Intuition

COLLABORATORS

	TITLE : Intuition	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 10, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Intuition

1.1 Intuition-related Classes for AmigaTalk© 1998:

Described herein are the classes & their methods for manipulating

Amiga-Intuition objects with AmigaTalk.

The class hierarchy is:

Glyph (parent class is Object)

Screen

Window

Menu

MenuItem

SubItem

Gadget

BoolGadget

StrGadget

PropGadget

Color

Requester

Border

BitMap

Painter

Image

AreaPaint

IStruct

Animation

IFF

Font

IText

1.2 Glyph Class:

Class Glyph is an abstract class that serves as a parent class for
all Intuition-related classes. The methods it defines are mostly
identical to those methods found in Object:
= aGlyph
== aGlyph
Return true if aGlyph is the same object as the receiver, else false.
~= aGlyph
~~ aGlyph
Return true if aGlyph is not the same object as the receiver,
else false.
asString
Return a String representing the receiver.
asSymbol
Return a Symbol representing the receiver.
class
Return an object representing the class of the receiver.
copy
Perform a shallowCopy of the receiver.
deepCopy
Perform a complete copy of the receiver.
do: aBlock
Perform aBlock on each element of the receiver collection.
error: aString
Display aString as an error message, then return nil.
first
Return the first item in sequence, which is by default the receiver.
isKindOf: aClass
Return true if class of the receiver is the same as the argument
aClass.
isMemberOf: aClass
Return true if receiver is an instance of the argument class.
isNil
Test whether the receiver is nil.
next
Return the next item in sequence, which is nil by default.
notNil
Test if receiver is not object nil.

print

Print the receiver class name as a String.

printString

Return the receiver class asString.

respondsTo: message

Return true if receiver will respond to the indicated message.

shallowCopy

Return the receiver.

addressOf

Return an Integer that represents the memory address of the receiver.

In class Glyph, this method simply returns an error message.

glyphType

Return the receiver class asString. In class Glyph, this

method simply returns an error message.

isDisplayed

Return true if the object is being displayed, else return false.

For class Glyph, this method simply returns as error message.

1.3 Screen Class:

Class Screen allows the AmigaTalk system to manipulate Amiga screens.

The Methods are:

new: newScreenTitle

Create a new Screen Object with the title of newScreenTitle

and the default screenModeID of DEFAULT_MONITOR_ID.

setScreenModeID: newScreenModeID

Set the Screen Object's ModeID to newScreenModeID.

NOTE: The order of methods for making a Screen is as follows:

scr <- Screen new: 'Test Screen'

scr setScreenModeID: 16r40D20001

scr open

getScreenModeID

Return the current screenModeID.

open

Open a Screen Object with the previously set title & modeID.

close

Close the Screen Object.

pullScreenUp: numLines

Move a screen up by numLines (see intuition.library

function MoveScreen()). numLines has to be <= 0.

pushScreenDown: numLines

Move a screen down by numLines (see intuition.library

function MoveScreen()). numLines has to be >= 0.

redrawScreen

Re-draw the Screen Object (see intuition.library functions

MakeScreen() & RethinkDisplay()).

reOpenScreen

Re-open the screen with any new parameters that were changed by the

user.

displayBeep

Call the intuition function DisplayBeep() for the given screen.

screenToBack

Place the given screen behind all other open screens (see

intuition.library function ScreenToBack()).

screenToFront

Place the given screen in front of all other open screens (see

intuition.library function ScreenToFront()).

turnOffTitle

Blank out the screenTitle (see intuition.library function

ShowTitle()).

showTitle

Enable the screenTitle display (see intuition.library function

ShowTitle()).

NOTE: All of the set Parameter methods won’t take effect until a

call to reOpenScreen is given.

In general, it’s best to perform all of the set parameter

methods before you open the screen, then a call to reOpenScreen

isn’t necessary.

setOrigin: aPoint

Set the starting point of the screen to the given point aPoint.

setScreenSize: sizePoint

Set the width & height of the Screen to (sizePoint x) &

(sizePoint y) respectively.

setScreenPens: pensPoint

Set the Foreground & Background pens to (pensPoint x) &

(pensPoint y) respectively.

setTitle: newTitle

Change the title of the Screen to newTitle.

setDepth: newDepth

Change the number of bit-planes that the screen will use. The depth of a screen determines how many colors it has.

setFont: newFontName

Change the Font used to render text in the screen.

setBitMap: newBitMapName

Change the bitmap of the given screen.

setType: newType

Change the Type of the given screen (see NewScreen structure for details of this value).

NOTE: Due to the plethora of graphic accelerator cards & different types of monitors, this method will likely be superceded in the future.

setViewMode: newViewMode

Change the ViewMode of the given screen (see NewScreen structure for details of this value).

NOTE: Due to the plethora of graphic accelerator cards & different types of monitors, this method will likely be superceded in the future.

getOrigin

Return the current value of the LeftEdge & TopEdge of the screen as a Point .

getScreenPens

Return the current value of the foreground & background pens for the screen as a Point .

getFlags

Return the current value of the Flags for the screen.

getType

Return the current value of the Type of the screen.

getViewMode

Return the current value of the ViewMode of the screen.

getTitle

Return the current value of the Title of the screen.

getDepth

Return the current value of the Depth of the screen.

getFontName

Return the name of the current Font for the screen.

getBitMapName

Return the name of the current bitmap for the screen.

1.4 Window Class:

Class Window allows the AmigaTalk system to manipulate Amiga windows.

The Methods are:

new: newWindowTitle

Create a new Window Object & set the title to newWindowTitle.

openOnScreen: screenName

Open a new window on the given screen.

close

Close the given window.

setPointer: spritePtr size: sizePt offset: offPt

Change the mouse pointer to the given sprite spritePtr.

sizePt & offPt are Points .

addGadget: gadgetName type: gadgetType

Add a gadget of the given type to the given window.

Valid gadgetType values:

Boolean = 0

String = 1

Proportional = 2

setFirstGadget: newGadget

Change the FirstGadget to newGadget.

NOTE: Make sure that the newGadget is chained to all the other

gadgets you want for the window (See setNextGadgetName methods

for Boolean , String , & Proportional).

refreshGadgets

Execute a call to RefreshGadgets() (see intuition.library).

removeGadget: gadgetName type: gadgetType

Delete a gadget from the Window Object.

reportMouse: boolvalue

Turn reportMouse events on or off.

getMouseCoords

Return the current x & y-coordinates of the mouse position as a

Point .

printIText: iTextName at: aPoint

Display the given IText structure in the Window at the

given Point .

reOpenWindow

Re-open a window so that any changed parameters will take effect.

handleIntuition

Wait for the user to select a Gadget or MenuItem from the window.
NOTE: Only IDCMP_GADGETUP & IDCMP_MENUPICK events are currently
recognized. This method will return the name of the first
Gadget or MenuItem selected by the user.
addDMRequest: dMRequesterName
Execute a SetDMRequest() function call for the Window
(see intuition.library).
removeDMRequest
Execute a ClearDMRequest() function call for the Window
(see intuition.library).
windowToBack
Place the Window behind all other windows.
windowToFront
Place the Window in front of all other windows.
showRequester: requesterName
Display the given Requester to the user.
addMenuStrip: menuName
Display the given menu(s) in the Window.
removeMenuStrip
Remove the current menu strip from the Window.
moveWindow: deltaPoint
Move the Window to the new point of origin.
infoReq: msg title: t
Display an information Requester to the user.
yesNoReq: msg title: t
Obtain a yes or no response from the user.
getUserChoice: msg title: t choices: bstr
Obtain a choice from the user from the given bstr (Button strings).
Example: 'YES|NO|MAYBE'
setWindowSize: sizePoint
Change the size of the Window to the given values (sizePoint x) &
(sizePoint y).
getOrigin
Return the current LeftEdge & TopEdge values for the Window
as a Point .
getWindowSize
Return the current Width & Height values for the Window
as a Point .
getWindowPens

Return the current foreground & background pen values for the Window
as a Point .
getFlags: windowTitle
Return the current Flags value for the given window.
getIDCMPFlags: windowTitle
Return the current IDCMPFlags value for the given window.
getTitle: windowTitle
Return the current Title value for the given window.
This a silly function since you have to know the title in order
to use it!
changeTitle: newTitle
Change the title of the Window to newTitle.
getScreenTitle
Get the title of the screen that the given window is attached to.
beginRefresh
Execute a call to BeginRefresh() for the Window.
(see intuition.library)
endRefresh
Execute a call to EndRefresh() for the Window.
(see intuition.library)
remakeDisplay
Execute a call to RemakeDisplay() for the Window.
(see intuition.library)
rethinkDisplay
Execute a call to RethinkDisplay() for the Window.
(see intuition.library)
NOTE: All of the set Parameter methods won't take effect until a
call to reOpenWindow is given.
In general, it's best to perform all of the set parameter
methods before you open the window, then a call to
reOpenWindow isn't necessary.
setWindowOrigin: aPoint
Change the LeftEdge & TopEdge of the given window to the
new values.
setWindowPens: pensPoint
Change the foreground & background pens of the Window to the new
values (foreground <- pensPoint x) & (background <- pensPoint y).
setFlags: newFlags
Change the Flags of the Window to the new value.

setIDCMPFlags: windowTitle to: newIDCMP

Change the IDCMPFlags of the Window to the new value.

setMinSize: newMinPoint

Change the MinWidth & MinHeight of the Window to the new values.

setMaxSize: newMaxPoint

Change the MaxWidth & MaxHeight of the Window to the new values.

getReqCount

Return a count of the Requesters (reqcount)for the Window.

getPointerSize

Return the width & height of the mouse pointer for the Window as a Point .

setCheckMark: newCheckMark

Change the CheckMark Image of the Window to the new value specified by 'newCheckMark', which is an Image name.

getWindowOffset

Return the current x & y-offset coordinates for the Window as a Point .

setBitMap: newBitMap

Change the BitMap of the Window to the new value specified by 'newBitMap', which is a BitMap name.

changeWindowSize: deltaPoint

Ask Intuition to size the Window the specified amounts.

NOTE: This method is different from setWindowSize:

1.5 Menu Class:

Class Menu allows the AmigaTalk system to manipulate Amiga menus.

Actual display of any Menus are taken care of in the Window Class by setMenuStrip & removeMenuStrip.

The Methods for the Menu Class are:

new: newMenuName

Add a menu to the AmigaTalk internal system list.

remove

Remove a Menu from the AmigaTalk internal system list.

registerTo: windowTitle

Inform the AmigaTalk internal system which window is the parent of

the Menu.

getStartPoint

Return the LeftEdge & TopEdge of the Menu.

getMenuSize

Return the Width & Height of the Menu.

getFlags

Return the Flags of the Menu.

getNextMenu

Return the name of the NextMenu for the Menu.

getFirstItem

Return the name of the FirstItem for the Menu.

getMenuName

Return the name of the Menu. This is a silly function, since

you have to know the name of the menu to use it!

setStartPoint: newPoint

Set the LeftEdge & TopEdge of the Menu to the given values.

setMenuSize: sizePoint

Set the Width & Height of the Menu to the given values.

setFlags: newFlags

Set the Flags of the Menu to the given value.

setNextMenu: newNextMenu

Set the NextMenu of the Menu to the given value.

setFirstItem: newFirstItem

Set the FirstItem of the Menu to the given value.

setMenuName: newMenuName

Set the name of the Menu to the new name given.

See Also MenuItems , SubItems

1.6 MenuItem Class:

Class MenuItem allows the AmigaTalk system to manipulate Amiga

MenuItems. Actual display of any Menus are taken care of in the

Window Class by setMenuStrip & removeMenuStrip.

The Methods for the MenuItem Class are:

new: newMenuItemName

Add a MenuItem to the AmigaTalk internal system list.

remove

Remove a MenuItem from the AmigaTalk internal system list.

getStartPoint

Return the LeftEdge & TopEdge of the MenuItem.

getItemSize

Return the Width & Height of the MenuItem.

setStartPoint: newPoint

Set the LeftEdge & TopEdge of the MenuItem to the given values.

setItemSize: sizePoint

Set the Width & Height of the MenuItem to the given values.

getFlags

Return the Flags of the MenuItem.

setFlags: newFlags

Set the Flags of the MenuItem to the given value.

getMutualExclude

Return the MutualExclude value of the MenuItem.

setMutualExclude: newMutualExclude

Set the MutualExclude of the MenuItem to the given value.

getCommand

Return the Command (menu key-equivalent) value of the MenuItem.

setCommand: newCommand

Set the Command (menu key-equivalent) of the MenuItem to the given value.

getNextItem

Return the name of the NextItem from the MenuItem.

setNextItem: newNextItem

Set the NextItem of the MenuItem to the given value.

setItemFill: newItemFill

Set the ItemFill (IText or Image name) of the MenuItem.

setSelectFill: newSelectFill

Set the SelectFill (IText or Image name) of the MenuItem.

setSubItem: newSubItem

Set the SubItem of the MenuItem to the given value.

getSubItem

Return the name of the first SubItem attached to the MenuItem.

getItemFill

Return the name of the ItemFill (either IText or Image) from the MenuItem.

getSelectFill

Return the name of the SelectFill (either IText or Image) from the MenuItem.

See Also Menus , SubItems

1.7 SubItem Class:

Class SubItem allows the AmigaTalk system to manipulate Amiga
Menu SubItems. Actual display of any Menus are taken care of in the
Window Class by setMenuStrip & removeMenuStrip.
The Methods for the SubItem Class are:
new: newSubItemName
Add a SubItem to the AmigaTalk internal system list.
remove
Remove a SubItem from the AmigaTalk internal system list.
getStartPoint
Return the LeftEdge & TopEdge of the SubItem.
getSubSize
Return the Width & Height of the SubItem.
setStartPoint: newPoint
Set the LeftEdge & TopEdge of the SubItem to the given values.
setSubSize: sizePoint
Set the Width & Height of the SubItem to the given values.
getFlags
Return the Flags of the SubItem.
setFlags: newFlags
Set the Flags of the SubItem to the given value.
getMutualExclude
Return the MutualExclude value of the SubItem.
setMutualExclude: newMutualExclude
Set the MutualExclude of the SubItem to the given value.
getCommand
Return the Command (menu key-equivalent) value of the
SubItem.
setCommand: newCommand
Set the Command (menu key-equivalent) of the SubItem
to the given value.
getNextItem
Return the name of the NextItem from the SubItem.
setNextItem: newNextItem
Set the NextItem of the SubItem to the given value.
setItemFill: newItemFill
Set the ItemFill (IText or Image name) of the SubItem.
setSelectFill: newSelectFill

Set the SelectFill (IText or Image name) of the SubItem.

getItemFill

Return the name of the ItemFill (either IText or Image) from

the SubItem.

getSelectFill

Return the name of the SelectFill (either IText or Image)

from the SubItem.

See Also Menus , MenuItems

1.8 Gadget Class:

Class Gadget is an abstract class that serves as a parent class for

all Gadget-related classes. The methods it defines are only useful

for system-wide purposes:

gadgetTypeIs: gadgetName

Return the type of the gadget given by gadgetName. The return

values are:

BOOLEAN = 1

PORPORTIONAL = 3

STRING = 4

new: newGadgetName

Create a new Gadget to the newGadgetName & the default

type of BoolGadget.

SubClasses: BoolGadget

StrGadget

PropGadget

1.9 Boolean Gadget Class:

Class BoolGadget allows the AmigaTalk system to man-

ipulate Amiga Boolean Gadgets. The Methods are:

new: newGadgetName

Add a BoolGadget to the AmigaTalk system. This method allocates

an internal memory structure to the AmigaTalk system.

remove

Remove a BoolGadget from the AmigaTalk system.

registerTo: windowTitle

Set the name of the gadget's parent to the windowTitle.

setStartPoint: newPoint

Set the origin of the gadget to the given point value.

setGadgetSizeTo: sizePoint

Set the size of the gadget to the given width & height.

getLeftEdge

Return the LeftEdge value of the BoolGadget.

getTopEdge

Return the TopEdge value of the BoolGadget.

getWidth

Return the Width value of the BoolGadget.

getHeight

Return the Height value of the BoolGadget.

getFlags

Return the Flags value of the BoolGadget.

getActivation

Return the Activation value of the BoolGadget.

getGadgetType

Return the Type of the BoolGadget.

NOTE: only needed because of GZZGADGET & REQGADGET type flags.

getGadgetID

Return the GadgetID number for the BoolGadget.

getNextGadgetName

Return the name of the NextGadget for the BoolGadget.

getITextName

Return the name of the IText attached to the BoolGadget.

getRenderName

Return the name of the gadget rendering (IText or Image) for the

BoolGadget.

getSelectName

Return the name of the gadget selection rendering (IText or

Image) for the BoolGadget.

setFlags: newFlags

Set the gadget Flags to the new value(s).

setActivation: newActivation

Set the gadget Activation to the new value.

setGadgetType: newGadgetType

Set the gadget Type to the new value.

NOTE: only needed because of GZZGADGET & REQGADGET type flags.

setGadgetID: newGadgetID

Set the GadgetID to the new value.

setNextGadgetName: newNextGadgetName

Set the NextGadget to the Gadget attached to newNextGadgetName.

setITextName: newITextName

Set the IText to the new value.

setRenderName: newRenderName

Set the BoolGadget rendering to the name of the IText or Image

supplied.

setSelectName: newSelectName

Set the gadget selection rendering to the name of the IText

or Image supplied.

1.10 String Gadget Class:

Class StrGadget allows the AmigaTalk system to man-

ipulate Amiga String Gadgets. The Methods are:

new: newGadgetName

Add a StrGadget to the AmigaTalk system. This method allocates

an internal memory structure to the AmigaTalk system.

remove

Remove a StrGadget from the AmigaTalk system.

registerTo: windowTitle

Set the name of the StrGadget's parent to the windowTitle.

setStartPoint: newPoint

Set the origin of the StrGadget to the given point value.

setGadgetSize: sizePoint

Set the size of the StrGadget to the given point value.

changeBufferSize: newSize

Change the internal buffer size for the StrGadget.

getBufferSize

Return the size of the StrGadget buffer (in bytes).

getLeftEdge

Return the LeftEdge value of the StrGadget.

getTopEdge

Return the TopEdge value of the StrGadget.

getWidth

Return the Width value of the StrGadget.

getHeight

Return the Height value of the StrGadget.

getFlags

Return the Flags value of the StrGadget.

getActivation

Return the Activation value of the StrGadget.

getGadgetType

Return the Type of the StrGadget.

NOTE: only needed because of GZZGADGET & REQGADGET type flags.

getGadgetID

Return the GadgetID number for the StrGadget.

getNextGadgetName

Return the name of the NextGadget for the StrGadget.

getITextName

Return the name of the IText attached to the StrGadget.

getRenderName

Return the name of the gadget rendering (IText or Image) for the

StrGadget.

getSelectName

Return the name of the gadget selection rendering (IText

or Image) for the StrGadget.

setFlags: newFlags

Set the StrGadget Flags to the new value(s).

setActivation: newActivation

Set the StrGadget Activation to the new value.

setGadgetType: newGadgetType

Set the StrGadget Type to the new value.

NOTE: only needed because of GZZGADGET & REQGADGET type flags.

setGadgetID: newGadgetID

Set the GadgetID to the new value.

setNextGadgetName: newNextGadgetName

Set the NextGadget to the Gadget attached to newNextGadgetName.

setITextName: newITextName

Set the @{ "IText " LINK "ITextClass"} to the new value.

setRenderName: newRenderName

Set the StrGadget rendering to the name of the IText or Image

supplied.

setSelectName: newSelectName

Set the StrGadget selection rendering to the name of the IText

or Image supplied.

1.11 Proportional Gadget Class:

Class PropGadget allows the AmigaTalk system to man-
ipulate Amiga Proportional Gadgets. The Methods are:
new: gadgetName
Add a PropGadget to the AmigaTalk system. This method allocates
an internal memory structure to the AmigaTalk system.
remove
Remove a PropGadget from the AmigaTalk system.
registerTo: windowTitle
Set the name of the PropGadget's parent to the windowTitle.
setStartPoint: newPoint
Set the origin of the PropGadget to the given point value.
setGadgetSize: sizePoint
Set the size of the PropGadget to the given width & height.
modifyProps: newFlags hPot: hp vPot: vp hBody: hb vBody: vb
windowName: windowTitle
Change the given porportional values for the PropGadget.
setProps: gadgetName flags: newFlags hPot: hp vPot: vp
hBody: hb vBody: vb
Initialize the porportional gadget values.
getPropFlags
Return the PropFlags value for the PropGadget.
getHPot
Return the HPot value of the PropGadget.
getVPot
Return the VPot value of the PropGadget.
getHBody
Return the HBody value of the PropGadget.
getVBody
Return the VBody value of the PropGadget.
getLeftEdge
Return the LeftEdge value of the PropGadget.
getTopEdge
Return the TopEdge value of the PropGadget.
getWidth
Return the Width value of the PropGadget.
getHeight
Return the Height value of the PropGadget.

getFlags

Return the Flags value of the PropGadget.

getActivation

Return the Activation value of the PropGadget.

getGadgetType

Return the Type of the PropGadget.

NOTE: only needed because of GZZGADGET & REQGADGET type flags.

getGadgetID

Return the GadgetID number for the PropGadget.

getNextGadgetName

Return the name of the NextGadget for the PropGadget.

getITextName

Return the name of the @{ " IText " LINK "ITextClass"} attached to the PropGadget.

getRenderName

Return the name of the gadget rendering (IText or Image) for the

PropGadget.

getSelectName

Return the name of the gadget selection rendering (IText

or Image) for the PropGadget.

setFlags: newFlags

Set the gadget Flags to the new value(s).

setActivation: newActivation

Set the PropGadget Activation to the new value.

setGadgetType: newGadgetType

Set the PropGadget Type to the new value.

NOTE: only needed because of GZZGADGET & REQGADGET type flags.

setGadgetID: newGadgetID

Set the GadgetID to the new value.

setNextGadgetName: newNextGadgetName

Set the NextGadget to the Gadget attached to newNextGadgetName.

setITextName: newITextName

Set the IText to the new value.

setRenderName: newRenderName

Set the PropGadget rendering to the name of the IText or Image

supplied.

setSelectName: newSelectName

Set the PropGadget selection rendering to the name of the IText

or Image supplied.

1.12 Colors Class:

Class Colors allows the AmigaTalk system to manipulate Amiga Colors.

The Methods are:

make: colorMapName size: numColors

Allocate a new ColorMap.

dispose

Free the given ColorMap from the system.

loadColors: c from: colorMapFileName

Load the amount of color registers c with the values given in the

colorMapFileName file.

getColor: sourceType from: sourceName which: n

Return an RGB representation from the given source (window = 1 or

colormap) for the nth register.

setWindowColorReg: n red: r green: g blue: b

Set the color register n to the RGB values supplied.

setMapValue: sourceType from: source num: n red: r green: g blue: b

Set the color register n for the given source (window = 1 or

colormap) to the RGB values supplied.

copyMap: source to: dest sourceType: type

Copy a ColorMap from the source of sourceType (window = 1 or colormap)

to the destination (ColorMap).

saveColorsTo: colorMapFileName

Save the color register values to the given filename.

1.13 Requester Class:

Class Requester implements control of Amiga Requesters for AmigaTalk,

except for displaying them, which is done inside the Window class.

The methods are:

initialize: requesterName withArray: reqValues

Initialize a Requester for the AmigaTalk system.

reqValues is an Array with the following fields:

LeftEdge, TopEdge,

Width, Height,

RelLeft, RelTop,

ReqGadget, ReqBorder, ReqText,

Flags BackFill, ImageBMap

new: requesterName

Register a Requester with the AmigaTalk system.

remove

Remove a Requester from the AmigaTalk system.

getStartPoint

Return the LeftEdge & TopEdge of the receiver.

getReqSize

Return the Width & Height of the receiver.

getRelativePoint

Return the RelLeft & RelTop variable of the receiver.

getFlags

Return the Flags variable of the receiver.

getBackFill

Return the background pen number of the receiver.

getReqText

Return the name of the IText attached to the receiver.

getReqGadget

Return the name of the first Gadget attached to the receiver.

getReqBorder

Return the name of the Border attached to the receiver.

getReqBitMap

Return the name of the BitMap attached to the receiver.

setStartPoint: newPoint

Change the LeftEdge & TopEdge of the receiver to

the supplied values.

setReqSize: sizePoint

Change the Width & Height of the receiver to the

values supplied.

setRelativePoint: newRelPoint

Change the RelLeft & RelTop of the receiver to the values

given.

NOTE: You should also add the POINTREL value to the Flags of the

receiver in order to use this feature.

receiver in order to use this feature.

setFlags: newFlags

Change the Flags of the receiver to newFlags.

setBackFill: newBackFill

Change the background pen number to newBackFill for the

receiver.

setReqText: newReqText

Change the IText attached to the reciever to newReqText.

setReqBorder: newReqBorder

Change the Border attached to the receiver to newReqBorder.

setReqGadget: newReqGadget

Change the Gadget attached to the receiver to newReqGadget.

setReqBitMap: newReqBMap

Change the BitMap attached to the receiver to newReqBMap.

1.14 Alert Class:

Class Alert implements control of Amiga Alerts for the AmigaTalk system.

The message string that the user supplies will be truncated at 128

characters & the alert number will be prepended to it.

new: newAlertName

Add an Alert to the AmigaTalk system.

remove

Remove an Alert from the AmigaTalk system.

getAlertNumber

Return the alert number attached to the receiver.

getAlertHeight

Return the height of the receiver.

getAlertMessage

Return the alert String attached to the receiver.

setAlertNumber: num

Change the alert number of the receiver to num.

setAlertHeight: height

Change the alert height of the receiver to height.

setAlertMessage: newMsg

Change the alert String of the receiver to newMsg.

displayAlert

Display the receiver to the user.

1.15 Border Class:

The Class Border (in this implementation) is an abstract class that

normally is attached to other objects, such as Gadgets or Requesters.

This is why there is no method for actually drawing borders into

Windows in this Class. See drawPolygon in the Class Painter

if you need to draw a Border in a Window .

SubClasses: Line

Triangle

Rectangle

new: newBorderName

Add the receiver to the AmigaTalk system. Initialize the new

Border Object as follows:

name <- newBorderName

nextBorderName <- nil

leftEdge <- 0

topEdge <- 0

frontPen <- 1

backPen <- 0

drawMode <- 1

count <- 2

remove

Remove the receiver from the AmigaTalk system.

registerTo: windowTitle

Set the parent name of the receiver to windowTitle.

getLeftEdge

Return the LeftEdge of the receiver.

getTopEdge

Return the TopEdge of the receiver.

getFrontPen

Return the foreground pen number of the receiver.

getBackPen

Return the background pen number of the receiver.

getDrawMode

Return the drawing mode of the receiver. Currently known values

are:

JAM1 = 0

JAM2 = 1

COMPLEMENT = 2

INVERSEVID = 4

getCount

Return the number of points in the receiver.

getNextBorderName

Return the name of the next Border attached to the receiver.

setStartPoint: sPoint

Change the starting point of the receiver to sPoint.

This method sets LeftEdge & TopEdge.

setFrontPen: newFrontPen

Change the foreground pen number of the receiver.

setBackPen: newBackPen

Change the background pen number of the receiver.

setDrawMode: newDrawMode

Change the drawing mode of the receiver. Currently known values

are:

JAM1 = 0

JAM2 = 1

COMPLEMENT = 2

INVERSEVID = 4

setCount: newCount

Change the number of points in the receiver.

setNextBorderName: newBorder

Change the name of the next Border attached to the receiver.

setBorderPoint: thePt to: newPoint

Change the value of a point in the receiver.

1.16 Line Class (Border sub-class):

makeLine: lineName from: fPoint to: tPoint

Add a Border with two points & register it with the AmigaTalk system.

1.17 Triangle Class (Border sub-class):

makeTriangle: triangleName vert1: v1Point vert2: v2Point vert3: v3Point

Add a Border with 4 points & register it with the AmigaTalk system.

1.18 Rectangle Class (Border sub-class):

makeRectangle: rectangleName from: fPoint to: tPoint

Add a Border with 5 points & register it with the AmigaTalk system.

1.19 Font Class:

Class Font implements control of Amiga Fonts.

The methods for Font are:

new: newFontName

Add a Font to the AmigaTalk system.

remove

Remove an Font from the AmigaTalk system.

getName

Return the name of the Font.

getYSize

Return the Height of the Font (in pixels - ta_YSize).

getStyle

Return the style (ta_Style) of the Font such as PLAIN, BOLD,

UNDERLINED or ITALIC, etc.

getFlags

Return the Flags of the receiver (ta_Flags).

setName: newName

Change the name of the Font to newName (ta_Name <- newName).

This method is probably not really needed.

setYSize: newYSize

Set the Height of the Font (ta_YSize <- newYSize).

setStyle: newStyle

Change the style of the Font (ta_Style <- newStyle).

setFlags: newFlags

Change the Flags of the Font (ta_Flags <- newFlags).

See Also IText Class

1.20 IText Class:

Class IText implements control of Amiga IntuiText except for actually

displaying it, which is in the Window class.

The methods for IText are:

new: newITextName

Add an IText (IntuiText) to the AmigaTalk system.

remove

Remove an IText from the AmigaTalk system.

registerTo: windowTitle

Set the parent name of the receiver to windowTitle.

getFrontPen

Return the foreground pen number of the receiver.

getBackPen

Return the background pen number of the receiver.

getDrawMode

Return the drawing mode of the receiver. Currently known values are:

JAM1 = 0

JAM2 = 1

COMPLEMENT = 2

INVERSEVID = 4

getLeftEdge

Return the LeftEdge of the receiver.

getTopEdge

Return the TopEdge of the receiver.

getFontName

Return the name of the rendering font of the receiver.

getIText

Return the String that the the receiver will display.

getNextText

Return the name of the next IntuiText of the receiver.

getTextLength

Return the length (in pixels) of the text of the receiver.

setFrontPen: newFrontPen

Change the foreground pen of the receiver.

setBackPen: newBackPen

Change the background pen of the receiver.

setDrawMode: newDrawMode

Change the drawing mode of the receiver. Currently known values are:

JAM1 = 0

JAM2 = 1

COMPLEMENT = 2

INVERSEVID = 4

setLeftEdge: newLeftEdge

Change the LeftEdge of the receiver.

setTopEdge: newTopEdge

Change the TopEdge of the receiver.

setFontName: newFontName

Change the name of the rendering font of the receiver.

setIText: iTextName to: newIText

Change the text String of the receiver.

setNextText: newNextText

Change the name of the next IText of the receiver.

See Also Font Class

1.21 BitMap Class:

Class BitMap implements control of Amiga BitMaps for the Amigatalk
system. Valid values for Flags are:
BMB_CLEAR = 0
BMB_DISPLAYABLE = 1
BMB_INTERLEAVED = 2
BMB_STANDARD = 3
BMB_MINPLANES = 4
new: newBitMapName
Add an instance of a BitMap to the AmigaTalk system.
Default values are as follows:
name <- newBitMapName
width <- 1
height <- 1
depth <- 1
remove
Remove a BitMap from the AmigaTalk system.
getBitMapWidth
Return the width (in pixels) of the receiver.
getBitMapHeight
Return the height (in pixels) of the receiver.
getBitMapFlags
Return the Flags of the receiver.
getBitMapDepth
Return the depth (number of bitplanes) of the receiver.
setBitMapWidth: newWidth
Change the width (in pixels) of the receiver.
setBitMapHeight: newHeight
Change the height (in pixels) of the receiver.
setBitMapFlags: newFlags
Change the Flags of the receiver.
setBitMapDepth: newDepth
Change the depth (number of bitplanes) of the receiver.
readBitMapFile: bitMapFile
Load a BitMap from the given file. The file format is unique to
AmigaTalk.
writeBitMapFile: bitMapFile
Save a BitMap to the given file. The file format is unique to
AmigaTalk.

1.22 Painter Class:

Class Painter allows the user to draw simple graphics onto
Windows . The methods are:
new: newOwnerWindow
Initialize a new instance of Painter by setting the Window that will
be used by the Painter.
setAPen: pen
Change the foreground pen number of the given Window.
setBPen: pen
Change the background pen number of the given Window.
setOPen: pen
Change the outline pen number of the given Window.
setDrawMode: mode
Change the drawing mode of the given Window. Currently known
values are:
JAM1 = 0
JAM2 = 1
COMPLEMENT = 2
INVERSEVID = 4
movePenTo: newPoint
Change the drawing Point (without drawing anything!) on the given
Window.
drawTo: aPoint
Draw a line from the current Window location to the given point.
drawLineFrom: fPoint to: tPoint
Draw a line from fPoint to tPoint.
drawBoxFrom: fPoint to: tPoint
Draw a box from fPoint to tPoint.
drawCircle: cPoint radius: r
Draw a circle of radius r with center point cPoint.
drawEllipse: cPoint minaxis: a maxaxis: b
Draw an ellipse at center point cPoint.
drawPolygon: borderName
Draw a Border Object.
drawPixelAt: aPoint
Draw a single pixel at the point given.
drawText: iTextName at: aPoint
Draw an IText at the point given.
NOTE: You should set the foreground pen, background pen & the
drawing mode before using this method!

1.23 Image Class:

Class Image allows the user to draw Amiga Images

onto Windows . The methods are:

new: newImageName

Add an Image to the AmigaTalk system.

registerTo: windowTitle

Set the parent name of the receiver Image to windowTitle.

removeImage

Remove an Image from the AmigaTalk system.

drawImageAt: aPoint

Display an Image on the given Window at the given

(aPoint x, aPoint y) coordinates.

setImageData: imageFile

Load Image data from the given file.

getStartPoint

Return the LeftEdge & TopEdge of the receiver.

getImageSize

Return the Width & Height of the receiver.

getImageDepth

Return the Depth of the receiver.

getImagePlanePick

Return the PlanePick variable of the receiver.

getImagePlaneOnOff

Return the PlaneOnOff variable of the receiver.

getNextImage

Return the name of the next Image of the receiver.

setOrigin: aPoint

Set the LeftEdge & TopEdge of the receiver.

setExtent: sizePoint

Set the Width & Height of the receiver.

setImageDepth: newDepth

Set the Depth of the receiver.

setImagePlanePick: pp

Set the PlanePick variable of the receiver.

setImagePlaneOnOff: po

Set the PlaneOnOff variable of the receiver.

setNextImage: nextImage

Set the name of the next Image of the receiver.

1.24 AreaPaint Class:

Not Implemented yet!

Class Area

1.25 IStruct Class:

Not Implemented yet!

Class IStruct allows access to various Amiga OS structures

used by Intuition. These are:

ViewPorts

Views

PlayFields

RastPorts

RasInfo

Layers

Blitter

Copper

1.26 Animation Class:

Not Implemented yet!

Class Anim

1.27 IFF Class:

Not Implemented yet!

Class IFF

CloseIFF

OpenIFF

FreeIFF

AllocIFF

InitIFF

InitIFFasDOS

InitIFFasClip

OpenClipboard

ParseIFF

ReadChunkBytes

ReadChunkRecords

StopChunk

CurrentChunk

PropChunk

FindProp

CollectionChunk

FindCollection

StopOnExit

EntryHandler

ExitHandler

PushChunk

PopChunk

ParentChunk

AllocLocalItem

LocalItemData

StoreLocalItem

StoreItemInContext

FindPropContext

FindLocalItem

FreeLocalItem

SetLocalItemPurge