Xt Widget Classes

ApplicationShell

Composite

Constraint

Core

Object

OverrideShell

RectObj

Shell

TopLevelShell

TransientShell

VendorShell

WMShell

ApplicationShell (library call)

NAME

ApplicationShell – The ApplicationShell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

ApplicationShell is used as the main top–level window for an application. An application should have more than one ApplicationShell only if it implements multiple logical applications.

Classes

ApplicationShell inherits behavior and resources from **Core**, **Composite**, **Shell**, **WMShell**, **VendorShell**, and **TopLevelShell**.

The class pointer is **applicationShellWidgetClass**.

The class name is **ApplicationShell**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

ApplicationShell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNargc	XmCArgc	int	0	CSG
XmNargv	XmCArgv	String *	NULL	CSG

XmNargc

Specifies the number of arguments given in the **XmNargv** resource. The function **XtInitialize** sets this resource on the shell widget instance it creates by using its parameters as the values.

XmNargv

Specifies the argument list required by a session manager to restart the application if it is killed. This list should be updated at appropriate points by the application if a new state has been reached that can be directly restarted. The function **XtInitialize** sets this resource on the shell widget instance it creates by using its parameters as the values. When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed.

Inherited Resources

ApplicationShell inherits behavior and resources from the following superclasses. For a complete description of each resource, refer to the reference page for that superclass.

TopLevelShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNiconic	XmCIconic	Boolean	False	CSG
XmNiconName	XmCIconName	String	NULL	CSG
XmNiconNameEncoding	XmCIconNameEncoding	Atom	dynamic	CSG

VendorShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNaudibleWarning	XmCAudibleWarning	unsigned char	XmBELL	CSG
XmNbuttonFontList	XmCButtonFontList	XmFontList	dynamic	CSG
XmNbuttonRenderTable	XmCButtonRenderTable	XmRenderTable	dynamic	CSG
XmNdefaultFontList	XmCDefaultFontList	XmFontList	dynamic	CG
XmNdeleteResponse	XmCDeleteResponse	unsigned char	XmDESTROY	CSG
XmNinputMethod	XmCInputMethod	String	NULL	CSG
XmNinputPolicy	XmCInputPolicy	XmInputPolicy	XmPER_SHELL	CSG
XmNkeyboardFocusPolicy	XmCKeyboardFocusPolicy	unsigned char	XmEXPLICIT	CSG
XmNlabelFontList	XmCLabelFontList	XmFontList	dynamic	CSG
XmNlabelRenderTable	XmCLabelRenderTabel	XmRenderTable	dynamic	CSG
XmNlayoutDirection	XmCLayoutDirection	XmDirection	XmLEFT_TO_RIGHT	CG
XmNmwmDecorations	XmCMwmDecorations	int	−1	CG
XmNmwmFunctions	XmCMwmFunctions	int	−1	CG
XmNmwmInputMode	XmCMwmInputMode	int	−1	CG
XmNmwmMenu	XmCMwmMenu	String	NULL	CG
XmNpreeditType	XmCPreeditType	String	dynamic	CSG
XmNshellUnitType	XmCShellUnitType	unsigned char	XmPIXELS	CSG
XmNtextFontList	XmCTextFontList	XmFontList	dynamic	CSG
XmNtextRenderTable	XmCTextRenderTable	XmRenderTable	dynamic	CSG
XmNunitType	XmCUnitType	unsigned char	XmPIXELS	CSG
XmNuseAsyncGeometry	XmCUseAsyncGeometry	Boolean	False	CSG

WMShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNbaseHeight	XmCBaseHeight	int	XtUnspecifiedShellInt	CSG
XmNbaseWidth	XmCBaseWidth	int	XtUnspecifiedShellInt	CSG
XmNheightInc	XmCHeightInc	int	XtUnspecifiedShellInt	CSG
XmNiconMask	XmCIconMask	Pixmap	NULL	CSG
XmNiconPixmap	XmCIconPixmap	Pixmap	NULL	CSG
XmNiconWindow	XmCIconWindow	Window	NULL	CSG
XmNiconX	XmCIconX	int	XtUnspecifiedShellInt	CSG
XmNiconY	XmCIconY	int	XtUnspecifiedShellInt	CSG
XmNinitialState	XmCInitialState	int	NormalState	CSG
XmNinput	XmCInput	Boolean	True	CSG
XmNmaxAspectX	XmCMaxAspectX	int	XtUnspecifiedShellInt	CSG
XmNmaxAspectY	XmCMaxAspectY	int	XtUnspecifiedShellInt	CSG
XmNmaxHeight	XmCMaxHeight	int	XtUnspecifiedShellInt	CSG
XmNmaxWidth	XmCMaxWidth	int	XtUnspecifiedShellInt	CSG
XmNminAspectX	XmCMinAspectX	int	XtUnspecifiedShellInt	CSG
XmNminAspectY	XmCMinAspectY	int	XtUnspecifiedShellInt	CSG
XmNminHeight	XmCMinHeight	int	XtUnspecifiedShellInt	CSG
XmNminWidth	XmCMinWidth	int	XtUnspecifiedShellInt	CSG
XmNtitle	XmCTitle	String	dynamic	CSG
XmNtitleEncoding	XmCTitleEncoding	Atom	dynamic	CSG
XmNtransient	XmCTransient	Boolean	False	CSG
XmNwaitForWm	XmCWaitForWm	Boolean	True	CSG
XmNwidthInc	XmCWidthInc	int	XtUnspecifiedShellInt	CSG
XmNwindowGroup	XmCWindowGroup	Window	dynamic	CSG
XmNwinGravity	XmCWinGravity	int	dynamic	CSG
XmNwmTimeout	XmCWmTimeout	int	5000 ms	CSG

Shell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	False	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	False	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG

Composite Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for ApplicationShell.

RELATED

Composite(3) , **Core**(3) , **Shell**(3) , **WMShell**(3) , **VendorShell**(3) , and **TopLevelShell**(3) .

Composite (library call)

NAME

Composite – The Composite widget class

SYNOPSIS

#include <Xm/Xm.h>

DESCRIPTION

Composite widgets are intended to be containers for other widgets and can have an arbitrary number of children. Their responsibilities (implemented either directly by the widget class or indirectly by Intrinsics functions) include:

- Overall management of children from creation to destruction.
- Destruction of descendants when the composite widget is destroyed.
- Physical arrangement (geometry management) of a displayable subset of managed children.
- Mapping and unmapping of a subset of the managed children. Instances of composite widgets need to specify the order in which their children are kept. For example, an application may want a set of command buttons in some logical order grouped by function, and it may want buttons that represent filenames to be kept in alphabetical order.

Classes

Composite inherits behavior and resources from **Core**.

The class pointer is **compositeWidgetClass**.

The class name is **Composite**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

Composite Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

XmNchildren

A read–only list of the children of the widget.

XmNinsertPosition

Points to the **XtOrderProc** function described below.

XmNnumChildren

A read–only resource specifying the length of the list of children in **XmNchildren**.

The following procedure pointer in a composite widget instance is of type **XtOrderProc**:
Cardinal (* XtOrderProc) (Widget *w*);

w

Specifies the widget.

Composite widgets that allow clients to order their children (usually homogeneous boxes) can call their widget instance's **XmNinsertPosition** procedure from the class's *insert_child* procedure to determine where a new child should go in its children array. Thus, a client of a composite class can apply different sorting criteria to widget instances of the class, passing in a different **XmNinsertPosition** procedure when it creates each composite widget instance.

The return value of the **XmNinsertPosition** procedure indicates how many children should go before the widget. A value of 0 (zero) indicates that the widget should go before all other children; returning the value of **XmNumChildren** indicates that it should go after all other children. By default, unless a subclass or an application provides an **XmNinsertPosition** procedure, each child is inserted at the end of the **XmNchildren** list. The **XmNinsertPosition** procedure can be overridden by a specific composite widget's resource list or by the argument list provided when the composite widget is created.

Inherited Resources

Composite inherits behavior and resources from the superclass described in the following table. For a complete description of each resource, refer to the reference page for that superclass.

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	1	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for Composite.

RELATED

Core(3) .

Constraint (library call)

NAME

Constraint –The Constraint widget class

SYNOPSIS

#include <Xm/Xm.h>

DESCRIPTION

Constraint widgets maintain additional state data for each child. For example, client−defined constraints on the child's geometry may be specified.

When a constrained composite widget defines constraint resources, all of that widget's children inherit all of those resources as their own. These constraint resources are set and read just the same as any other resources defined for the child. This resource inheritance extends exactly one generation down, which means only the first−generation children of a constrained composite widget inherit the parent widget's constraint resources.

Because constraint resources are defined by the parent widgets and not the children, the child widgets never directly use the constraint resource data. Instead, the parents use constraint resource data to attach child−specific data to children.

Classes

Constraint inherits behavior and resources from **Composite** and **Core**.

The class pointer is **constraintWidgetClass**.

The class name is **Constraint**.

New Resources

Constraint defines no new resources.

Inherited Resources

Constraint inherits behavior and resources from **Composite** and **Core**. The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	1	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for Constraint.

RELATED

Composite(3) and **Core**(3) .

Core (library call)

NAME

Core –The Core widget class

SYNOPSIS

#include <Xm/Xm.h>

DESCRIPTION

Core is the Xt Intrinsic base class for windowed widgets. The **Object** and **RectObj** classes provide support for windowless widgets.

Classes

All widgets are built from **Core**.

The class pointer is **widgetClass**.

The class name is **Core**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	1	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

XmNaccelerators

Specifies a translation table that is bound with its actions in the context of a particular widget. The accelerator table can then be installed on some destination widget. Note that the default accelerators for any widget will always be installed, no matter whether this resource is specified or not.

XmNancestorSensitive

Specifies whether the immediate parent of the widget receives input events. Use the function **XtSetSensitive** to change the argument to preserve data integrity (see **XmNsensitive**). For shells, the default is copied from the parent's **XmNancestorSensitive** resource if there is a parent; otherwise, it is True. For other widgets, the default is the bitwise AND of the parent's **XmNsensitive** and **XmNancestorSensitive** resources.

XmNbackground

Specifies the background color for the widget.

XmNbackgroundPixmap

Specifies a pixmap for tiling the background. The first tile is placed at the upper left corner of the widget's window.

XmNborderColor

Specifies the color of the border in a pixel value.

XmNborderPixmap

Specifies a pixmap to be used for tiling the border. The first tile is placed at the upper left corner of the border.

XmNborderWidth

Specifies the width of the border that surrounds the widget's window on all four sides. The width is specified in pixels. A width of 0 (zero) means that no border shows. Note that you should use resources like **XmNshadowThickness** and **XmNhighlightThickness** instead of **XmNborderWidth** to specify border widths.

XmNcolormap

Specifies the colormap that is used for conversions to the type *Pixel* for this widget instance. When this resource is changed, previously generated pixel values are not affected, but newly generated values are in the new colormap. For shells without parents, the default is the default colormap of the widget's screen. Otherwise, the default is copied from the parent.

XmNdepth

Specifies the number of bits that can be used for each pixel in the widget's window. Applications should not change or set the value of this resource as it is set by the Xt Intrinsics when the widget is created. For shells without parents, the default is the default depth of the widget's screen. Otherwise, the default is copied from the parent.

XmNdestroyCallback

Specifies a list of callbacks that is called when the widget is destroyed.

XmNheight

Specifies the inside height (excluding the border) of the widget's window.

XmNinitialResourcesPersistent

Specifies whether or not resources are reference counted. If the value is True when the widget is created, the resources referenced by the widget are not reference counted, regardless of how the resource type converter is registered. An application that expects to destroy the widget and wants to have resources deallocated should specify a value of False. The default is True, implying an assumption that the widget will not be destroyed during the life of the application.

XmNmappedWhenManaged

If this resource is set to True, it maps the widget (makes it visible) as soon as it is both realized and managed. If this resource is set to False, the client is responsible for mapping and unmapping the widget. If the value is changed from True to False after the widget has been realized and managed, the widget is unmapped.

XmNscreen

Specifies the screen on which a widget instance resides. It is read only. When the Toolkit is initialized, the top–level widget obtains its default value from the default screen of the display. Otherwise, the default is copied from the parent.

XmNsensitive

Determines whether a widget receives input events. If a widget is sensitive, the Xt Intrinsics' Event Manager dispatches to the widget all keyboard, mouse button, motion, window enter/leave, and focus events. Insensitive widgets do not receive these events. Use the function **XtSetSensitive** to change the sensitivity argument. Using **XtSetSensitive** ensures that if a parent widget has **XmNsensitive** set to False, the ancestor−sensitive flag of all its children is appropriately set.

XmNtranslations

Points to a translations list. A translations list is a list of events and actions that are to be performed when the events occur. Note that the default translations for any widget will always be installed, no matter whether this resource is specified or not.

XmNwidth

Specifies the inside width (excluding the border) of the widget's window.

XmNx

Specifies the x−coordinate of the upper left outside corner of the widget's window. The value is relative to the upper left inside corner of the parent window.

XmNy

Specifies the y−coordinate of the upper left outside corner of the widget's window. The value is relative to the upper left inside corner of the parent window.

Translations

There are no translations for Core.

RELATED

Object(3) and **RectObj**(3) .

Object (library call)

NAME

Object – The Object widget class

SYNOPSIS

#include <Xm/Xm.h>

DESCRIPTION

Object is never instantiated. Its sole purpose is as a supporting superclass for other widget classes.

Classes

The class pointer is **objectClass**.

The class name is **Object**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

Object Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C

XmNdestroyCallback

Specifies a list of callbacks that is called when the gadget is destroyed.

Translations

There are no translation for Object.

OverrideShell (library call)

NAME

OverrideShell –The OverrideShell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

OverrideShell is used for shell windows that completely bypass the window manager, for example, PopupMenu shells.

Classes

OverrideShell inherits behavior and resources from **Core**, **Composite**, and **Shell**.

The class pointer is **overrideShellWidgetClass**.

The class name is **OverrideShell**.

New Resources

OverrideShell defines no new resources, but overrides the **XmNoverrideRedirect** and **XmNsaveUnder** resources in the **Shell** class.

Inherited Resources

OverrideShell inherits behavior and resources from the following superclasses. For a complete description of each resource, refer to the reference page for that superclass.

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

Shell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	True	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	True	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG

Composite Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for OverrideShell.

RELATED

Composite(3) , **Core**(3) , and **Shell**(3) .

RectObj (library call)

NAME

RectObj –The RectObj widget class

SYNOPSIS

#include <Xm/Xm.h>

DESCRIPTION

RectObj is never instantiated. Its sole purpose is as a supporting superclass for other widget classes.

Classes

RectObj inherits behavior and a resource from **Object**.

The class pointer is **rectObjClass**.

The class name is **RectObj**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

RectObj Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNborderWidth	XmCBorderWidth	Dimension	1	CSG
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

XmNancestorSensitive

Specifies whether the immediate parent of the gadget receives input events. Use the function **XtSetSensitive** if you are changing the argument to preserve data integrity (see **XmNsensitive**). The default is the bitwise AND of the parent's **XmNsensitive** and **XmNancestorSensitive** resources.

XmNborderWidth

Specifies the width of the border placed around the RectObj's rectangular display area.

XmNheight

Specifies the inside height (excluding the border) of the RectObj's rectangular display area.

XmNsensitive

Determines whether a RectObj receives input events. If a RectObj is sensitive, the parent dispatches to the gadget all keyboard, mouse button, motion, window enter/leave, and focus events. Insensitive gadgets do not receive these events. Use the function **XtSetSensitive** to

change the sensitivity argument. Using **XtSetSensitive** ensures that if a parent widget has **XmNsensitive** set to False, the ancestor −sensitive flag of all its children is appropriately set.

XmNwidth

Specifies the inside width (excluding the border) of the RectObj's rectangular display area.

XmNx

Specifies the x−coordinate of the upper left outside corner of the RectObj's rectangular display area. The value is relative to the upper left inside corner of the parent window.

XmNy

Specifies the y−coordinate of the upper left outside corner of the RectObj's rectangular display area. The value is relative to the upper left inside corner of the parent window.

Inherited Resources

RectObj inherits behavior and a resource from **Object**. For a description of this resource, refer to the **Object** reference page.

Object Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C

Translations

There are no translations for RectObj.

RELATED

Object(3) .

Shell (library call)

NAME

Shell –The Shell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

Shell is a top–level widget (with only one managed child) that encapsulates the interaction with the window manager.

At the time the shell's child is managed, the child's width is used for both widgets if the shell is unrealized and no width has been specified for the shell. Otherwise, the shell's width is used for both widgets. The same relations hold for the height of the shell and its child.

Classes

Shell inherits behavior and resources from **Composite** and **Core**.

The class pointer is **shellWidgetClass**.

The class name is **Shell**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

Shell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	False	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	False	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG

XmNallowShellResize

Specifies that if this resource is False, the Shell widget instance returns **XtGeometryNo** to all geometry requests from its children. All Motif convenience create dialog functions override this default value and set **XmNallowShellResize** to True.

XmNcreatePopupChildProc

Specifies the pointer to a function that is called when the Shell widget instance is popped up by **XtPopup**. The function creates the child widget when the shell is popped up instead of when the application starts up. This can be used if the child needs to be reconfigured each time the shell is popped up. The function takes one argument, the popup shell, and returns no result. It is called after the popup callbacks specified by **XmNpopupCallback**.

XmNgeometry

Specifies the desired geometry for the widget instance. This resource is examined only when the widget instance is unrealized and the number of its managed children is changed. It is used to change the values of the **XmNx**, **XmNy**, **XmNwidth**, and **XmNheight** resources. When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed. In addition, this resource is not copied on creation or by **XtSetValues**. The application must ensure that the string remains valid until the shell is realized.

XmNoverrideRedirect

If True, specifies that the widget instance is a temporary window that should be ignored by the window manager. Applications and users should not normally alter this resource.

XmNpopdownCallback

Specifies a list of callbacks that is called when the widget instance is popped down by **XtPopdown**.

XmNpopupCallback

Specifies a list of callbacks that is called when the widget instance is popped up by **XtPopup**. The second argument to **XtPopup** must be **XtGrabNone**.

XmNsaveUnder

If True, specifies that it is desirable to save the contents of the screen beneath this widget instance, avoiding expose events when the instance is unmapped. This is a hint, and an implementation may save contents whenever it desires, including always or never.

XmNvisual

Specifies the visual used in creating the widget.

Inherited Resources

Shell inherits behavior and resources from the superclass described in the following table. For a complete description of each resource, refer to the reference page for that superclass.

Composite Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for Shell.

RELATED

Composite(3) and **Core**(3) .

TopLevelShell (library call)

NAME

TopLevelShell –The TopLevelShell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

TopLevelShell is used for normal top–level windows such as any additional top–level widgets an application needs.

Classes

TopLevelShell inherits behavior and resources from **Core**, **Composite**, **Shell**, **WMShell**, and **VendorShell**.

The class pointer is **topLevelShellWidgetClass**.

The class name is **TopLevelShell**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

TopLevelShell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNiconic	XmCIconic	Boolean	False	CSG
XmNiconName	XmCIconName	String	NULL	CSG
XmNiconNameEncoding	XmCIconNameEncoding	Atom	dynamic	CSG

XmNiconic

If True when the widget is *created*, specifies that the widget should start as an icon when it is realized. A value of False indicates that the widget is not to be realized as an icon. This resource will only override the **XmNinitialState** resource when specified in the call that creates the widget.

XmNiconName

Specifies the short form of the application name to be displayed by the window manager when the application is iconified. When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed.

XmNiconNameEncoding

Specifies a property type that represents the encoding of the **XmNiconName** string. If a language procedure has been set, the default is None; otherwise, the default is **XA_STRING**. When the widget is realized, if the value is None, the corresponding name is assumed to be in the current locale. The name is passed to **XmbTextListToTextProperty** with an encoding style of **XStdICCTextStyle**. The resulting encoding is **STRING** if the name is fully convertible to **STRING**, otherwise **COMPOUND_TEXT**. The values of the encoding resources are not changed; they remain None.

Inherited Resources

TopLevelShell inherits behavior and resources from the following superclasses. For a complete description of each resource, refer to the reference page for that superclass.

VendorShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNaudibleWarning	XmCAudibleWarning	unsigned char	XmBELL	CSG
XmNbuttonFontList	XmCButtonFontList	XmFontList	dynamic	CSG
XmNbuttonRenderTable	XmCButtonRenderTable	XmRenderTable	dynamic	CSG
XmNdefaultFontList	XmCDefaultFontList	XmFontList	dynamic	CG
XmNdeleteResponse	XmCDeleteResponse	unsigned char	XmDESTROY	CSG
XmNinputMethod	XmCInputMethod	String	NULL	CSG
XmNinputPolicy	XmCInputPolicy	XmInputPolicy	XmPER_SHELL	CSG
XmNkeyboardFocusPolicy	XmCKeyboardFocusPolicy	unsigned char	XmEXPLICIT	CSG
XmNlabelFontList	XmCLabelFontList	XmFontList	dynamic	CSG
XmNlabelRenderTable	XmCLabelRenderTabel	XmRenderTable	dynamic	CSG
XmNlayoutDirection	XmCLayoutDirection	XmDirection	XmLEFT_TO_RIGHT	CG
XmNmwmDecorations	XmCMwmDecorations	int	−1	CG
XmNmwmFunctions	XmCMwmFunctions	int	−1	CG
XmNmwmInputMode	XmCMwmInputMode	int	−1	CG
XmNmwmMenu	XmCMwmMenu	String	NULL	CG
XmNpreeditType	XmCPreeditType	String	dynamic	CSG
XmNshellUnitType	XmCShellUnitType	unsigned char	XmPIXELS	CSG
XmNtextFontList	XmCTextFontList	XmFontList	dynamic	CSG
XmNtextRenderTable	XmCTextRenderTable	XmRenderTable	dynamic	CSG
XmNunitType	XmCUnitType	unsigned char	XmPIXELS	CSG
XmNuseAsyncGeometry	XmCUseAsyncGeometry	Boolean	False	CSG

WMShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNbaseHeight	XmCBaseHeight	int	XtUnspecifiedShellInt	CSG
XmNbaseWidth	XmCBaseWidth	int	XtUnspecifiedShellInt	CSG
XmNheightInc	XmCHeightInc	int	XtUnspecifiedShellInt	CSG
XmNiconMask	XmCIconMask	Pixmap	NULL	CSG
XmNiconPixmap	XmCIconPixmap	Pixmap	NULL	CSG
XmNiconWindow	XmCIconWindow	Window	NULL	CSG
XmNiconX	XmCIconX	int	XtUnspecifiedShellInt	CSG
XmNiconY	XmCIconY	int	XtUnspecifiedShellInt	CSG
XmNinitialState	XmCInitialState	int	NormalState	CSG
XmNinput	XmCInput	Boolean	True	CSG
XmNmaxAspectX	XmCMaxAspectX	int	XtUnspecifiedShellInt	CSG
XmNmaxAspectY	XmCMaxAspectY	int	XtUnspecifiedShellInt	CSG
XmNmaxHeight	XmCMaxHeight	int	XtUnspecifiedShellInt	CSG
XmNmaxWidth	XmCMaxWidth	int	XtUnspecifiedShellInt	CSG
XmNminAspectX	XmCMinAspectX	int	XtUnspecifiedShellInt	CSG
XmNminAspectY	XmCMinAspectY	int	XtUnspecifiedShellInt	CSG
XmNminHeight	XmCMinHeight	int	XtUnspecifiedShellInt	CSG
XmNminWidth	XmCMinWidth	int	XtUnspecifiedShellInt	CSG
XmNtitle	XmCTitle	String	dynamic	CSG
XmNtitleEncoding	XmCTitleEncoding	Atom	dynamic	CSG
XmNtransient	XmCTransient	Boolean	False	CSG
XmNwaitForWm	XmCWaitForWm	Boolean	True	CSG
XmNwidthInc	XmCWidthInc	int	XtUnspecifiedShellInt	CSG
XmNwindowGroup	XmCWindowGroup	Window	dynamic	CSG
XmNwinGravity	XmCWinGravity	int	dynamic	CSG
XmNwmTimeout	XmCWmTimeout	int	5000 ms	CSG

Shell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	False	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	False	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG

Composite Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for TopLevelShell.

RELATED

Composite(3) , **Core**(3) , **Shell**(3) , **WMShell**(3) , and **VendorShell**(3) .

TransientShell (library call)

NAME

TransientShell – The TransientShell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

TransientShell is used for shell windows that can be manipulated by the window manager, but are not allowed to be iconified separately. For example, DialogBoxes make no sense without their associated application. They are iconified by the window manager only if the main application shell is iconified.

Classes

TransientShell inherits behavior and resources from **Core**, **Composite**, **Shell**, **WMShell**, and **VendorShell**.

The class pointer is **transientShellWidgetClass**.

The class name is **TransientShell**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

In addition to these new resources, **TransientShell** overrides the **XmNsaveUnder** resource in **Shell** and the **XmNtransient** resource in **WMShell**.

TransientShell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNtransientFor	XmCTransientFor	Widget	NULL	CSG

XmNtransientFor

Specifies a widget that the shell acts as a pop–up for. If this resource is NULL or is a widget that has not been realized, the **XmNwindowGroup** is used instead.

Inherited Resources

TransientShell inherits behavior and resources from the superclasses described in the following tables, which define sets of widget resources used by the programmer to specify data. For a complete description of each resource, refer to the reference page for that superclass.

The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

VendorShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNaudibleWarning	XmCAudibleWarning	unsigned char	XmBELL	CSG
XmNbuttonFontList	XmCButtonFontList	XmFontList	dynamic	CSG
XmNbuttonRenderTable	XmCButtonRenderTable	XmRenderTable	dynamic	CSG
XmNdefaultFontList	XmCDefaultFontList	XmFontList	dynamic	CG
XmNdeleteResponse	XmCDeleteResponse	unsigned char	XmDESTROY	CSG
XmNinputMethod	XmCInputMethod	String	NULL	CSG
XmNinputPolicy	XmCInputPolicy	XmInputPolicy	XmPER_SHELL	CSG
XmNkeyboardFocusPolicy	XmCKeyboardFocusPolicy	unsigned char	XmEXPLICIT	CSG
XmNlabelFontList	XmCLabelFontList	XmFontList	dynamic	CSG
XmNlabelRenderTable	XmCLabelRenderTabel	XmRenderTable	dynamic	CSG
XmNlayoutDirection	XmCLayoutDirection	XmDirection	XmLEFT_TO_RIGHT	CG
XmNmwmDecorations	XmCMwmDecorations	int	−1	CG
XmNmwmFunctions	XmCMwmFunctions	int	−1	CG
XmNmwmInputMode	XmCMwmInputMode	int	−1	CG
XmNmwmMenu	XmCMwmMenu	String	NULL	CG
XmNpreeditType	XmCPreeditType	String	dynamic	CSG
XmNshellUnitType	XmCShellUnitType	unsigned char	XmPIXELS	CSG
XmNtextFontList	XmCTextFontList	XmFontList	dynamic	CSG
XmNtextRenderTable	XmCTextRenderTable	XmRenderTable	dynamic	CSG
XmNunitType	XmCUnitType	unsigned char	XmPIXELS	CSG
XmNuseAsyncGeometry	XmCUseAsyncGeometry	Boolean	False	CSG

WMShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNbaseHeight	XmCBaseHeight	int	XtUnspecifiedShellInt	CSG
XmNbaseWidth	XmCBaseWidth	int	XtUnspecifiedShellInt	CSG
XmNheightInc	XmCHeightInc	int	XtUnspecifiedShellInt	CSG
XmNiconMask	XmCIconMask	Pixmap	NULL	CSG
XmNiconPixmap	XmCIconPixmap	Pixmap	NULL	CSG
XmNiconWindow	XmCIconWindow	Window	NULL	CSG
XmNiconX	XmCIconX	int	XtUnspecifiedShellInt	CSG
XmNiconY	XmCIconY	int	XtUnspecifiedShellInt	CSG
XmNinitialState	XmCInitialState	int	NormalState	CSG
XmNinput	XmCInput	Boolean	True	CSG
XmNmaxAspectX	XmCMaxAspectX	int	XtUnspecifiedShellInt	CSG
XmNmaxAspectY	XmCMaxAspectY	int	XtUnspecifiedShellInt	CSG
XmNmaxHeight	XmCMaxHeight	int	XtUnspecifiedShellInt	CSG
XmNmaxWidth	XmCMaxWidth	int	XtUnspecifiedShellInt	CSG
XmNminAspectX	XmCMinAspectX	int	XtUnspecifiedShellInt	CSG
XmNminAspectY	XmCMinAspectY	int	XtUnspecifiedShellInt	CSG
XmNminHeight	XmCMinHeight	int	XtUnspecifiedShellInt	CSG
XmNminWidth	XmCMinWidth	int	XtUnspecifiedShellInt	CSG
XmNtitle	XmCTitle	String	dynamic	CSG
XmNtitleEncoding	XmCTitleEncoding	Atom	dynamic	CSG
XmNtransient	XmCTransient	Boolean	True	CSG
XmNwaitForWm	XmCWaitForWm	Boolean	True	CSG
XmNwidthInc	XmCWidthInc	int	XtUnspecifiedShellInt	CSG
XmNwindowGroup	XmCWindowGroup	Window	dynamic	CSG
XmNwinGravity	XmCWinGravity	int	dynamic	CSG
XmNwmTimeout	XmCWmTimeout	int	5000 ms	CSG

Shell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	False	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	True	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG

Composite Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for TransientShell.

RELATED

Composite(3) , **Core**(3) , **Shell**(3) , **VendorShell**(3) , and **WMShell**(3) .

VendorShell (library call)

NAME

VendorShell –The VendorShell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

VendorShell is a Motif widget class used as a supporting superclass for all shell classes that are visible to the window manager and that are not override redirect. It contains resources that describe the MWM−specific look and feel. It also manages the MWM−specific communication needed by all VendorShell subclasses. See the **mwm** reference page for more information.

If an application uses the **XmNmwmDecorations**, **XmNmwmFunctions**, or **XmNmwmInputMode** resource, it should include the file `Xm/MwmUtil.h`.

Setting **XmNheight**, **XmNwidth**, or **XmNborderWidth** for either a VendorShell or its managed child usually sets that resource to the same value in both the parent and the child. When an off−the−spot input method exists, the height and width of the shell may be greater than those of the managed child in order to accommodate the input method. In this case, setting **XmNheight** or **XmNwidth** for the shell does not necessarily set that resource to the same value in the managed child, and setting **XmNheight** or **XmNwidth** for the child does not necessarily set that resource to the same value in the shell.

For the managed child of a VendorShell, regardless of the value of the shell's **XmNallowShellResize**, setting **XmNx** or **XmNy** sets the corresponding resource of the parent but does not change the child's position relative to the parent. **XtGetValues** for the child's **XmNx** or **XmNy** yields the value of the corresponding resource in the parent. The x and y−coordinates of the child's upper left outside corner relative to the parent's upper left inside corner are both 0 (zero) minus the value of **XmNborderWidth**.

Note that the *Inter−Client Communication Conventions Manual* (ICCCM) allows a window manager to change or control the border width of a reparented top−level window.

VendorShell holds the **XmQTspecifyRenderTable** trait.

Classes

VendorShell inherits behavior, resources, and traits from the **Core**, **Composite**, **Shell**, and **WMShell** classes.

The class pointer is **vendorShellWidgetClass**.

The class name is **VendorShell**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a subresource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a subresource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given subresource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

VendorShell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaudibleWarning	XmCAudibleWarning	unsigned char	XmBELL	CSG
XmNbuttonFontList	XmCButtonFontList	XmFontList	dynamic	CSG
XmNbuttonRenderTable	XmCButtonRenderTable	XmRenderTable	dynamic	CSG
XmNdefaultFontList	XmCDefaultFontList	XmFontList	dynamic	CG
XmNdeleteResponse	XmCDeleteResponse	unsigned char	XmDESTROY	CSG
XmNinputMethod	XmCInputMethod	string	NULL	CSG
XmNinputPolicy	XmCInputPolicy	XmInputPolicy	XmPER_SHELL	CSG
XmNkeyboardFocusPolicy	XmCKeyboardFocusPolicy	unsigned char	XmEXPLICIT	CSG
XmNlabelFontList	XmCLabelFontList	XmFontList	dynamic	CSG
XmNlabelRenderTable	XmCLabelRenderTabel	XmRenderTable	dynamic	CSG
XmNlayoutDirection	XmCLayoutDirection	XmDirection	XmLEFT_TO_RIGHT	CG
XmNmwmDecorations	XmCMwmDecorations	int	−1	CG
XmNmwmFunctions	XmCMwmFunctions	int	−1	CG
XmNmwmInputMode	XmCMwmInputMode	int	−1	CG
XmNmwmMenu	XmCMwmMenu	String	NULL	CG
XmNpreeditType	XmCPreeditType	String	dynamic	CSG
XmNverifyPreedit	XmCVerifyPreedit	Boolean	False	CSG
XmNshellUnitType	XmCShellUnitType	unsigned char	XmPIXELS	CSG
XmNtextFontList	XmCTextFontList	XmFontList	dynamic	CSG
XmNtextRenderTable	XmCTextRenderTable	XmRenderTable	dynamic	CSG
XmNunitType	XmCUnitType	unsigned char	XmPIXELS	CSG
XmNuseAsyncGeometry	XmCUseAsyncGeometry	Boolean	False	CSG

XmNaudibleWarning

Determines whether an action activates its associated audible cue. The possible values are **XmBELL** and **XmNONE**.

XmNbuttonFontList

Specifies the font list used for button descendants. See the **XmNbuttonRenderTable** resource.

XmNbuttonRenderTable

Specifies the render table used for VendorShell's button descendants. If this value is NULL at initialization and if the value of **XmNdefaultFontList** is not NULL, **XmNbuttonRenderTable** is initialized to the value of **XmNdefaultFontList**. If the value of **XmNdefaultFontList** is NULL, the parent hierarchy of the widget is searched for an ancestor that holds the **XmQTspecifyRenderTable** trait. If such an ancestor is found, **XmNbuttonRenderTable** is initialized to the **XmBUTTON_RENDER_TABLE** value of the ancestor widget. If no such ancestor is found, the default is implementation dependent. Refer to **XmRenderTable(3)** for more information on the creation and structure of a render table.

XmNdefaultFontList

Specifies a default font list for VendorShell's descendants. This resource is obsolete and exists for compatibility with earlier releases. It has been replaced by **XmNbuttonFontList**, **XmNlabelFontList**, and **XmNtextFontList**.

XmNdeleteResponse

Determines what action the shell takes in response to a **WM_DELETE_WINDOW** message. The setting can be one of three values: **XmDESTROY**, **XmUNMAP**, and **XmDO_NOTHING**. The resource is scanned, and the appropriate action is taken after the **WM_DELETE_WINDOW** callback list (if any) that is registered with the Protocol manager has been called.

XmNinputMethod

Specifies the string that sets the locale modifier for the input method. When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed.

XmNinputPolicy

Specifies the policy to follow for creating an Input Context (IC) for this shell. This resource can have the following values:

XmPER_SHELL

Specifies that only one XIC is created per shell.

XmPER_WIDGET

Specifies that one XIC is created for each widget.

XmNkeyboardFocusPolicy

Determines allocation of keyboard focus within the widget hierarchy rooted at this shell. The X keyboard focus must be directed to somewhere in the hierarchy for this client−side focus management to take effect. Possible values are **XmEXPLICIT**, specifying a click−to−type policy, and **XmPOINTER**, specifying a pointer−driven policy.

XmNlabelFontList

Specifies the font list used for label descendants. See the **XmNlabelRenderTable** resource.

XmNlabelRenderTable

Specifies the font list used for VendorShell's label descendants (Labels and LabelGadgets). If this value is NULL at initialization and if the value of **XmNdefaultFontList** is not NULL, **XmNlabelFontList** is initialized to the value of **XmNdefaultFontList**. If the value of **XmNdefaultFontList** is NULL, the parent hierarchy of the widget is searched for an ancestor that holds the **XmQTspecifyRenderTable** trait. If such an ancestor is found, **XmNlabelRenderTable** is initialized to the **XmLABEL_RENDER_TABLE** of the ancestor widget. If no such ancestor is found, the default is implementation dependent. Refer to **XmRenderTable**(3) for more information on the creation and structure of a render table.

XmNlayoutDirection

Specifies the direction in which the subwidgets, children of a widget, or other visual components are to be laid out. This policy will apply as the default layout policy for all descendants of this VendorShell.

XmNmwmDecorations

Specifies the decoration flags (specific decorations to add or remove from the window manager frame) for the **_MOTIF_WM_HINTS** property. If any decoration flags are specified by the **_MOTIF_WM_HINTS** property, only decorations indicated by both that property and the MWM *clientDecoration* and *transientDecoration* resources are displayed. If no decoration flags are specified by the **_MOTIF_WM_HINTS** property, decorations indicated by the MWM *clientDecoration* and *transientDecoration* resources are displayed. The default for the **XmNmwmDecorations** resource is not to specify any decoration flags for the **_MOTIF_WM_HINTS** property.

The value of this resource is the bitwise inclusive OR of one or more flag bits. The possible flag bit constants, defined in the include file `Xm/MwmUtil.h`, are

MWM_DECOR_ALL

All decorations *except* those specified by other flag bits that are set

MWM_DECOR_BORDER

Client window border

MWM_DECOR_RESIZEH

Resize frame handles

MWM_DECOR_TITLE

Title bar

MWM_DECOR_MENU

Window menu button

MWM_DECOR_MINIMIZE

Minimize window button

MWM_DECOR_MAXIMIZE

Maximize window button

XmNmwmFunctions

Specifies the function flags (specific window manager functions to apply or not apply to the client window) for the **_MOTIF_WM_HINTS** property. If any function flags are specified by the **_MOTIF_WM_HINTS** property, only functions indicated by both that property and the MWM *clientFunctions* and *transientFunctions* resources are applied. If no function flags are specified by the **_MOTIF_WM_HINTS** property, functions indicated by the MWM *clientFunctions* and *transientFunctions* resources are applied. The default for the **XmNmwmFunctions** resource is not to specify any function flags for the **_MOTIF_WM_HINTS** property.

The value of this resource is the bitwise inclusive OR of one or more flag bits. The possible flag bit constants, defined in the include file `Xm/MwmUtil.h`, are

MWM_FUNC_ALL

All functions *except* those specified by other flag bits that are set

MWM_FUNC_RESIZE

f.resize

MWM_FUNC_MOVE

f.move

MWM_FUNC_MINIMIZE

f.minimize

MWM_FUNC_MAXIMIZE

f.maximize

MWM_FUNC_CLOSE

f.kill

XmNmwmInputMode

Specifies the input mode flag (application modal or system modal input constraints) for the **_MOTIF_WM_HINTS** property. If no input mode flag is specified by the **_MOTIF_WM_HINTS** property, no input constraints are applied, and input goes to any window. The default for the **XmNmwmInputMode** resource is not to specify any input mode flag for the **_MOTIF_WM_HINTS** property.

An application that sets input constraints on a dialog usually uses the BulletinBoard's **XmNdialogStyle** resource rather than the parent DialogShell's **XmNmwmInputMode** resource.

The possible values for this resource, defined in the include file `Xm/MwmUtil.h`, are

MWM_INPUT_MODELESS

Input goes to any window.

MWM_INPUT_PRIMARY_APPLICATION_MODAL

Input does not go to ancestors of this window.

MWM_INPUT_SYSTEM_MODAL

Input goes only to this window.

MWM_INPUT_FULL_APPLICATION_MODAL

Input does not go to other windows in this application.

XmNmwmMenu

Specifies the menu items that the Motif window manager should add to the end of the window menu. The string contains a list of items separated by \n with the following format:
label [mnemonic] [accelerator] function

If more than one item is specified, the items should be separated by a newline character.

When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed.

XmNpreeditType

Specifies the input method style or styles available to the input manager. The resource can be a comma–separated list of the following values:

Preedit Values	
Preedit Value	*XIM Style*
OffTheSpot	XIMPreeditArea
Root	XIMPreeditNothing
None	XIMPreeditNone
OverTheSpot	XIMPreeditPosition
OnTheSpot	XIMPreeditCallbacks

When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed.

XmNshellUnitType

This resource is obsolete, and is included only for compatibility with earlier releases of Motif. Use the **XmNunitType** resource instead.

XmNtextFontList

Specifies the font list used for text descendants. See the **XmNtextRenderTable** resource.

XmNtextRenderTable

Specifies the render table used for VendorShell's Text and List descendants. If this value is NULL at initialization and if the value of **XmNdefaultFontList** is not NULL, **XmNtextRenderTable** is initialized to the value of **XmNdefaultFontList**. If the value of **XmNdefaultFontList** is NULL, the parent hierarchy of the widget is searched for an ancestor that holds the **XmQTspecifyRenderTable** trait. If such an ancestor is found, **XmNtextRenderTable** is initialized to the **XmTEXT_RENDER_TABLE** value of the ancestor widget. If no such ancestor is found, the default is implementation dependent. Refer to **XmRenderTable**(3) for more information on the creation and structure of a render table.

XmNunitType

Provides the basic support for resolution independence. It defines the type of units a widget uses with sizing and positioning resources. The resource has a default unit type of **XmPIXELS**.

The unit type can also be specified in resource files, with the following format:

<floating value><unit>

where:

unit

is <" ", pixels, inches, centimeters, millimeters, points, font units>

pixels

is <*pix*, *pixel*, *pixels*>

inches

is <*in*, *inch*, *inches*>

centimeter

is <*cm*, *centimeter*, *centimeters*>

millimeters

is <*mm*, *millimeter*, *millimeters*>

points

is <*pt*, *point*, *points*>

font units

is <*fu*, *font_unit*, *font_units*>

float

is {"+"|"−"}{{<"0"−"9">*}.}<"0"−"9">*

Note that the type Dimension must always be positive.

For example,

```
xmfonts*XmMainWindow.height: 10.4cm
*PostIn.width: 3inches
```

XmNunitType can have the following values:

XmPIXELS

All values provided to the widget are treated as normal pixel values.

XmMILLIMETERS

All values provided to the widget are treated as normal millimeter values.

Xm100TH_MILLIMETERS

All values provided to the widget are treated as 1/100 of a millimeter.

XmCENTIMETERS

All values provided to the widget are treated as normal centimeter values.

XmINCHES

All values provided to the widget are treated as normal inch values.

Xm1000TH_INCHES

All values provided to the widget are treated as 1/1000 of an inch.

XmPOINTS

All values provided to the widget are treated as normal point values. A point is a unit used in text processing applications and is defined as 1/72 of an inch.

Xm100TH_POINTS

All values provided to the widget are treated as 1/100 of a point. A point is a unit used in text processing applications and is defined as 1/72 of an inch.

XmFONT_UNITS

All values provided to the widget are treated as normal font units. A font unit has horizontal and vertical components. These are the values of the XmScreen resources **XmNhorizontalFontUnit** and **XmNverticalFontUnit**.

Xm100TH_FONT_UNITS

All values provided to the widget are treated as 1/100 of a font unit. A font unit has horizontal and vertical components. These are

the values of the **XmScreen** resources **XmNhorizontalFontUnit** and **XmNverticalFontUnit**.

For more information about units, refer to the **XmConvertUnits** reference page.

XmNuseAsyncGeometry

Specifies whether the geometry manager should wait for confirmation of a geometry request to the window manager. When the value of this resource is True, the geometry manager forces **XmNwaitForWm** to False and **XmNwmTimeout** to 0, and it relies on asynchronous notification. When the value of this resource is False, **XmNwaitForWm** and **XmNwmTimeout** are unaffected. The default is False.

Inherited Resources

VendorShell inherits behavior and resources from the superclasses described in the following tables. For a complete description of each resource, refer to the reference page for that superclass.

WMShell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNbaseHeight	XmCBaseHeight	int	XtUnspecifiedShellInt	CSG
XmNbaseWidth	XmCBaseWidth	int	XtUnspecifiedShellInt	CSG
XmNheightInc	XmCHeightInc	int	XtUnspecifiedShellInt	CSG
XmNiconMask	XmCIconMask	Pixmap	NULL	CSG
XmNiconPixmap	XmCIconPixmap	Pixmap	NULL	CSG
XmNiconWindow	XmCIconWindow	Window	NULL	CSG
XmNiconX	XmCIconX	int	XtUnspecifiedShellInt	CSG
XmNiconY	XmCIconY	int	XtUnspecifiedShellInt	CSG
XmNinitialState	XmCInitialState	int	NormalState	CSG
XmNinput	XmCInput	Boolean	True	CSG
XmNmaxAspectX	XmCMaxAspectX	int	XtUnspecifiedShellInt	CSG
XmNmaxAspectY	XmCMaxAspectY	int	XtUnspecifiedShellInt	CSG
XmNmaxHeight	XmCMaxHeight	int	XtUnspecifiedShellInt	CSG
XmNmaxWidth	XmCMaxWidth	int	XtUnspecifiedShellInt	CSG
XmNminAspectX	XmCMinAspectX	int	XtUnspecifiedShellInt	CSG
XmNminAspectY	XmCMinAspectY	int	XtUnspecifiedShellInt	CSG
XmNminHeight	XmCMinHeight	int	XtUnspecifiedShellInt	CSG
XmNminWidth	XmCMinWidth	int	XtUnspecifiedShellInt	CSG
XmNtitle	XmCTitle	String	dynamic	CSG
XmNtitleEncoding	XmCTitleEncoding	Atom	dynamic	CSG
XmNtransient	XmCTransient	Boolean	False	CSG
XmNwaitForWm	XmCWaitForWm	Boolean	True	CSG
XmNwidthInc	XmCWidthInc	int	XtUnspecifiedShellInt	CSG
XmNwindowGroup	XmCWindowGroup	Window	dynamic	CSG
mNwinGravity	XmCWinGravity	int	dynamic	CSG
XmNwmTimeout	XmCWmTimeout	int	5000 ms	CSG
Shell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	False	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	False	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG
Composite Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for VendorShell.

RELATED

Composite(3) , **Core**(3) , **mwm**(1) , **Shell**(3) , **WMShell**(3) , **XmActivateProtocol**(3) , **XmActivateWMProtocol**(3) , **XmAddProtocolCallback**(3) , **XmAddWMProtocolCallback**(3) , **XmAddProtocols**(3) , **XmAddWMProtocols**(3) , **XmDeactivateProtocol**(3) , **XmDeactivateWMProtocol**(3) , **XmGetAtomName**(3) , **XmInternAtom**(3) , **XmIsMotifWMRunning**(3) , **XmRemoveProtocolCallback**(3) , **XmRemoveWMProtocolCallback**(3) , **XmRemoveProtocols**(3) , **XmRemoveWMProtocols**(3) , **XmScreen**(3) , **XmSetProtocolHooks**(3) , and **XmSetWMProtocolHooks**(3) .

WMShell (library call)

NAME

WMShell – The WMShell widget class

SYNOPSIS

#include <Xm/Xm.h>
#include <X11/Shell.h>

DESCRIPTION

WMShell is a top–level widget that encapsulates the interaction with the window manager.

Classes

WMShell inherits behavior and resources from the **Core**, **Composite**, and **Shell** classes.

The class pointer is **wmShellWidgetClass**.

The class name is **WMShell**.

New Resources

The following table defines a set of widget resources used by the programmer to specify data. The programmer can also set the resource values for the inherited classes to set attributes for this widget. To reference a resource by name or by class in a `.Xdefaults` file, remove the **XmN** or **XmC** prefix and use the remaining letters. To specify one of the defined values for a resource in a `.Xdefaults` file, remove the **Xm** prefix and use the remaining letters (in either lowercase or uppercase, but include any underscores between words). The codes in the access column indicate if the given resource can be set at creation time (C), set by using **XtSetValues** (S), retrieved by using **XtGetValues** (G), or is not applicable (N/A).

WMShell Resource Set

Name	*Class*	*Type*	*Default*	*Access*
XmNbaseHeight	XmCBaseHeight	int	XtUnspecifiedShellInt	CSG
XmNbaseWidth	XmCBaseWidth	int	XtUnspecifiedShellInt	CSG
XmNheightInc	XmCHeightInc	int	XtUnspecifiedShellInt	CSG
XmNiconMask	XmCIconMask	Pixmap	NULL	CSG
XmNiconPixmap	XmCIconPixmap	Pixmap	NULL	CSG
XmNiconWindow	XmCIconWindow	Window	NULL	CSG
XmNiconX	XmCIconX	int	XtUnspecifiedShellInt	CSG
XmNiconY	XmCIconY	int	XtUnspecifiedShellInt	CSG
XmNinitialState	XmCInitialState	int	NormalState	CSG
XmNinput	XmCInput	Boolean	False	CSG
XmNmaxAspectX	XmCMaxAspectX	int	XtUnspecifiedShellInt	CSG
XmNmaxAspectY	XmCMaxAspectY	int	XtUnspecifiedShellInt	CSG
XmNmaxHeight	XmCMaxHeight	int	XtUnspecifiedShellInt	CSG
XmNmaxWidth	XmCMaxWidth	int	XtUnspecifiedShellInt	CSG
XmNminAspectX	XmCMinAspectX	int	XtUnspecifiedShellInt	CSG
XmNminAspectY	XmCMinAspectY	int	XtUnspecifiedShellInt	CSG
XmNminHeight	XmCMinHeight	int	XtUnspecifiedShellInt	CSG
XmNminWidth	XmCMinWidth	int	XtUnspecifiedShellInt	CSG
XmNtitle	XmCTitle	String	dynamic	CSG
XmNtitleEncoding	XmCTitleEncoding	Atom	dynamic	CSG
XmNtransient	XmCTransient	Boolean	False	CSG
XmNwaitForWm	XmCWaitForWm	Boolean	True	CSG
XmNwidthInc	XmCWidthInc	int	XtUnspecifiedShellInt	CSG
XmNwindowGroup	XmCWindowGroup	Window	dynamic	CSG
XmNwinGravity	XmCWinGravity	int	dynamic	CSG
XmNwmTimeout	XmCWmTimeout	int	5000 ms	CSG

XmNbaseHeight

Specifies the base for a progression of preferred heights for the window manager to use in sizing the widget. The preferred heights are **XmNbaseHeight** plus integral multiples of **XmNheightInc**, with a minimum of **XmNminHeight** and a maximum of **XmNmaxHeight**. If an initial value is not supplied for **XmNbaseHeight** but is supplied for **XmNbaseWidth**, the value of **XmNbaseHeight** is set to 0 (zero) when the widget is realized.

XmNbaseWidth

Specifies the base for a progression of preferred widths for the window manager to use in sizing the widget. The preferred widths are **XmNbaseWidth** plus integral multiples of **XmNwidthInc**, with a minimum of **XmNminWidth** and a maximum of **XmNmaxWidth**. If an initial value is not supplied for **XmNbaseWidth** but is supplied for **XmNbaseHeight**, the value of **XmNbaseWidth** is set to 0 (zero) when the widget is realized.

XmNheightInc

Specifies the increment for a progression of preferred heights for the window manager to use in sizing the widget. The preferred heights are **XmNbaseHeight** plus integral multiples of **XmNheightInc**, with a minimum of **XmNminHeight** and a maximum of **XmNmaxHeight**. If an initial value is not supplied for **XmNheightInc** but is supplied for **XmNwidthInc**, the value of **XmNheightInc** is set to 1 when the widget is realized.

XmNiconMask

Specifies a bitmap that could be used by the window manager to clip the **XmNiconPixmap** bitmap to make the icon nonrectangular.

XmNiconPixmap

Specifies a bitmap that could be used by the window manager as the application's icon.

XmNiconWindow

Specifies the ID of a window that could be used by the window manager as the application's icon.

XmNiconX

Specifies a suitable place to put the application's icon; this is a hint to the window manager in root window coordinates. Because the window manager controls icon placement policy, this resource may be ignored.

XmNiconY

Specifies a suitable place to put the application's icon; this is a hint to the window manager in root window coordinates. Because the window manager controls icon placement policy, this resource may be ignored.

XmNinitialState

Specifies the state the application wants the widget instance to start in. It must be one of the constants **NormalState** or **IconicState**.

XmNinput

Specifies the application's input model for this widget and its descendants. The meaning of a True or False value for this resource depends on the presence or absence of a WM_TAKE_FOCUS atom in the WM_PROTOCOLS property:

Input Model	**XmNinput**	**WM_TAKE_FOCUS**
No input	False	Absent
Passive	True	Absent
Locally active	True	Present
Globally active	False	Present

For more information on input models, see the X Consortium Standard *Inter−Client Communication Conventions Manual* (ICCCM).

XmNmaxAspectX

Specifies the numerator of the maximum aspect ratio (X/Y) that the application wants the widget instance to have.

XmNmaxAspectY

Specifies the denominator of the maximum aspect ratio (X/Y) that the application wants the widget instance to have.

XmNmaxHeight

Specifies the maximum height that the application wants the widget instance to have.

XmNmaxWidth

Specifies the maximum width that the application wants the widget instance to have.

XmNminAspectX

Specifies the numerator of the minimum aspect ratio (X/Y) that the application wants the widget instance to have.

XmNminAspectY

Specifies the denominator of the minimum aspect ratio (X/Y) that the application wants the widget instance to have.

XmNminHeight

Specifies the minimum height that the application wants the widget instance to have.

XmNminWidth

Specifies the minimum width that the application wants the widget instance to have.

XmNtitle

Specifies the application name to be displayed by the window manager. The default is the icon name, if specified; otherwise, it is the name of the application. When **XtGetValues** is called on this resource, the returned value is a pointer to the actual resource value and should not be freed.

XmNtitleEncoding

Specifies a property type that represents the encoding of the **XmNtitle** string. If a language procedure has been set, the default is None; otherwise, the default is **XA_STRING**. When the widget is realized, if the value is None, the corresponding name is assumed to be in the current locale. The name is passed to **XmbTextListToTextProperty** with an encoding style of **XStdICCTextStyle**. The resulting encoding is **STRING** if the name is fully convertible to **STRING**; otherwise it is **COMPOUND_TEXT**. The values of the encoding resources are not changed; they remain None.

XmNtransient

Specifies a Boolean value that is True if the widget instance is transient, typically a popup on behalf of another widget. The window manager may treat a transient widget's window differently from other windows. For example, a window manager may not iconify a transient window separately from its associated application. Applications and users should not normally alter this resource.

XmNwaitForWm

When True, specifies that the Intrinsics waits the length of time given by the **XmNwmTimeout** resource for the window manager to respond to certain actions before assuming that there is no window manager present. This resource is altered by the Intrinsics as it receives, or fails to receive, responses from the window manager.

XmNwidthInc

Specifies the base for a progression of preferred widths for the window manager to use in sizing the widget. The preferred widths are **XmNbaseWidth** plus integral multiples of **XmNwidthInc**, with a minimum of **XmNminWidth** and a maximum of **XmNmaxWidth**. If an initial value is not supplied for **XmNwidthInc** but is supplied for **XmNheightInc**, the value of **XmNwidthInc** is set to 1 when the widget is realized.

XmNwindowGroup

Specifies the ID of a window with which this widget instance is associated. By convention, this window is the "leader" of a group of windows. A window manager may treat all windows in a group in some way; for example, it may always move or iconify them together.

If no initial value is specified, the value is set to the window of the first realized ancestor widget in the parent hierarchy when the widget is realized. If a value of **XtUnspecifiedWindowGroup** is specified, no window group is set.

XmNwinGravity

Specifies the window gravity for use by the window manager in positioning the widget. If no initial value is specified, the value is set when the widget is realized. If **XmNgeometry** is not NULL, **XmNwinGravity** is set to the window gravity returned by **XWMGeometry**. Otherwise, **XmNwinGravity** is set to **NorthWestGravity**.

XmNwmTimeout

Specifies the length of time that the Intrinsics waits for the window manager to respond to certain actions before assuming that there is no window manager present. The value is in milliseconds and must not be negative.

Inherited Resources

WMShell inherits behavior and resources from the superclasses described in the following tables. For a complete description of each resource, refer to the reference page for that superclass.

Shell Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNallowShellResize	XmCAllowShellResize	Boolean	False	CG
XmNcreatePopupChildProc	XmCCreatePopupChildProc	XtCreatePopupChildProc	NULL	CSG
XmNgeometry	XmCGeometry	String	NULL	CSG
XmNoverrideRedirect	XmCOverrideRedirect	Boolean	False	CSG
XmNpopdownCallback	XmCCallback	XtCallbackList	NULL	C
XmNpopupCallback	XmCCallback	XtCallbackList	NULL	C
XmNsaveUnder	XmCSaveUnder	Boolean	False	CSG
XmNvisual	XmCVisual	Visual *	CopyFromParent	CSG

Composite Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNchildren	XmCReadOnly	WidgetList	NULL	G
XmNinsertPosition	XmCInsertPosition	XtOrderProc	NULL	CSG
XmNnumChildren	XmCReadOnly	Cardinal	0	G

Core Resource Set				
Name	*Class*	*Type*	*Default*	*Access*
XmNaccelerators	XmCAccelerators	XtAccelerators	dynamic	CSG
XmNancestorSensitive	XmCSensitive	Boolean	dynamic	G
XmNbackground	XmCBackground	Pixel	dynamic	CSG
XmNbackgroundPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderColor	XmCBorderColor	Pixel	XtDefaultForeground	CSG
XmNborderPixmap	XmCPixmap	Pixmap	XmUNSPECIFIED_PIXMAP	CSG
XmNborderWidth	XmCBorderWidth	Dimension	0	CSG
XmNcolormap	XmCColormap	Colormap	dynamic	CG
XmNdepth	XmCDepth	int	dynamic	CG
XmNdestroyCallback	XmCCallback	XtCallbackList	NULL	C
XmNheight	XmCHeight	Dimension	dynamic	CSG
XmNinitialResourcesPersistent	XmCInitialResourcesPersistent	Boolean	True	C
XmNmappedWhenManaged	XmCMappedWhenManaged	Boolean	True	CSG
XmNscreen	XmCScreen	Screen *	dynamic	CG
XmNsensitive	XmCSensitive	Boolean	True	CSG
XmNtranslations	XmCTranslations	XtTranslations	dynamic	CSG
XmNwidth	XmCWidth	Dimension	dynamic	CSG
XmNx	XmCPosition	Position	0	CSG
XmNy	XmCPosition	Position	0	CSG

Translations

There are no translations for WMShell.

RELATED

Composite(3) , **Core**(3) , and **Shell**(3) .