(), 7
 f XuiGetGridName(), 6
 f XuiGetGridNumber(), 6

f XuiGetGridType(), 6
 f XuiGetGridTypeName(), 6
 f XuiGetGroup(), 13
 f XuiGetHelpString(), 14
 f XuiGetHintString(), 14
 f XuiGetImage(), 8
 f XuiGetImageCoords(), 8
 f XuiGetIndent(), 11
 f XuiGetJustify(), 10, 11
 f XuiGetKidArray(), 14
 f XuiGetKids(), 14
 f XuiGetMessageFunc(), 9, 14
 f XuiGetMessageFuncArray(), 14
 f XuiGetMessageSub(), 14
 f XuiGetMessageSubArray(), 14
 f XuiGetParent(), 7
 f XuiGetRedrawFlags(), 10
 f XuiGetSize(), 8
 f XuiGetState(), 13
 f XuiGetStyle(), 13
 f XuiGetTextArray(), 15
 f XuiGetTextString(), 15
 f XuiGetTexture(), 11
 f XuiGetTimer(), 10
 f XuiGetValue(), 15
 f XuiGetValueArray(), 15
 f XuiGetValues(), 15
 f XuiGetWindow(), 7
 f XuiGrabTextArray(), 15
 f XuiGrabValueArray(), 15
 f XuiKillTimer(), 10
 f XuiPokeTextArray(), 15
 f XuiPokeValueArray(), 15
 f XuiRegisterGridType(), 6, 7
 f XuiResize(), 8
 f XuiSetAlign(), 10, 11
 f XuiSetBorder(), 12
 f XuiSetBuffer(), 8
 f XuiSetCallback(), 7
 f XuiSetCan(), 13
 f XuiSetColor(), 9
 f XuiSetColorAll(), 9
 f XuiSetColorExtra(), 9
 f XuiSetColorExtraAll(), 9
 f XuiSetCursor(), 8
 f XuiSetDefaultColors(), 9
 f XuiSetFont(), 12
 f XuiSetFontNumber(), 12
 f XuiSetGridFunction(), 7
 f XuiSetGridFunctionName(), 7
 f XuiSetGridName(), 6
 f XuiSetGridType(), 6
 f XuiSetGridTypeName(), 6
 f XuiSetGroup(), 13
 f XuiSetHelpString(), 14
 f XuiSetHintString(), 14
 f XuiSetImage(), 8
 f XuiSetImageCoords(), 8
 f XuiSetIndent(), 11
 f XuiSetJustify(), 10
 f XuiSetMessageFunc(), 9, 14
 f XuiSetMessageFuncArray(), 14
 f XuiSetMessageSub(), 14
 f XuiSetMessageSubArray(), 14
 f XuiSetRedrawFlags(), 10
 f XuiSetSize(), 8
 f XuiSetState(), 13
 f XuiSetStyle(), 13
 f XuiSetTextArray(), 15
 f XuiSetTextString(), 15
 f XuiSetTexture(), 11
 f XuiSetTimer(), 10
 f XuiSetValue(), 15
 f XuiSetValueArray(), 15
 f XuiSetValues(), 15
 f XuiStartTimer(), 10
 focusKid, 3, 5, 13
 font, 3, 5
 font\$, 5
 fontAngle, 5, 12
 fontItalic, 3, 5, 12
 fontName, 12
 fontNumber, 3, 12
 fontSize, 3, 5, 12
 fontWeight, 3, 5, 12
 general purpose grid properties, 1, 2
 Get, 2
 GetAlign, 5
 GetBorder, 5
 GetBuffer, 4
 GetCallback, 4
 GetColor, 4
 GetColorExtra, 4
 GetCursor, 4
 GetFont, 5
 GetFontNumber, 5
 GetGridFunction, 4
 GetGridFunctionName, 4
 GetGridName, 4
 GetGridNumber, 5
 GetGridType, 4
 GetGridTypeName, 4
 GetGroup, 5

GetHelpString, 5
 GetHintString, 5
 GetImage, 4
 GetImageCoords, 4
 GetIndent, 5
 GetJustify, 5
 GetKidArray, 5
 GetKids, 5
 GetMaxMinSize, 4
 GetParent, 4, 5
 GetRedrawFlags, 4
 GetSize, 4
 GetState, 5
 GetStyle, 5
 GetTextArray, 2, 5
 GetTextString, 2, 5
 GetTexture, 5
 GetTimer, 5
 GetValue, 2
 GetValueArray, 2
 GetValues, 2
 GetWindow, 4
 Grab, 2
 GrabArray, 2, 5
 GrabTextArray, 2, 5
 GrabTextString, 2
 GrabValueArray, 2
 grid, 1, 3, 4, 6
 grid function, 1, 49
 grid number, 6
 grid property, 1
 grid type, 6, 49
 gridFunction, 3, 4, 7
 gridFunctionName, 3, 4, 7
 gridName, 3, 4, 6
 gridType, 3, 4, 6
 gridTypeName, 3, 4, 6
 group, 3, 5, 13

 height, 3, 4, 8
 helpString, 3, 5, 14
 highlightColor, 3, 4, 9
 hightextColor, 3, 4, 9
 hintString, 3, 5, 14

 image, 3, 4, 8
 imageAlign, 3, 8
 imageHeight, 3, 4, 8
 imageIndentX, 3, 4, 8
 imageIndentY, 3, 4, 8
 imageName, 4, 8
 imageStartX, 3, 4, 8
 imageStartY, 3, 4, 8
 imageWidth, 3, 4, 8
 indentBottom, 3, 5, 11
 indentLeft, 3, 5, 11
 indentRight, 3, 5, 11
 indentTop, 3, 5, 11
 Initialize, 4, 5
 inputTextArray, 3, 5, 13
 inputTextString, 3, 5, 13

 justify, 3, 5, 10

 keyboard, 3, 5, 13
 kidArray, 3, 5

 lowlightColor, 3, 4, 9
 lowtextColor, 3, 4, 9

 m #Create, 6, 7, 8
 m #CreateWindow, 6, 7, 8
 m #GetAlign, 10, 11
 m #GetBorder, 12
 m #GetBuffer, 8
 m #GetCallback, 7
 m #GetColor, 9
 m #GetColorExtra, 9
 m #GetCursor, 8
 m #GetFont, 12
 m #GetFontNumber, 12
 m #GetGridFunction, 7
 m #GetGridFunctionName, 7
 m #GetGridName, 6
 m #GetGridNumber, 6
 m #GetGridType, 6
 m #GetGridTypeName, 6
 m #GetGroup, 13
 m #GetHelpString, 14
 m #GetHintString, 14
 m #GetImage, 8
 m #GetImageCoords, 8
 m #GetIndent, 11
 m #GetJustify, 10, 11
 m #GetKidArray, 14
 m #GetKids, 14
 m #GetMaxMinSize, 9
 m #GetMessageFunc, 14
 m #GetMessageFuncArray, 14
 m #GetMessageSub, 14
 m #GetMessageSubArray, 14
 m #GetParent, 7
 m #GetRedrawFlags, 10
 m #GetSize, 8

- m #GetState, 13
- m #GetStyle, 13
- m #GetTextArray, 15
- m #GetTextString, 15
- m #GetTexture, 11
- m #GetTimer, 10
- m #GetValue, 15
- m #GetValueArray, 15
- m #GetValues, 15
- m #GetWindow, 7
- m #GrabTextArray, 15
- m #GrabValueArray, 15
- m #KillTimer, 10
- m #PokeTextArray, 15
- m #PokeValueArray, 15
- m #Resize, 8
- m #SetAlign, 10, 11
- m #SetBorder, 12
- m #SetBuffer, 8
- m #SetCallback, 7
- m #SetColor, 9
- m #SetColorExtra, 9
- m #SetCursor, 8
- m #SetFont, 12
- m #SetFontNumber, 12
- m #SetGridFunction, 7
- m #SetGridFunctionName, 7
- m #SetGridName, 6
- m #SetGridType, 6
- m #SetGridTypeName, 6
- m #SetGroup, 13
- m #SetHelpString, 14
- m #SetHintString, 14
- m #SetImage, 8
- m #SetImageCoords, 8
- m #SetIndent, 11
- m #SetJustify, 10
- m #SetMaxMinSize, 9
- m #SetMessageFunc, 14
- m #SetMessageFuncArray, 14
- m #SetMessageSub, 14
- m #SetMessageSubArray, 14
- m #SetRedrawFlags, 10
- m #SetSize, 8
- m #SetState, 13
- m #SetStyle, 13
- m #SetTextArray, 15
- m #SetTextString, 15
- m #SetTexture, 11
- m #SetTimer, 10
- m #SetValue, 15
- m #SetValueArray, 15

- m #SetValues, 15
- m #StartTimer, 10
- m GetCan, 13
- m SetCan, 13
- maxHeight, 3, 4, 9
- maxWidth, 3, 4, 9
- message, 1, 17, 49, 50
- message function array, 49
- message processing function, 1, 17, 49, 50
- message processing subroutine, 1
- messageFuncArray, 3, 14
- messageSubArray, 3, 14
- minHeight, 3, 4, 9
- minWidth, 3, 4, 9
- mouse, 3, 5, 13

- parent, 3, 4, 7
- Poke, 2
- PokeArray, 2, 5
- PokeTextArray, 2, 5
- PokeTextString, 2
- PokeValueArray, 2
- property, 1

- redraw, 3, 5, 13
- redrawFlags, 3, 4, 10
- RedrawGrid, 4
- Resize, 4

- s Create, 6, 7, 8
- s CreateWindow, 6, 7, 8
- s Initialize, 6, 7, 8, 9, 13, 14
- Set, 2
- SetAlign, 5
- SetBorder, 5
- SetBuffer, 4
- SetCallback, 4
- SetColor, 4
- SetColorExtra, 4
- SetCursor, 4
- SetFont, 5
- SetFontNumber, 5
- SetGridFunction, 4
- SetGridFunctionName, 4
- SetGridName, 4
- SetGridType, 4
- SetGridTypeName, 4
- SetGroup, 5
- SetHelpString, 5
- SetHintString, 5
- SetImage, 4
- SetImageCoords, 4

SetIndent, 5
 SetJustify, 5
 SetMaxMinSize, 4
 SetRedrawFlags, 4
 SetState, 5
 SetStyle, 5
 SetTextArray, 2, 5
 SetTextString, 2, 5
 SetTexture, 5
 SetTimer, 5
 SetValue, 2
 SetValueArray, 2
 SetValues, 2
 standard grid properties, 1
 StartTimer, 5
 state, 3, 5, 13
 style, 3, 5, 13
 styleMax, 3, 5, 13
 SUB Create, 4, 18
 SUB CreateWindow, 4, 18
 SUB Initialize, 4, 5

 textArray, 2, 3, 5, 15
 textString, 2, 3, 5, 15
 texture, 3, 5, 11
 timer, 3, 5, 10

 valueArray, 2, 3, 5, 15

 width, 3, 4, 8
 window, 3, 4, 7

 x, 3, 4, 8
 Xui(), 67
 XuiAlignNameToNumber(), 65
 XuiAlignNumberToName(), 66
 XuiBorderNameToNumber(), 65
 XuiBorderNumberToName(), 66
 XuiBorderNumberToWidth(), 66
 XuiCallback(), 17, 50
 XuiCanNameToNumber(), 65
 XuiCanNumberToName(), 66
 XuiCanNumberToString(), 66
 XuiCanStringToNumber(), 65
 XuiCreateGrid(), 67
 XuiCreateValueArray(), 2, 50
 XuiCreateWindow(), 71
 XuiDestroy(), 50
 XuiDestroyWindow(), 50
 XuiDialog(), 71
 XuiDisable(), 50
 XuiDisplayWindow(), 50
 XuiEnable(), 50
 XuiGetAlign(), 5, 50
 XuiGetBorder(), 5, 51
 XuiGetBuffer(), 4
 XuiGetCallback(), 4, 51
 XuiGetColor(), 4, 51
 XuiGetColorExtra(), 4, 51
 XuiGetCursor(), 4, 51
 XuiGetDefaultColors(), 67
 XuiGetDefaultCursor(), 67
 XuiGetDefaultMessageFuncArray(), 67
 XuiGetDisplay(), 51
 XuiGetEnclosedGrids(), 51
 XuiGetEnclosingGrid(), 51
 XuiGetFont(), 5, 51
 XuiGetFontNumber(), 5, 51
 XuiGetGridFuncName(), 4
 XuiGetGridFunction(), 4, 52
 XuiGetGridFunctionName(), 52
 XuiGetGridName(), 4, 52
 XuiGetGridNumber(), 5, 52
 XuiGetGridRegion(), 67
 XuiGetGridType(), 4, 52
 XuiGetGridTypeMessageFuncArray(), 67
 XuiGetGridTypeMessageSubArray(), 67
 XuiGetGridTypeName(), 4, 52
 XuiGetGridTypeValue(), 68
 XuiGetGridValue(), 68
 XuiGetGroup(), 5, 52
 XuiGetHelp(), 52
 XuiGetHelpFile(), 52
 XuiGetHelpString(), 5, 52
 XuiGetHelpStrings(), 52
 XuiGetHelpWindowGrid(), 68
 XuiGetHintString(), 5, 52
 XuiGetImage(), 4, 53
 XuiGetImageCoords(), 4, 53
 XuiGetIndent(), 5, 53
 XuiGetJustify(), 5, 53
 XuiGetKeyboardFocus(), 53
 XuiGetKeyboardFocusGrid(), 53
 XuiGetKidArray(), 5, 53
 XuiGetKidNumber(), 53
 XuiGetKids(), 5, 53
 XuiGetMaxMinSize(), 4, 53
 XuiGetMessageFunc(), 54
 XuiGetMessageFuncArray(), 54
 XuiGetMessageSub(), 54
 XuiGetMessageSubArray(), 54
 XuiGetModalInfo(), 54
 XuiGetMouseFocus(), 54
 XuiGetNextCallback(), 71

XuiGetParent(), 4, 5, 54
XuiGetRedrawFlags(), 4, 54
XuiGetReply(), 71
XuiGetResponse(), 71
XuiGetSize(), 4, 54
XuiGetSmallestSize(), 54
XuiGetState(), 5, 54
XuiGetStyle(), 5, 55
XuiGetTextArray(), 2, 5, 55
XuiGetTextArrayLine(), 55
XuiGetTextArrayLines(), 55
XuiGetTextArraySize(), 68
XuiGetTextSelectionGrid(), 68
XuiGetTextString(), 2, 5, 55
XuiGetTextStrings(), 55
XuiGetTexture(), 5, 55
XuiGetTimer(), 5, 55
XuiGetValue(), 2, 5, 55
XuiGetValueArray(), 2, 5, 55
XuiGetValues(), 2, 5, 55
XuiGetWindow(), 4, 56
XuiGetWindowFunction(), 56
XuiGetWindowGrid(), 56
XuiGetWindowIcon(), 56
XuiGetWindowSize(), 56
XuiGetWindowTitle(), 56
XuiGotKeyboardFocus(), 56
XuiGrabArray(), 2, 5, 56
XuiGrabTextArray(), 2, 5, 56
XuiGrabTextString(), 2, 56
XuiGrabValueArray(), 2, 5, 56
XuiGridFuncNameToAddr(), 66
XuiGridKid(), 68
XuiGridOverlapsWin(), 68
XuiGridTypeToGridFunc(), 66
XuiHideWindow(), 56
XuiInitialize(), 57
XuiJustifyNameToNumber(), 65
XuiJustifyNumberToName(), 66
XuiKeyboardFocusBackward(), 57
XuiKeyboardFocusForward(), 57
XuiLostKeyboardFocus(), 57
XuiMaximizeWindow(), 57
XuiMessage(), 71
XuiMinimizeWindow(), 57
XuiMonitorContext(), 57
XuiMonitorHelp(), 57
XuiMonitorKeyboard(), 57
XuiMonitorMouse(), 57
XuiMouseEnter(), 57
XuiPassOn(), 68
XuiPlaceWindow(), 68
XuiPokeArray(), 2, 5, 57
XuiPokeTextArray(), 2, 5, 57
XuiPokeTextString(), 2, 58
XuiPokeValueArray(), 2, 5, 58
XuiPositionGrid(), 68
XuiProcessMessage(), 68
XuiQueueCallbacks(), 71
XuiRedraw(), 58
XuiRedrawGrid(), 4, 58
XuiRedrawWindow(), 58
XuiRegisterGridType(), 4, 69
XuiResize(), 4, 58
XuiResizeNot(), 58
XuiResizeWindow(), 58
XuiResizeWindowToGrid(), 58
XuiSelectWindow(), 58
XuiSendMessage(), 1, 69
XuiSendStringMessage(), 71
XuiSendToKid(), 69
XuiSetAlign(), 5, 58
XuiSetBorder(), 5, 59
XuiSetBuffer(), 4
XuiSetCallback(), 4, 59
XuiSetColor(), 4, 59
XuiSetColorAll(), 59
XuiSetColorExtra(), 4, 59
XuiSetColorExtraAll(), 59
XuiSetCursor(), 4, 59
XuiSetDefaultColors(), 69
XuiSetDefaultCursor(), 69
XuiSetFont(), 5, 59
XuiSetFontNumber(), 5, 59
XuiSetGridFuncName(), 4
XuiSetGridFunction(), 4, 60
XuiSetGridFunctionName(), 60
XuiSetGridName(), 4, 60
XuiSetGridType(), 4, 60
XuiSetGridTypeName(), 4, 60
XuiSetGridTypeValue(), 69
XuiSetGridValue(), 69
XuiSetGroup(), 5, 60
XuiSetHelp(), 60
XuiSetHelpFile(), 60
XuiSetHelpString(), 5, 60
XuiSetHelpStrings(), 60
XuiSetHintString(), 5, 60
XuiSetImage(), 4, 60
XuiSetImageCoords(), 4, 61
XuiSetIndent(), 5, 61
XuiSetJustify(), 5, 61
XuiSetKeyboardFocus(), 61
XuiSetKeyboardFocusGrid(), 61

XuiSetMaxMinSize(), 4, 61
XuiSetMessageFunc(), 61
XuiSetMessageFuncArray(), 61
XuiSetMessageSub(), 61
XuiSetMessageSubArray(), 61
XuiSetMouseFocus(), 61
XuiSetRedrawFlags(), 4, 62
XuiSetSize(), 62
XuiSetState(), 5, 62
XuiSetStyle(), 5, 62
XuiSetTextArray(), 2, 5, 62
XuiSetTextArrayLine(), 62
XuiSetTextArrayLines(), 62
XuiSetTextSelectionGrid(), 69
XuiSetTextString(), 2, 5, 62
XuiSetTextStrings(), 62
XuiSetTexture(), 5, 62
XuiSetTimer(), 5, 62
XuiSetValue(), 2, 5, 63
XuiSetValueArray(), 2, 5, 63
XuiSetValues(), 2, 5, 63
XuiSetWindowFunction(), 63
XuiSetWindowIcon(), 63
XuiSetWindowTitle(), 63
XuiShowWindow(), 63
XuiStartTimer(), 5, 63
XuiTextureNameToNumber(), 65
XuiTextureNumberToName(), 66
XuiVersion\$, 69
XuiWindow(), 69

y, 3, 4, 8

