

MBS MacControls Plugin Documentation

Christian Schmitz

March 10, 2024

0.1 Introduction

This is the PDF version of the documentation for the Xojo Plug-in from Monkeybread Software Germany.
Plugin part: MBS MacControls Plugin

0.2 Content

- 1 List of all topics 3
- 2 List of all classes 141
- 3 List of all controls 145
- 4 List of all global methods 147
- 5 All items in this plugin 149
- 19 List of Questions in the FAQ 989
- 20 The FAQ 999

Chapter 1

List of Topics

• 5 Addressbook	149
– 5.1.1 control ABPeoplePickerViewControlMBS	149
* 5.1.3 View as ABPeoplePickerViewMBS	149
* 5.1.5 BoundsChanged	150
* 5.1.6 Close	150
* 5.1.7 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	150
* 5.1.8 ContextualMenuItemAction(hitItem as MenuItem) as Boolean	150
* 5.1.9 didCloseContextualMenu(menu as NSMenuItemMBS, NSEvent as NSEventMBS)	150
* 5.1.10 DisplayedPropertyDidChange	150
* 5.1.11 EnableMenuItems	151
* 5.1.12 FrameChanged	151
* 5.1.13 GotFocus	151
* 5.1.14 GroupDoubleClick	151
* 5.1.15 GroupSelectionDidChange	151
* 5.1.16 LostFocus	151
* 5.1.17MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	152
* 5.1.18 MouseDrag(x as Integer, y as Integer)	152
* 5.1.19 MouseUp(x as Integer, y as Integer)	152
* 5.1.20 NameDoubleClick	152
* 5.1.21 NameSelectionDidChange	153
* 5.1.22 Open	153
* 5.1.23 ScaleFactorChanged(NewFactor as Double)	153
* 5.1.24 ValueSelectionDidChange	153
* 5.1.25 willShowContextualMenu(menu as NSMenuItemMBS, NSEvent as NSEventMBS)	153
– 5.2.1 class ABPeoplePickerViewMBS	154
* 5.2.3 ABPeoplePickerDisplayedPropertyDidChangeNotification as string	154
* 5.2.4 ABPeoplePickerGroupSelectionDidChangeNotification as string	154

* 5.2.5 ABPeoplePickerNameSelectionDidChangeNotification as string	155
* 5.2.6 ABPeoplePickerValueSelectionDidChangeNotification as string	155
* 5.2.7 addProperty(PropertyName as string)	155
* 5.2.8 clearSearchField	155
* 5.2.9 Constructor	155
* 5.2.10 Constructor(Handle as Integer)	156
* 5.2.11 Constructor(left as Double, top as Double, width as Double, height as Double)	156
* 5.2.12 deselectAll	156
* 5.2.13 deselectGroup(group as ABGroupMBS)	156
* 5.2.14 deselectIdentifier(identifier as string, person as ABPersonMBS)	156
* 5.2.15 deselectRecord(record as ABRecordMBS)	157
* 5.2.16 editInAddressBook	157
* 5.2.17 properties as string()	157
* 5.2.18 removeProperty(PropertyName as string)	157
* 5.2.19 selectedGroups as ABGroupMBS()	157
* 5.2.20 selectedIdentifiersForPerson(person as ABPersonMBS) as string()	157
* 5.2.21 selectedRecords as ABRecordMBS()	158
* 5.2.22 selectedValues as Variant()	158
* 5.2.23 selectGroup(group as ABGroupMBS, byExtendingSelection as boolean)	158
* 5.2.24 selectIdentifier(identifier as string, person as ABPersonMBS, byExtendingSelection as boolean)	158
* 5.2.25 selectInAddressBook	158
* 5.2.26 selectRecord(group as ABRecordMBS, byExtendingSelection as boolean)	159
* 5.2.28 accessoryView as NSViewMBS	159
* 5.2.29 allowsGroupSelection as boolean	159
* 5.2.30 allowsMultipleSelection as boolean	159
* 5.2.31 autosaveName as string	160
* 5.2.32 displayedProperty as string	160
* 5.2.33 valueSelectionBehavior as Integer	160
* 5.2.34 columnTitleForProperty(propertyName as string) as string	160
* 5.2.36 DisplayedPropertyDidChange	160
* 5.2.37 GroupDoubleClick	161
* 5.2.38 GroupSelectionDidChange	161
* 5.2.39 NameDoubleClick	161
* 5.2.40 NameSelectionDidChange	161
* 5.2.41 ValueSelectionDidChange	161
– 5.3.1 control ABPersonViewControlMBS	162
* 5.3.3 retainObject	162
* 5.3.5 View as ABPersonViewMBS	162
* 5.3.7 BoundsChanged	162
* 5.3.8 Close	163

* 5.3.9 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	163
* 5.3.10 ContextualMenuAction(hitItem as MenuItem) as Boolean	163
* 5.3.11 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	163
* 5.3.12 EnableMenuItems	163
* 5.3.13 FrameChanged	164
* 5.3.14 GotFocus	164
* 5.3.15 LostFocus	164
* 5.3.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	164
* 5.3.17 MouseDrag(x as Integer, y as Integer)	164
* 5.3.18 MouseUp(x as Integer, y as Integer)	165
* 5.3.19 Open	165
* 5.3.20 ScaleFactorChanged(NewFactor as Double)	165
* 5.3.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	165
– 5.4.1 class ABPersonViewMBS	166
* 5.4.3 available as Boolean	166
* 5.4.4 Constructor	166
* 5.4.5 Constructor(Handle as Integer)	167
* 5.4.6 Constructor(left as Double, top as Double, width as Double, height as Double)	167
* 5.4.8 editing as Boolean	167
* 5.4.9 person as ABPersonMBS	167
* 5.4.10 shouldShowLinkedPeople as Boolean	168

• 9 Cocoa Controls	297
– 9.1.1 class BevelButton	297
* 9.1.3 NSButtonMBS as NSButtonMBS	297
– 9.2.1 class CanvasGesturesMBS	298
* 9.2.3 AddCanvas(c as Canvas)	298
* 9.2.4 AddCanvas(c as DesktopCanvas)	299
* 9.2.5 Constructor	299
* 9.2.6 Destructor	299
* 9.2.7 RemoveCanvas(c as Canvas)	299
* 9.2.8 RemoveCanvas(c as DesktopCanvas)	299
* 9.2.10 CanvasCount as Integer	300
* 9.2.12 beginGestureWithEvent(can as Variant, e as NSEventMBS) as boolean	300
* 9.2.13 endGestureWithEvent(can as Variant, e as NSEventMBS) as boolean	300
* 9.2.14 magnifyWithEvent(can as Variant, e as NSEventMBS) as boolean	301
* 9.2.15 rotateWithEvent(can as Variant, e as NSEventMBS) as boolean	301
* 9.2.16 scrollWheel(can as Variant, e as NSEventMBS) as boolean	301
* 9.2.17 smartMagnifyWithEvent(can as Variant, e as NSEventMBS) as boolean	301
* 9.2.18 swipeWithEvent(can as Variant, e as NSEventMBS) as boolean	302
* 9.2.19 touchesBeganWithEvent(can as Variant, e as NSEventMBS) as boolean	302
* 9.2.20 touchesCancelledWithEvent(can as Variant, e as NSEventMBS) as boolean	302
* 9.2.21 touchesEndedWithEvent(can as Variant, e as NSEventMBS) as boolean	303
* 9.2.22 touchesMovedWithEvent(can as Variant, e as NSEventMBS) as boolean	303
– 9.3.1 class Checkbox	304
* 9.3.3 NSButtonMBS as NSButtonMBS	304
– 9.4.1 class ComboBox	305
* 9.4.3 NSComboBoxMBS as NSComboBoxMBS	305
– 9.5.1 class CustomNSScrollerMBS	306
* 9.5.3 Constructor	306
* 9.5.4 Constructor(Handle as Integer)	306
* 9.5.5 Constructor(left as Double, top as Double, width as Double, height as Double)	306
* 9.5.6 Destructor	307
* 9.5.8 acceptsFirstMouse(e as NSEventMBS) as boolean	307
* 9.5.9 acceptsFirstResponder as boolean	307
* 9.5.10 becomeFirstResponder as boolean	307
* 9.5.11 beginGestureWithEvent(e as NSEventMBS) as boolean	308
* 9.5.12 canBecomeKeyView as boolean	308
* 9.5.13 Close	308
* 9.5.14 concludeDragOperation(sender as NSDraggingInfoMBS)	308
* 9.5.15 draggingEnded(sender as NSDraggingInfoMBS)	309
* 9.5.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	309

* 9.5.17 draggingExited(sender as NSDraggingInfoMBS)	309
* 9.5.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	310
* 9.5.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	310
* 9.5.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	310
* 9.5.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	311
* 9.5.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	311
* 9.5.23 drawArrow(g as NSGraphicsMBS, Arrow as Integer, highlight as boolean)	312
* 9.5.24 drawKnob(g as NSGraphicsMBS)	312
* 9.5.25 drawKnobSlotInRect(g as NSGraphicsMBS, slotRect as NSRectMBS, highlight as boolean)	312
* 9.5.26 drawParts(g as NSGraphicsMBS)	312
* 9.5.27 endGestureWithEvent(e as NSEventMBS) as boolean	312
* 9.5.28 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	313
* 9.5.29 isOpaque as boolean	313
* 9.5.30 keyDown(e as NSEventMBS) as boolean	313
* 9.5.31 keyUp(e as NSEventMBS) as boolean	313
* 9.5.32 magnifyWithEvent(e as NSEventMBS) as boolean	313
* 9.5.33 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS	314
* 9.5.34 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	314
* 9.5.35 mouseDownCanMoveWindow as boolean	314
* 9.5.36 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	314
* 9.5.37 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	315
* 9.5.38 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	315
* 9.5.39 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	315
* 9.5.40 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	315
* 9.5.41 Open	315
* 9.5.42 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	316
* 9.5.43 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	316
* 9.5.44 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	316
* 9.5.45 performDragOperation(sender as NSDraggingInfoMBS) as boolean	316
* 9.5.46 prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	317
* 9.5.47 pressureChange(e as NSEventMBS) as boolean	317
* 9.5.48 resignFirstResponder as boolean	317
* 9.5.49 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	317
* 9.5.50 rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	318
* 9.5.51 rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	318
* 9.5.52 rotateWithEvent(e as NSEventMBS) as boolean	318
* 9.5.53 scrollWheel(e as NSEventMBS) as boolean	318

* 9.5.54	swipeWithEvent(e as NSEventMBS) as boolean	318
* 9.5.55	updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	319
* 9.5.56	viewDidMoveToWindow	319
* 9.5.57	wantsPeriodicDraggingUpdates as boolean	319

	9
• 7 Cocoa	171
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.8 acceptsFirstMouse(e as NSEventMBS) as boolean	322
* 9.6.12 canBecomeKeyView as boolean	323
* 9.6.31 mouseDownCanMoveWindow as boolean	328
* 9.6.52 viewDidMoveToWindow	333

• 9 Cocoa Controls	297
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.3 Constructor	321
* 9.6.4 Constructor(Handle as Integer)	321
* 9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double)	321
* 9.6.6 Destructor	322
* 9.6.9 acceptsFirstResponder as boolean	322
* 9.6.10 becomeFirstResponder as boolean	322
* 9.6.11 beginGestureWithEvent(e as NSEventMBS) as boolean	322
* 9.6.13 Close	323
* 9.6.14 concludeDragOperation(sender as NSDraggingInfoMBS)	323
* 9.6.15 draggingEnded(sender as NSDraggingInfoMBS)	324
* 9.6.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	324
* 9.6.17 draggingExited(sender as NSDraggingInfoMBS)	324
* 9.6.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	325
* 9.6.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	325
* 9.6.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	325
* 9.6.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	326
* 9.6.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	326
* 9.6.23 endGestureWithEvent(e as NSEventMBS) as boolean	327
* 9.6.24 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	327
* 9.6.25 isOpaque as boolean	327
* 9.6.26 keyDown(e as NSEventMBS) as boolean	327
* 9.6.27 keyUp(e as NSEventMBS) as boolean	327
* 9.6.28 magnifyWithEvent(e as NSEventMBS) as boolean	328
* 9.6.29 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS	328
* 9.6.30 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	328
* 9.6.32 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.33 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.34 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.35 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.36 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.37 Open	330
* 9.6.38 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.39 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.40 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.41 performDragOperation(sender as NSDraggingInfoMBS) as boolean	330

* 9.6.42 prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	331
* 9.6.43 pressureChange(e as NSEventMBS) as boolean	331
* 9.6.44 resignFirstResponder as boolean	331
* 9.6.45 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.46 rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.47 rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.48 rotateWithEvent(e as NSEventMBS) as boolean	332
* 9.6.49 scrollWheel(e as NSEventMBS) as boolean	332
* 9.6.50 swipeWithEvent(e as NSEventMBS) as boolean	333
* 9.6.51 updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	333
* 9.6.53 wantsPeriodicDraggingUpdates as boolean	334

• 7 Cocoa	171
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.8 acceptsFirstMouse(e as NSEventMBS) as boolean	322
* 9.6.12 canBecomeKeyView as boolean	323
* 9.6.31 mouseDownCanMoveWindow as boolean	328
* 9.6.52 viewDidMoveToWindow	333

	13
• 9 Cocoa Controls	297
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.3 Constructor	321
* 9.6.4 Constructor(Handle as Integer)	321
* 9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double)	321
* 9.6.6 Destructor	322
* 9.6.9 acceptsFirstResponder as boolean	322
* 9.6.10 becomeFirstResponder as boolean	322
* 9.6.11 beginGestureWithEvent(e as NSEventMBS) as boolean	322
* 9.6.13 Close	323
* 9.6.14 concludeDragOperation(sender as NSDraggingInfoMBS)	323
* 9.6.15 draggingEnded(sender as NSDraggingInfoMBS)	324
* 9.6.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	324
* 9.6.17 draggingExited(sender as NSDraggingInfoMBS)	324
* 9.6.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	325
* 9.6.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	325
* 9.6.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	325
* 9.6.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	326
* 9.6.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	326
* 9.6.23 endGestureWithEvent(e as NSEventMBS) as boolean	327
* 9.6.24 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	327
* 9.6.25 isOpaque as boolean	327
* 9.6.26 keyDown(e as NSEventMBS) as boolean	327
* 9.6.27 keyUp(e as NSEventMBS) as boolean	327
* 9.6.28 magnifyWithEvent(e as NSEventMBS) as boolean	328
* 9.6.29 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS	328
* 9.6.30 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	328
* 9.6.32 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.33 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.34 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.35 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.36 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.37 Open	330
* 9.6.38 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.39 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.40 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.41 performDragOperation(sender as NSDraggingInfoMBS) as boolean	330

* 9.6.42	prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	331
* 9.6.43	pressureChange(e as NSEventMBS) as boolean	331
* 9.6.44	resignFirstResponder as boolean	331
* 9.6.45	rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.46	rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.47	rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.48	rotateWithEvent(e as NSEventMBS) as boolean	332
* 9.6.49	scrollWheel(e as NSEventMBS) as boolean	332
* 9.6.50	swipeWithEvent(e as NSEventMBS) as boolean	333
* 9.6.51	updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	333
* 9.6.53	wantsPeriodicDraggingUpdates as boolean	334

	15
• 7 Cocoa	171
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.8 acceptsFirstMouse(e as NSEventMBS) as boolean	322
* 9.6.12 canBecomeKeyView as boolean	323
* 9.6.31 mouseDownCanMoveWindow as boolean	328
* 9.6.52 viewDidMoveToWindow	333

• 9 Cocoa Controls	297
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.3 Constructor	321
* 9.6.4 Constructor(Handle as Integer)	321
* 9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double)	321
* 9.6.6 Destructor	322
* 9.6.9 acceptsFirstResponder as boolean	322
* 9.6.10 becomeFirstResponder as boolean	322
* 9.6.11 beginGestureWithEvent(e as NSEventMBS) as boolean	322
* 9.6.13 Close	323
* 9.6.14 concludeDragOperation(sender as NSDraggingInfoMBS)	323
* 9.6.15 draggingEnded(sender as NSDraggingInfoMBS)	324
* 9.6.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	324
* 9.6.17 draggingExited(sender as NSDraggingInfoMBS)	324
* 9.6.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	325
* 9.6.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	325
* 9.6.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	325
* 9.6.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	326
* 9.6.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	326
* 9.6.23 endGestureWithEvent(e as NSEventMBS) as boolean	327
* 9.6.24 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	327
* 9.6.25 isOpaque as boolean	327
* 9.6.26 keyDown(e as NSEventMBS) as boolean	327
* 9.6.27 keyUp(e as NSEventMBS) as boolean	327
* 9.6.28 magnifyWithEvent(e as NSEventMBS) as boolean	328
* 9.6.29 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS	328
* 9.6.30 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	328
* 9.6.32 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.33 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.34 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.35 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.36 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.37 Open	330
* 9.6.38 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.39 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.40 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.41 performDragOperation(sender as NSDraggingInfoMBS) as boolean	330

* 9.6.42 prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	331
* 9.6.43 pressureChange(e as NSEventMBS) as boolean	331
* 9.6.44 resignFirstResponder as boolean	331
* 9.6.45 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.46 rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.47 rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.48 rotateWithEvent(e as NSEventMBS) as boolean	332
* 9.6.49 scrollWheel(e as NSEventMBS) as boolean	332
* 9.6.50 swipeWithEvent(e as NSEventMBS) as boolean	333
* 9.6.51 updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	333
* 9.6.53 wantsPeriodicDraggingUpdates as boolean	334

• 7 Cocoa	171
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.8 acceptsFirstMouse(e as NSEventMBS) as boolean	322
* 9.6.12 canBecomeKeyView as boolean	323
* 9.6.31 mouseDownCanMoveWindow as boolean	328
* 9.6.52 viewDidMoveToWindow	333

	19
• 9 Cocoa Controls	297
– 9.6.1 class CustomNSTextFieldMBS	321
* 9.6.3 Constructor	321
* 9.6.4 Constructor(Handle as Integer)	321
* 9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double)	321
* 9.6.6 Destructor	322
* 9.6.9 acceptsFirstResponder as boolean	322
* 9.6.10 becomeFirstResponder as boolean	322
* 9.6.11 beginGestureWithEvent(e as NSEventMBS) as boolean	322
* 9.6.13 Close	323
* 9.6.14 concludeDragOperation(sender as NSDraggingInfoMBS)	323
* 9.6.15 draggingEnded(sender as NSDraggingInfoMBS)	324
* 9.6.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	324
* 9.6.17 draggingExited(sender as NSDraggingInfoMBS)	324
* 9.6.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	325
* 9.6.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	325
* 9.6.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	325
* 9.6.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	326
* 9.6.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	326
* 9.6.23 endGestureWithEvent(e as NSEventMBS) as boolean	327
* 9.6.24 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	327
* 9.6.25 isOpaque as boolean	327
* 9.6.26 keyDown(e as NSEventMBS) as boolean	327
* 9.6.27 keyUp(e as NSEventMBS) as boolean	327
* 9.6.28 magnifyWithEvent(e as NSEventMBS) as boolean	328
* 9.6.29 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS	328
* 9.6.30 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	328
* 9.6.32 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.33 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.34 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.35 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.36 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	329
* 9.6.37 Open	330
* 9.6.38 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.39 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.40 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	330
* 9.6.41 performDragOperation(sender as NSDraggingInfoMBS) as boolean	330

* 9.6.42	prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	331
* 9.6.43	pressureChange(e as NSEventMBS) as boolean	331
* 9.6.44	resignFirstResponder as boolean	331
* 9.6.45	rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.46	rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.47	rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	332
* 9.6.48	rotateWithEvent(e as NSEventMBS) as boolean	332
* 9.6.49	scrollWheel(e as NSEventMBS) as boolean	332
* 9.6.50	swipeWithEvent(e as NSEventMBS) as boolean	333
* 9.6.51	updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	333
* 9.6.53	wantsPeriodicDraggingUpdates as boolean	334

• 11 Cocoa Toolbar	775
– 11.1.1 class CustomNSToolbarItemMBS	775
* 11.1.3 Constructor(itemIdentifier as string)	775
* 11.1.4 Destructor	776
* 11.1.6 Action	776
* 11.1.7 allowsDuplicatesInToolbar as boolean	776
* 11.1.8 validate as boolean	776
– 11.2.1 class CustomNSToolbarMBS	777
* 11.2.3 Constructor(Identifier as string)	777
* 11.2.4 Destructor	777
* 11.2.6 allowsSizeMode(mode as Integer, SuperAllows as boolean) as boolean	777
* 11.2.7 itemForItemIdentifier(identifier as string, willBeInsertedIntoToolbar as boolean) as NSToolbarItemMBS	778
* 11.2.8 toolbarAllowedItemIdentifiers as string()	779
* 11.2.9 toolbarDefaultItemIdentifiers as string()	779
* 11.2.10 toolbarDidRemoveItem(item as NSToolbarItemMBS, notification as NSNotificationMBS)	779
* 11.2.11 toolbarItemAction(item as NSToolbarItemMBS)	779
* 11.2.12 toolbarItemAllowsDuplicatesInToolbar(item as NSToolbarItemMBS) as boolean	780
* 11.2.13 toolbarItemValidate(item as NSToolbarItemMBS) as boolean	780
* 11.2.14 toolbarSelectableItemIdentifiers as string()	780
* 11.2.15 toolbarWillAddItem(item as NSToolbarItemMBS, notification as NSNotificationMBS)	781

• 9 Cocoa Controls	297
– 9.7.1 class CustomNSViewMBS	335
* 9.7.3 Constructor	335
* 9.7.4 Constructor(Handle as Integer)	335
* 9.7.5 Constructor(left as Double, top as Double, width as Double, height as Double)	336
* 9.7.6 Destructor	336
* 9.7.8 acceptsFirstMouse(e as NSEventMBS) as boolean	336
* 9.7.9 acceptsFirstResponder as boolean	336
* 9.7.10 becomeFirstResponder as boolean	337
* 9.7.11 beginGestureWithEvent(e as NSEventMBS) as boolean	337
* 9.7.12 canBecomeKeyView as boolean	337
* 9.7.13 Close	337
* 9.7.14 concludeDragOperation(sender as NSDraggingInfoMBS)	337
* 9.7.15 draggingEnded(sender as NSDraggingInfoMBS)	338
* 9.7.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	338
* 9.7.17 draggingExited(sender as NSDraggingInfoMBS)	339
* 9.7.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	339
* 9.7.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	339
* 9.7.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	340
* 9.7.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	340
* 9.7.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	340
* 9.7.23 drawFocusRingMask(g as NSGraphicsMBS) as boolean	341
* 9.7.24 DrawRect(g as NSGraphicsMBS, left as Double, top as Double, width as Double, height as Double)	341
* 9.7.25 endGestureWithEvent(e as NSEventMBS) as boolean	342
* 9.7.26 focusRingMaskBounds as NSRectMBS	342
* 9.7.27 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	342
* 9.7.28 isFlipped as Boolean	342
* 9.7.29 isOpaque as boolean	343
* 9.7.30 keyDown(e as NSEventMBS) as boolean	343
* 9.7.31 keyUp(e as NSEventMBS) as boolean	343
* 9.7.32 magnifyWithEvent(e as NSEventMBS) as boolean	343
* 9.7.33 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS	344
* 9.7.34 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	344
* 9.7.35 mouseDownCanMoveWindow as boolean	344
* 9.7.36 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	345
* 9.7.37 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	345

* 9.7.38	mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	345
* 9.7.39	mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	345
* 9.7.40	mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	345
* 9.7.41	Open	345
* 9.7.42	otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	346
* 9.7.43	otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	346
* 9.7.44	otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	346
* 9.7.45	performDragOperation(sender as NSDraggingInfoMBS) as boolean	346
* 9.7.46	prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	347
* 9.7.47	pressureChange(e as NSEventMBS) as boolean	347
* 9.7.48	resignFirstResponder as boolean	347
* 9.7.49	rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	347
* 9.7.50	rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	348
* 9.7.51	rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	348
* 9.7.52	rotateWithEvent(e as NSEventMBS) as boolean	348
* 9.7.53	scrollWheel(e as NSEventMBS) as boolean	348
* 9.7.54	swipeWithEvent(e as NSEventMBS) as boolean	348
* 9.7.55	updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	349
* 9.7.56	viewDidMoveToWindow	349
* 9.7.57	wantsPeriodicDraggingUpdates as boolean	349
– 9.8.1	class DatePicker	351
* 9.8.3	NSDatePickerMBS as NSDatePickerMBS	351
– 9.9.1	class DesktopBevelButton	352
* 9.9.3	NSButtonMBS as NSButtonMBS	352
– 9.10.1	class DesktopButton	353
* 9.10.3	NSButtonMBS as NSButtonMBS	353
– 9.11.1	class DesktopCheckBox	354
* 9.11.3	NSButtonMBS as NSButtonMBS	354
– 9.12.1	class DesktopComboBox	355
* 9.12.3	NSComboBoxMBS as NSComboBoxMBS	355
– 9.13.1	class DesktopDatePicker	356
* 9.13.3	NSDatePickerMBS as NSDatePickerMBS	356
– 9.14.1	class DesktopDisclosureTriangle	357
* 9.14.3	NSButtonMBS as NSButtonMBS	357
– 9.15.1	class DesktopGroupBox	358
* 9.15.3	NSBoxMBS as NSBoxMBS	358
– 9.16.1	class DesktopImageViewer	359
* 9.16.3	NSImageViewMBS as NSImageViewMBS	359

- **12 Controls** 801
 - 12.1.1 class DesktopListbox 801
 - * 12.1.3 HorizontalNSScrollerMBS as NSScrollerMBS 801
 - * 12.1.4 VerticalNSScrollerMBS as NSScrollerMBS 801

	25
• 9 Cocoa Controls	297
– 9.17.1 control DesktopNSButtonControlMBS	360
* 9.17.3 AlternateTitle as String	360
* 9.17.4 BezelStyle as Integer	360
* 9.17.5 ButtonType as Integer	360
* 9.17.6 Title as String	361
* 9.17.7 View as NSButtonMBS	361
* 9.17.9 Action	361
* 9.17.10 BoundsChanged	361
* 9.17.11 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	362
* 9.17.12 FocusLost	362
* 9.17.13 FocusReceived	362
* 9.17.14 FrameChanged	362
* 9.17.15 MenuBarSelected	363
* 9.17.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	363
* 9.17.17 MouseDrag(x as Integer, y as Integer)	363
* 9.17.18 MouseUp(x as Integer, y as Integer)	363
* 9.17.19 ScaleFactorChanged(NewFactor as Double)	364
* 9.17.20 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	364

• 8 Cocoa Collection View	195
– 8.1.1 control DesktopNSCollectionViewControlMBS	195
* 8.1.3 performBatchUpdates(tag as variant)	196
* 8.1.5 ScrollView as NSScrollViewMBS	197
* 8.1.6 View as NSCollectionViewMBS	197
* 8.1.8 acceptDrop(draggingInfo as NSDraggingInfoMBS, indexPath as NSIndexPathMBS, dropOperation as Integer) as Integer	197
* 8.1.9 BoundsChanged	198
* 8.1.10 cancelPrefetchingForItems(indexPaths() as NSIndexPathMBS)	198
* 8.1.11 canDragItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS) as Boolean	198
* 8.1.12 didChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer)	198
* 8.1.13 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	199
* 8.1.14 didDeselectItems(indexPaths() as NSIndexPathMBS)	199
* 8.1.15 didEndDisplayingItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)	199
* 8.1.16 didEndDisplayingSupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)	200
* 8.1.17 didSelectItems(indexPaths() as NSIndexPathMBS)	200
* 8.1.18 draggingImageForItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS, byref dragImageOffset as NSPointMBS) as NSImageMBS	200
* 8.1.19 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, dragOperation as Integer)	201
* 8.1.20 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, indexPaths() as NSIndexPathMBS)	201
* 8.1.21 FocusLost	202
* 8.1.22 FocusReceived	202
* 8.1.23 FrameChanged	202
* 8.1.24 insetForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as NSEdgeInsetsMBS	202
* 8.1.25 ItemApplyLayoutAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS)	203
* 8.1.26 ItemDidTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)	203
* 8.1.27 itemForRepresentedObject(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS	203
* 8.1.28 ItemPreferredLayoutAttributesFittingAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS) as NSCollectionViewLayoutAttributesMBS	204
* 8.1.29 ItemPrepareForReuse(item as NSCollectionViewItemMBS)	204
* 8.1.30 ItemWillTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)	205
* 8.1.31 MenuBarSelected	205

- * 8.1.32 `minimumInteritemSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double` 205
- * 8.1.33 `minimumLineSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double` 206
- * 8.1.34 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean` 206
- * 8.1.35 `MouseDown(x as Integer, y as Integer)` 207
- * 8.1.36 `MouseDown(x As Integer, y As Integer)` 207
- * 8.1.37 `namesOfPromisedFilesDroppedAtDestination(dropURL as String, indexPaths() as NSIndexPathMBS) as String()` 207
- * 8.1.38 `numberOfItemsInSection(section as Integer) as Integer` 208
- * 8.1.39 `numberOfSections as Integer` 208
- * 8.1.40 `pasteboardWriterForItem(indexPath as NSIndexPathMBS) as NSPasteboardItemMBS` 208
- * 8.1.41 `performBatchUpdatesCompleted(tag as variant, finished as boolean)` 209
- * 8.1.42 `performBatchUpdatesWork(tag as variant)` 209
- * 8.1.43 `prefetchItems(indexPaths() as NSIndexPathMBS)` 209
- * 8.1.44 `referenceSizeForFooterInSection(layout as NSCollectionViewLayoutMBS, section as Integer) as NSSizeMBS` 209
- * 8.1.45 `referenceSizeForHeaderInSection(layout as NSCollectionViewLayoutMBS, section as Integer) as NSSizeMBS` 210
- * 8.1.46 `ScaleFactorChanged(NewFactor as double)` 210
- * 8.1.47 `shouldChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer) as NSIndexPathMBS()` 211
- * 8.1.48 `shouldDeselectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()` 211
- * 8.1.49 `shouldSelectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()` 212
- * 8.1.50 `sizeForItemAtIndexPath(layout as NSCollectionViewLayoutMBS, indexPath as NSIndexPathMBS) as NSSizeMBS` 212
- * 8.1.51 `transitionLayout(fromLayout as NSCollectionViewLayoutMBS, toLayout as NSCollectionViewLayoutMBS) as NSCollectionViewTransitionLayoutMBS` 213
- * 8.1.52 `updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)` 213
- * 8.1.53 `validateDrop(draggingInfo as NSDraggingInfoMBS, byref proposedIndexPath as NSIndexPathMBS, byref dropOperation as Integer) as Integer` 214
- * 8.1.54 `viewForSupplementaryElement(kind as String, indexPath as NSIndexPathMBS) as NSViewMBS` 214
- * 8.1.55 `willDisplayItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)` 215
- * 8.1.56 `willDisplaySupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)` 215
- * 8.1.57 `willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)` 215
- * 8.1.58 `writeItems(indexPaths() as NSIndexPathMBS, Pasteboard as NSPasteboardMBS) as Boolean` 216

• 9 Cocoa Controls	297
– 9.18.1 control DesktopNSComboBoxControlMBS	365
* 9.18.3 View as NSComboBoxMBS	365
* 9.18.5 Action	365
* 9.18.6 BoundsChanged	366
* 9.18.7 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	366
* 9.18.8 FocusLost	366
* 9.18.9 FocusReceived	366
* 9.18.10 FrameChanged	366
* 9.18.11 MenuBarSelected	367
* 9.18.12 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	367
* 9.18.13 MouseDrag(x as Integer, y as Integer)	367
* 9.18.14 MouseUp(x As Integer, y As Integer)	367
* 9.18.15 ScaleFactorChanged(NewFactor as double)	368
* 9.18.16 TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	368
* 9.18.17 TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	368
* 9.18.18 TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	369
* 9.18.19 textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	369
* 9.18.20 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean	369
* 9.18.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	369
– 9.19.1 control DesktopNSDatePickerControlMBS	370
* 9.19.3 AcceptTabs as Boolean	370
* 9.19.4 View as NSDatePickerMBS	370
* 9.19.6 Action	370
* 9.19.7 BoundsChanged	371
* 9.19.8 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	371
* 9.19.9 FocusLost	371
* 9.19.10 FocusReceived	371
* 9.19.11 FrameChanged	372
* 9.19.12 MenuBarSelected	372
* 9.19.13 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	372
* 9.19.14 MouseDrag(x as Integer, y as Integer)	372
* 9.19.15 MouseUp(x as Integer, y as Integer)	373
* 9.19.16 ScaleFactorChanged(NewFactor as Double)	373
* 9.19.17 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	373
– 9.20.1 control DesktopNSOutlineControlMBS	374
* 9.20.3 AcceptTabs as Boolean	374
* 9.20.4 allowsColumnReordering as Boolean	374
* 9.20.5 allowsColumnResizing as Boolean	375

* 9.20.6 allowsColumnSelection as Boolean	375
* 9.20.7 allowsEmptySelection as Boolean	375
* 9.20.8 allowsMultipleSelection as Boolean	375
* 9.20.9 autohidesScrollers as Boolean	376
* 9.20.10 hasHorizontalScroller as Boolean	376
* 9.20.11 hasVerticalScroller as Boolean	376
* 9.20.12 ScrollView as NSScrollViewMBS	376
* 9.20.13 View as NSOutlineViewMBS	376
* 9.20.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean	377
* 9.20.16 BoundsChanged	377
* 9.20.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	377
* 9.20.18 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)	378
* 9.20.19 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	378
* 9.20.20 concludeDragOperation(info as NSDraggingInfoMBS)	378
* 9.20.21 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS	379
* 9.20.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	379
* 9.20.23 didClickTableColumn(tableColumn as NSTableColumnMBS)	379
* 9.20.25 didDragTableColumn(tableColumn as NSTableColumnMBS)	380
* 9.20.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	380
* 9.20.27 didTile	380
* 9.20.28 DoubleClick	380
* 9.20.29 draggingEnded(info as NSDraggingInfoMBS)	381
* 9.20.30 draggingExited(info as NSDraggingInfoMBS)	381
* 9.20.31 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	381
* 9.20.32 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)	382
* 9.20.33 FocusLost	382
* 9.20.34 FocusReceived	382
* 9.20.35 FrameChanged	383
* 9.20.36 heightOfRowByItem(item as NSOutlineViewItemMBS) as Double	383
* 9.20.37 isGroupItem(item as NSOutlineViewItemMBS) as Boolean	383
* 9.20.38 isItemExpandable(item as NSOutlineViewItemMBS) as Boolean	384
* 9.20.39 ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	384
* 9.20.40 ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	384
* 9.20.41 itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS	385

* 9.20.42 ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	385
* 9.20.43 ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	385
* 9.20.44 LeftMouseDown(e as NSEventMBS) as Boolean	385
* 9.20.45 LeftMouseDragged(e as NSEventMBS) as Boolean	386
* 9.20.46 LeftMouseUp(e as NSEventMBS) as Boolean	386
* 9.20.47 MenuBarSelected	386
* 9.20.48 MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean	386
* 9.20.49 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	387
* 9.20.50 MouseDrag(x as Integer, y as Integer)	387
* 9.20.51 MouseUp(x as Integer, y as Integer)	387
* 9.20.52 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()	387
* 9.20.53 nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS	388
* 9.20.54 numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	388
* 9.20.55 objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant	388
* 9.20.56 OtherMouseDown(e as NSEventMBS) as Boolean	389
* 9.20.57 OtherMouseDragged(e as NSEventMBS) as Boolean	389
* 9.20.58 OtherMouseUp(e as NSEventMBS) as Boolean	389
* 9.20.59 pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS	389
* 9.20.60 persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant	390
* 9.20.61 RightMouseDown(e as NSEventMBS) as Boolean	390
* 9.20.62 RightMouseDragged(e as NSEventMBS) as Boolean	390
* 9.20.63 RightMouseUp(e as NSEventMBS) as Boolean	390
* 9.20.64 rowViewForItem(item as NSOutlineViewItemMBS) as NSTableRowViewMBS	391
* 9.20.65 ScaleFactorChanged(NewFactor as Double)	391
* 9.20.66 SelectionDidChange(notification as NSNotificationMBS)	391
* 9.20.67 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	391
* 9.20.68 SelectionIsChanging(notification as NSNotificationMBS)	392
* 9.20.69 selectionShouldChangeInOutlineView as Boolean	392
* 9.20.70 setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)	392
* 9.20.71 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean	393
* 9.20.72 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean	393
* 9.20.73 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean	393
* 9.20.74 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean	394

- * 9.20.75 shouldReorderColumn(columnIndex as Integer, newColumnIndex as Integer) as Boolean
394
- * 9.20.76 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean 395
- * 9.20.77 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean 395
- * 9.20.78 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 395
- * 9.20.79 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean 396
- * 9.20.80 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 396
- * 9.20.81 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean 396
- * 9.20.82 sizeToFitWidthOfColumn(Column as Integer) as Double 397
- * 9.20.83 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS) 397
- * 9.20.84 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 397
- * 9.20.85 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 398
- * 9.20.86 toolTipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String 398
- * 9.20.87 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String 398
- * 9.20.88 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS) 399
- * 9.20.89 validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer 399
- * 9.20.90 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS 399
- * 9.20.91 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 400
- * 9.20.92 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 400
- * 9.20.94 willTile 401
- * 9.20.95 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean 401

• 6 AVFoundation	169
– 9.20.1 control DesktopNSOutlineControlMBS	374
* 9.20.24 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	380
* 9.20.93 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	401

	33
• 9 Cocoa Controls	297
– 9.20.1 control DesktopNSOutlineControlMBS	374
* 9.20.3 AcceptTabs as Boolean	374
* 9.20.4 allowsColumnReordering as Boolean	374
* 9.20.5 allowsColumnResizing as Boolean	375
* 9.20.6 allowsColumnSelection as Boolean	375
* 9.20.7 allowsEmptySelection as Boolean	375
* 9.20.8 allowsMultipleSelection as Boolean	375
* 9.20.9 autohidesScrollers as Boolean	376
* 9.20.10 hasHorizontalScroller as Boolean	376
* 9.20.11 hasVerticalScroller as Boolean	376
* 9.20.12 ScrollView as NSScrollViewMBS	376
* 9.20.13 View as NSOutlineViewMBS	376
* 9.20.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean	377
* 9.20.16 BoundsChanged	377
* 9.20.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	377
* 9.20.18 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)	378
* 9.20.19 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	378
* 9.20.20 concludeDragOperation(info as NSDraggingInfoMBS)	378
* 9.20.21 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS	379
* 9.20.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	379
* 9.20.23 didClickTableColumn(tableColumn as NSTableColumnMBS)	379
* 9.20.25 didDragTableColumn(tableColumn as NSTableColumnMBS)	380
* 9.20.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	380
* 9.20.27 didTile	380
* 9.20.28 DoubleClick	380
* 9.20.29 draggingEnded(info as NSDraggingInfoMBS)	381
* 9.20.30 draggingExited(info as NSDraggingInfoMBS)	381
* 9.20.31 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	381
* 9.20.32 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)	382
* 9.20.33 FocusLost	382
* 9.20.34 FocusReceived	382
* 9.20.35 FrameChanged	383
* 9.20.36 heightOfRowByItem(item as NSOutlineViewItemMBS) as Double	383
* 9.20.37 isGroupItem(item as NSOutlineViewItemMBS) as Boolean	383

* 9.20.38	isItemExpandable(item as NSOutlineViewItemMBS) as Boolean	384
* 9.20.39	ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	384
* 9.20.40	ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	384
* 9.20.41	itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS	385
* 9.20.42	ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	385
* 9.20.43	ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	385
* 9.20.44	LeftMouseDown(e as NSEventMBS) as Boolean	385
* 9.20.45	LeftMouseDragged(e as NSEventMBS) as Boolean	386
* 9.20.46	LeftMouseUp(e as NSEventMBS) as Boolean	386
* 9.20.47	MenuBarSelected	386
* 9.20.48	MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean	386
* 9.20.49	mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	387
* 9.20.50	MouseDown(x as Integer, y as Integer)	387
* 9.20.51	MouseUp(x as Integer, y as Integer)	387
* 9.20.52	namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()	387
* 9.20.53	nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS	388
* 9.20.54	numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	388
* 9.20.55	objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant	388
* 9.20.56	OtherMouseDown(e as NSEventMBS) as Boolean	389
* 9.20.57	OtherMouseDragged(e as NSEventMBS) as Boolean	389
* 9.20.58	OtherMouseUp(e as NSEventMBS) as Boolean	389
* 9.20.59	pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS	389
* 9.20.60	persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant	390
* 9.20.61	RightMouseDown(e as NSEventMBS) as Boolean	390
* 9.20.62	RightMouseDragged(e as NSEventMBS) as Boolean	390
* 9.20.63	RightMouseUp(e as NSEventMBS) as Boolean	390
* 9.20.64	rowViewForItem(item as NSOutlineViewItemMBS) as NSTableRowViewMBS	391
* 9.20.65	ScaleFactorChanged(NewFactor as Double)	391
* 9.20.66	SelectionDidChange(notification as NSNotificationMBS)	391
* 9.20.67	selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	391
* 9.20.68	SelectionIsChanging(notification as NSNotificationMBS)	392
* 9.20.69	selectionShouldChangeInOutlineView as Boolean	392
* 9.20.70	setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)	392

- * 9.20.71 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean 393
- * 9.20.72 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean 393
- * 9.20.73 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 393
- * 9.20.74 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean 394
- * 9.20.75 shouldReorderColumn(columnIndex as Integer, newIndex as Integer) as Boolean 394
- * 9.20.76 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean 395
- * 9.20.77 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean 395
- * 9.20.78 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 395
- * 9.20.79 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean 396
- * 9.20.80 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 396
- * 9.20.81 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean 396
- * 9.20.82 sizeToFitWidthOfColumn(column as Integer) as Double 397
- * 9.20.83 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS) 397
- * 9.20.84 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 397
- * 9.20.85 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 398
- * 9.20.86 tooltipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String 398
- * 9.20.87 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String 398
- * 9.20.88 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS) 399
- * 9.20.89 validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer 399
- * 9.20.90 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS 399
- * 9.20.91 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 400
- * 9.20.92 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 400
- * 9.20.94 willTile 401
- * 9.20.95 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean 401

• 6 AVFoundation	169
– 9.20.1 control DesktopNSOutlineControlMBS	374
* 9.20.24 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	380
* 9.20.93 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	401

	37
• 9 Cocoa Controls	297
– 9.20.1 control DesktopNSOutlineControlMBS	374
* 9.20.3 AcceptTabs as Boolean	374
* 9.20.4 allowsColumnReordering as Boolean	374
* 9.20.5 allowsColumnResizing as Boolean	375
* 9.20.6 allowsColumnSelection as Boolean	375
* 9.20.7 allowsEmptySelection as Boolean	375
* 9.20.8 allowsMultipleSelection as Boolean	375
* 9.20.9 autohidesScrollers as Boolean	376
* 9.20.10 hasHorizontalScroller as Boolean	376
* 9.20.11 hasVerticalScroller as Boolean	376
* 9.20.12 ScrollView as NSScrollViewMBS	376
* 9.20.13 View as NSOutlineViewMBS	376
* 9.20.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean	377
* 9.20.16 BoundsChanged	377
* 9.20.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	377
* 9.20.18 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)	378
* 9.20.19 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	378
* 9.20.20 concludeDragOperation(info as NSDraggingInfoMBS)	378
* 9.20.21 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS	379
* 9.20.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	379
* 9.20.23 didClickTableColumn(tableColumn as NSTableColumnMBS)	379
* 9.20.25 didDragTableColumn(tableColumn as NSTableColumnMBS)	380
* 9.20.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	380
* 9.20.27 didTile	380
* 9.20.28 DoubleClick	380
* 9.20.29 draggingEnded(info as NSDraggingInfoMBS)	381
* 9.20.30 draggingExited(info as NSDraggingInfoMBS)	381
* 9.20.31 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	381
* 9.20.32 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)	382
* 9.20.33 FocusLost	382
* 9.20.34 FocusReceived	382
* 9.20.35 FrameChanged	383
* 9.20.36 heightOfRowByItem(item as NSOutlineViewItemMBS) as Double	383
* 9.20.37 isGroupItem(item as NSOutlineViewItemMBS) as Boolean	383

* 9.20.38	isItemExpandable(item as NSOutlineViewItemMBS) as Boolean	384
* 9.20.39	ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	384
* 9.20.40	ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	384
* 9.20.41	itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS	385
* 9.20.42	ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	385
* 9.20.43	ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	385
* 9.20.44	LeftMouseDown(e as NSEventMBS) as Boolean	385
* 9.20.45	LeftMouseDragged(e as NSEventMBS) as Boolean	386
* 9.20.46	LeftMouseUp(e as NSEventMBS) as Boolean	386
* 9.20.47	MenuBarSelected	386
* 9.20.48	MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean	386
* 9.20.49	mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	387
* 9.20.50	MouseDown(x as Integer, y as Integer)	387
* 9.20.51	MouseUp(x as Integer, y as Integer)	387
* 9.20.52	namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()	387
* 9.20.53	nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS	388
* 9.20.54	numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	388
* 9.20.55	objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant	388
* 9.20.56	OtherMouseDown(e as NSEventMBS) as Boolean	389
* 9.20.57	OtherMouseDragged(e as NSEventMBS) as Boolean	389
* 9.20.58	OtherMouseUp(e as NSEventMBS) as Boolean	389
* 9.20.59	pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS	389
* 9.20.60	persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant	390
* 9.20.61	RightMouseDown(e as NSEventMBS) as Boolean	390
* 9.20.62	RightMouseDragged(e as NSEventMBS) as Boolean	390
* 9.20.63	RightMouseUp(e as NSEventMBS) as Boolean	390
* 9.20.64	rowViewForItem(item as NSOutlineViewItemMBS) as NSTableRowViewMBS	391
* 9.20.65	ScaleFactorChanged(NewFactor as Double)	391
* 9.20.66	SelectionDidChange(notification as NSNotificationMBS)	391
* 9.20.67	selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	391
* 9.20.68	SelectionIsChanging(notification as NSNotificationMBS)	392
* 9.20.69	selectionShouldChangeInOutlineView as Boolean	392
* 9.20.70	setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)	392

- * 9.20.71 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean 393
- * 9.20.72 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean 393
- * 9.20.73 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 393
- * 9.20.74 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean 394
- * 9.20.75 shouldReorderColumn(columnIndex as Integer, newIndex as Integer) as Boolean 394
- * 9.20.76 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean 395
- * 9.20.77 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean 395
- * 9.20.78 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 395
- * 9.20.79 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean 396
- * 9.20.80 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 396
- * 9.20.81 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean 396
- * 9.20.82 sizeToFitWidthOfColumn(column as Integer) as Double 397
- * 9.20.83 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS) 397
- * 9.20.84 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 397
- * 9.20.85 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 398
- * 9.20.86 toolTipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String 398
- * 9.20.87 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String 398
- * 9.20.88 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS) 399
- * 9.20.89 validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer 399
- * 9.20.90 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS 399
- * 9.20.91 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 400
- * 9.20.92 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 400
- * 9.20.94 willTile 401
- * 9.20.95 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean 401
- 9.21.1 control DesktopNSPopUpButtonControlMBS 402
 - * 9.21.3 View as NSPopUpButtonMBS 402
 - * 9.21.5 Action 402
 - * 9.21.6 BoundsChanged 402

* 9.21.7	didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	403
* 9.21.8	FocusLost	403
* 9.21.9	FocusReceived	403
* 9.21.10	FrameChanged	403
* 9.21.11	MenuBarSelected	403
* 9.21.12	MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	404
* 9.21.13	MouseDown(x as Integer, y as Integer)	404
* 9.21.14	MouseUp(x as Integer, y as Integer)	404
* 9.21.15	ScaleFactorChanged(NewFactor as Double)	404
* 9.21.16	willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	405
– 9.22.1	control DesktopNSSwitchControlMBS	406
* 9.22.3	State as Boolean	406
* 9.22.4	View as NSSwitchMBS	406
* 9.22.6	Action	407
* 9.22.7	BoundsChanged	407
* 9.22.8	didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	407
* 9.22.9	FocusLost	407
* 9.22.10	FocusReceived	407
* 9.22.11	FrameChanged	408
* 9.22.12	MenuBarSelected	408
* 9.22.13	MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	408
* 9.22.14	MouseDown(x as Integer, y as Integer)	408
* 9.22.15	MouseUp(x As Integer, y As Integer)	409
* 9.22.16	ScaleFactorChanged(NewFactor as double)	409
* 9.22.17	willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	409
– 9.23.1	control DesktopNSTableControlMBS	410
* 9.23.3	AcceptTabs as Boolean	410
* 9.23.4	allowsColumnReordering as Boolean	411
* 9.23.5	allowsColumnResizing as Boolean	411
* 9.23.6	allowsColumnSelection as Boolean	411
* 9.23.7	allowsEmptySelection as Boolean	411
* 9.23.8	allowsMultipleSelection as Boolean	412
* 9.23.9	autohidesScrollers as Boolean	412
* 9.23.10	disableCellEvents as Boolean	412
* 9.23.11	disableViewEvents as Boolean	412
* 9.23.12	hasHorizontalScroller as Boolean	413
* 9.23.13	hasVerticalScroller as Boolean	413
* 9.23.14	ScrollView as NSScrollViewMBS	413
* 9.23.15	View as NSTableViewMBS	413
* 9.23.17	acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean	413

	41
* 9.23.18 BoundsChanged	414
* 9.23.19 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	414
* 9.23.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	414
* 9.23.21 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	414
* 9.23.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	415
* 9.23.23 didClickTableColumn(tableColumn as NSTableColumnMBS)	415
* 9.23.25 didDragTableColumn(tableColumn as NSTableColumnMBS)	415
* 9.23.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	416
* 9.23.27 didTile	416
* 9.23.28 DoubleClick	416
* 9.23.29 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	417
* 9.23.30 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)	417
* 9.23.31 FocusLost	417
* 9.23.32 FocusReceived	418
* 9.23.33 FrameChanged	418
* 9.23.34 heightOfRow(row as Int64) as Double	418
* 9.23.35 isGroupRow(row as Int64) as boolean	418
* 9.23.36 LeftMouseDown(e as NSEventMBS) as boolean	419
* 9.23.37 LeftMouseDragged(e as NSEventMBS) as boolean	419
* 9.23.38 LeftMouseUp(e as NSEventMBS) as boolean	419
* 9.23.39 MenuBarSelected	419
* 9.23.40 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	420
* 9.23.41 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	420
* 9.23.42 MouseDrag(x as Integer, y as Integer)	420
* 9.23.43 MouseUp(x as Integer, y as Integer)	420
* 9.23.44 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()	421
* 9.23.45 nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64	421
* 9.23.46 numberOfRowsInTableView as Integer	421
* 9.23.47 objectValue(column as NSTableColumnMBS, row as Integer) as Variant	422
* 9.23.48 OtherMouseDown(e as NSEventMBS) as boolean	422
* 9.23.49 OtherMouseDragged(e as NSEventMBS) as boolean	422
* 9.23.50 OtherMouseUp(e as NSEventMBS) as boolean	422
* 9.23.51 pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS	422
* 9.23.52 RightMouseDown(e as NSEventMBS) as boolean	423
* 9.23.53 RightMouseDragged(e as NSEventMBS) as boolean	423
* 9.23.54 RightMouseUp(e as NSEventMBS) as boolean	423

- * 9.23.55 `rowActionsForRow(row as Integer, edge as Integer)` as `NSTableViewRowActionMBS()`
423
- * 9.23.56 `rowViewForRow(row as Integer)` as `NSTableRowViewMBS` 424
- * 9.23.57 `ScaleFactorChanged(NewFactor as Double)` 424
- * 9.23.58 `SelectionDidChange(notification as NSNotificationMBS)` 425
- * 9.23.59 `selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS)`
as `NSIndexSetMBS` 425
- * 9.23.60 `SelectionIsChanging(notification as NSNotificationMBS)` 425
- * 9.23.61 `selectionShouldChangeInTableView` as `boolean` 425
- * 9.23.62 `setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)`
426
- * 9.23.63 `shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64)` as `boolean`
426
- * 9.23.64 `shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64)` as `boolean`
426
- * 9.23.65 `shouldSelectRow(row as Int64)` as `boolean` 427
- * 9.23.66 `shouldSelectTableColumn(tableColumn as NSTableColumnMBS)` as `boolean` 427
- * 9.23.67 `shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64)` as
`Boolean` 427
- * 9.23.68 `shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as`
`Int64)` as `Boolean` 428
- * 9.23.69 `shouldTypeSelectForEvent(e as NSEventMBS, searchString as string)` as `Boolean` 428
- * 9.23.70 `sizeToFitWidthOfColumn(column as Int64)` as `Double` 429
- * 9.23.71 `sortDescriptorsDidChange(oldDescriptors())` as `NSSortDescriptorMBS)` 429
- * 9.23.72 `textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS)` as
`boolean` 429
- * 9.23.73 `textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS)` as `boolean`
430
- * 9.23.74 `toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumn-`
`columnMBS, row as Int64, mouseLocation as NSPointMBS)` as `string` 430
- * 9.23.75 `typeSelectString(tableColumn as NSTableColumnMBS, row as Int64)` as `string` 430
- * 9.23.76 `updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)` 431
- * 9.23.77 `validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation`
`as Integer)` as `Integer` 431
- * 9.23.78 `view(tableColumn as NSTableColumnMBS, row as Integer)` as `NSViewMBS` 432
- * 9.23.79 `willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as`
`Int64)` 432
- * 9.23.81 `willTile` 432
- * 9.23.82 `writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS)`
as `boolean` 433

	43
• 6 AVFoundation	169
– 9.23.1 control DesktopNSTableControlMBS	410
* 9.23.24 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	415
* 9.23.80 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	432

• 9 Cocoa Controls	297
– 9.23.1 control DesktopNSTableControlMBS	410
* 9.23.3 AcceptTabs as Boolean	410
* 9.23.4 allowsColumnReordering as Boolean	411
* 9.23.5 allowsColumnResizing as Boolean	411
* 9.23.6 allowsColumnSelection as Boolean	411
* 9.23.7 allowsEmptySelection as Boolean	411
* 9.23.8 allowsMultipleSelection as Boolean	412
* 9.23.9 autohidesScrollers as Boolean	412
* 9.23.10 disableCellEvents as Boolean	412
* 9.23.11 disableViewEvents as Boolean	412
* 9.23.12 hasHorizontalScroller as Boolean	413
* 9.23.13 hasVerticalScroller as Boolean	413
* 9.23.14 ScrollView as NSScrollViewMBS	413
* 9.23.15 View as NSTableViewMBS	413
* 9.23.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean	413
* 9.23.18 BoundsChanged	414
* 9.23.19 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	414
* 9.23.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	414
* 9.23.21 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	414
* 9.23.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	415
* 9.23.23 didClickTableColumn(tableColumn as NSTableColumnMBS)	415
* 9.23.25 didDragTableColumn(tableColumn as NSTableColumnMBS)	415
* 9.23.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	416
* 9.23.27 didTile	416
* 9.23.28 DoubleClick	416
* 9.23.29 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	417
* 9.23.30 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)	417
* 9.23.31 FocusLost	417
* 9.23.32 FocusReceived	418
* 9.23.33 FrameChanged	418
* 9.23.34 heightOfRow(row as Int64) as Double	418
* 9.23.35 isGroupRow(row as Int64) as boolean	418
* 9.23.36 LeftMouseDown(e as NSEventMBS) as boolean	419
* 9.23.37 LeftMouseDragged(e as NSEventMBS) as boolean	419
* 9.23.38 LeftMouseUp(e as NSEventMBS) as boolean	419
* 9.23.39 MenuBarSelected	419

- * 9.23.40 `MouseDown(x as Integer, y as Integer, Modifiers as Integer)` As Boolean 420
- * 9.23.41 `mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)` 420
- * 9.23.42 `MouseDown(x as Integer, y as Integer)` 420
- * 9.23.43 `MouseUp(x as Integer, y as Integer)` 420
- * 9.23.44 `namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS)` as string() 421
- * 9.23.45 `nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string)` as Int64 421
- * 9.23.46 `numberOfRowsInTableView` as Integer 421
- * 9.23.47 `objectValue(column as NSTableColumnMBS, row as Integer)` as Variant 422
- * 9.23.48 `OtherMouseDown(e as NSEventMBS)` as boolean 422
- * 9.23.49 `OtherMouseDragged(e as NSEventMBS)` as boolean 422
- * 9.23.50 `OtherMouseUp(e as NSEventMBS)` as boolean 422
- * 9.23.51 `pasteboardItemForRow(row as Integer)` as NSPasteboardItemMBS 422
- * 9.23.52 `RightMouseDown(e as NSEventMBS)` as boolean 423
- * 9.23.53 `RightMouseDragged(e as NSEventMBS)` as boolean 423
- * 9.23.54 `RightMouseUp(e as NSEventMBS)` as boolean 423
- * 9.23.55 `rowActionsForRow(row as Integer, edge as Integer)` as NSTableViewRowActionMBS() 423
- * 9.23.56 `rowViewForRow(row as Integer)` as NSTableRowViewMBS 424
- * 9.23.57 `ScaleFactorChanged(NewFactor as Double)` 424
- * 9.23.58 `SelectionDidChange(notification as NSNotificationMBS)` 425
- * 9.23.59 `selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS)` as NSIndexSetMBS 425
- * 9.23.60 `SelectionIsChanging(notification as NSNotificationMBS)` 425
- * 9.23.61 `selectionShouldChangeInTableView` as boolean 425
- * 9.23.62 `setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)` 426
- * 9.23.63 `shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64)` as boolean 426
- * 9.23.64 `shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64)` as boolean 426
- * 9.23.65 `shouldSelectRow(row as Int64)` as boolean 427
- * 9.23.66 `shouldSelectTableColumn(tableColumn as NSTableColumnMBS)` as boolean 427
- * 9.23.67 `shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64)` as Boolean 427
- * 9.23.68 `shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)` as Boolean 428
- * 9.23.69 `shouldTypeSelectForEvent(e as NSEventMBS, searchString as string)` as Boolean 428
- * 9.23.70 `sizeToFitWidthOfColumn(column as Int64)` as Double 429
- * 9.23.71 `sortDescriptorsDidChange(oldDescriptors())` as NSSortDescriptorMBS) 429
- * 9.23.72 `textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS)` as boolean 429

- * 9.23.73 `textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS)` as boolean
430
- * 9.23.74 `toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS)` as string 430
- * 9.23.75 `typeSelectString(tableColumn as NSTableColumnMBS, row as Int64)` as string 430
- * 9.23.76 `updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)` 431
- * 9.23.77 `validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer)` as Integer 431
- * 9.23.78 `view(tableColumn as NSTableColumnMBS, row as Integer)` as NSViewMBS 432
- * 9.23.79 `willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)` 432
- * 9.23.81 `willTile` 432
- * 9.23.82 `writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS)`
as boolean 433

	47
• 6 AVFoundation	169
– 9.23.1 control DesktopNSTableControlMBS	410
* 9.23.24 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	415
* 9.23.80 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	432

• 9 Cocoa Controls	297
– 9.23.1 control DesktopNSTableControlMBS	410
* 9.23.3 AcceptTabs as Boolean	410
* 9.23.4 allowsColumnReordering as Boolean	411
* 9.23.5 allowsColumnResizing as Boolean	411
* 9.23.6 allowsColumnSelection as Boolean	411
* 9.23.7 allowsEmptySelection as Boolean	411
* 9.23.8 allowsMultipleSelection as Boolean	412
* 9.23.9 autohidesScrollers as Boolean	412
* 9.23.10 disableCellEvents as Boolean	412
* 9.23.11 disableViewEvents as Boolean	412
* 9.23.12 hasHorizontalScroller as Boolean	413
* 9.23.13 hasVerticalScroller as Boolean	413
* 9.23.14 ScrollView as NSScrollViewMBS	413
* 9.23.15 View as NSTableViewMBS	413
* 9.23.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean	413
* 9.23.18 BoundsChanged	414
* 9.23.19 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	414
* 9.23.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	414
* 9.23.21 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	414
* 9.23.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	415
* 9.23.23 didClickTableColumn(tableColumn as NSTableColumnMBS)	415
* 9.23.25 didDragTableColumn(tableColumn as NSTableColumnMBS)	415
* 9.23.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	416
* 9.23.27 didTile	416
* 9.23.28 DoubleClick	416
* 9.23.29 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	417
* 9.23.30 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)	417
* 9.23.31 FocusLost	417
* 9.23.32 FocusReceived	418
* 9.23.33 FrameChanged	418
* 9.23.34 heightOfRow(row as Int64) as Double	418
* 9.23.35 isGroupRow(row as Int64) as boolean	418
* 9.23.36 LeftMouseDown(e as NSEventMBS) as boolean	419
* 9.23.37 LeftMouseDragged(e as NSEventMBS) as boolean	419
* 9.23.38 LeftMouseUp(e as NSEventMBS) as boolean	419
* 9.23.39 MenuBarSelected	419

* 9.23.40	MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	420
* 9.23.41	mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	420
* 9.23.42	MouseDown(x as Integer, y as Integer)	420
* 9.23.43	MouseUp(x as Integer, y as Integer)	420
* 9.23.44	namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()	421
* 9.23.45	nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64	421
* 9.23.46	numberOfRowsInTableView as Integer	421
* 9.23.47	objectValue(column as NSTableColumnMBS, row as Integer) as Variant	422
* 9.23.48	OtherMouseDown(e as NSEventMBS) as boolean	422
* 9.23.49	OtherMouseDragged(e as NSEventMBS) as boolean	422
* 9.23.50	OtherMouseUp(e as NSEventMBS) as boolean	422
* 9.23.51	pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS	422
* 9.23.52	RightMouseDown(e as NSEventMBS) as boolean	423
* 9.23.53	RightMouseDragged(e as NSEventMBS) as boolean	423
* 9.23.54	RightMouseUp(e as NSEventMBS) as boolean	423
* 9.23.55	rowActionsForRow(row as Integer, edge as Integer) as NSTableViewRowActionMBS()	423
* 9.23.56	rowViewForRow(row as Integer) as NSTableRowViewMBS	424
* 9.23.57	ScaleFactorChanged(NewFactor as Double)	424
* 9.23.58	SelectionDidChange(notification as NSNotificationMBS)	425
* 9.23.59	selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	425
* 9.23.60	SelectionIsChanging(notification as NSNotificationMBS)	425
* 9.23.61	selectionShouldChangeInTableView as boolean	425
* 9.23.62	setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)	426
* 9.23.63	shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean	426
* 9.23.64	shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64) as boolean	426
* 9.23.65	shouldSelectRow(row as Int64) as boolean	427
* 9.23.66	shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean	427
* 9.23.67	shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean	427
* 9.23.68	shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean	428
* 9.23.69	shouldTypeSelectForEvent(e as NSEventMBS, searchString as string) as Boolean	428
* 9.23.70	sizeToFitWidthOfColumn(column as Int64) as Double	429
* 9.23.71	sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)	429
* 9.23.72	textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean	429

* 9.23.73	textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean	430
* 9.23.74	toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string	430
* 9.23.75	typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string	430
* 9.23.76	updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)	431
* 9.23.77	validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer) as Integer	431
* 9.23.78	view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS	432
* 9.23.79	willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)	432
* 9.23.81	willTile	432
* 9.23.82	writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS) as boolean	433
– 9.24.1	control DesktopNSTokenFieldControlMBS	434
* 9.24.3	View as NSTokenFieldMBS	434
* 9.24.5	BoundsChanged	434
* 9.24.6	completionsForSubstring(substring as string, tokenIndex as Integer, byref selectedIndex as Integer) as Variant()	434
* 9.24.7	didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	435
* 9.24.8	displayStringForRepresentedObject(representedObject as Variant) as string	435
* 9.24.9	editingStringForRepresentedObject(representedObject as Variant) as string	435
* 9.24.10	FocusLost	435
* 9.24.11	FocusReceived	436
* 9.24.12	FrameChanged	436
* 9.24.13	hasMenuForRepresentedObject(representedObject as Variant) as boolean	436
* 9.24.14	MenuBarSelected	436
* 9.24.15	menuForRepresentedObject(representedObject as Variant) as NSMenuMBS	437
* 9.24.16	MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	437
* 9.24.17	MouseDown(x as Integer, y as Integer)	437
* 9.24.18	MouseDown(x as Integer, y as Integer)	437
* 9.24.19	readFromPasteboard(pboard as NSPasteboardMBS) as Variant()	438
* 9.24.20	representedObjectForEditingString(editingString as string) as Variant	438
* 9.24.21	ScaleFactorChanged(NewFactor as Double)	438
* 9.24.22	shouldAddObjects(tokens() as Variant, index as Integer) as Variant()	438
* 9.24.23	styleForRepresentedObject(representedObject as Variant) as Integer	439
* 9.24.24	TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	439
* 9.24.25	TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	440
* 9.24.26	TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	440
* 9.24.27	textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	440
* 9.24.28	textShouldEndEditing(fieldEditor as NSTextMBS) as boolean	441

	51
* 9.24.29 tokenFieldAction	441
* 9.24.30 tokenFieldTextShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	441
* 9.24.31 tokenFieldTextShouldEndEditing(fieldEditor as NSTextMBS) as boolean	441
* 9.24.32 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	441
* 9.24.33 writeRepresentedObjects(objects() as Variant, pboard as NSPasteboardMBS) as boolean	442
– 9.25.1 class DesktopPopupMenu	443
* 9.25.3 NSButtonMBS as NSButtonMBS	443
* 9.25.4 NSPopUpButtonMBS as NSPopUpButtonMBS	443

- **9 Cocoa Controls** 297
 - 12.2.1 class DesktopProgressbar 802
 - * 12.2.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS 802
 - 9.26.1 class DesktopProgressWheel 445
 - * 9.26.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS 445

	53
• 16 Quartz Composer	949
– 16.1.1 control DesktopQCViewControlMBS	949
* 16.1.3 View as QCViewMBS	949
* 16.1.5 BoundsChanged	950
* 16.1.6 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	950
* 16.1.7 DidStartRendering	950
* 16.1.8 DidStopRendering	950
* 16.1.9 FocusLost	950
* 16.1.10 FocusReceived	951
* 16.1.11 FrameChanged	951
* 16.1.12 MenuBarSelected	951
* 16.1.13MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	951
* 16.1.14 MouseDrag(x as Integer, y as Integer)	952
* 16.1.15 MouseUp(x as Integer, y as Integer)	952
* 16.1.16 ScaleFactorChanged(NewFactor as Double)	952
* 16.1.17 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	952

• 9 Cocoa Controls	297
– 9.27.1 class DesktopRadioButton	446
* 9.27.3 NSButtonMBS as NSButtonMBS	446
– 9.28.1 class DesktopRadioGroup	447
* 9.28.3 NSButtonsMBS as NSButtonMBS()	447
– 9.29.1 class DesktopScrollBar	448
* 9.29.3 NSScrollerMBS as NSScrollerMBS	448
– 9.30.1 class DesktopSeparator	449
* 9.30.3 NSBoxMBS as NSBoxMBS	449
– 9.31.1 class DesktopSlider	450
* 9.31.3 NSSliderMBS as NSSliderMBS	450
– 9.32.1 class DesktopTextArea	451
* 9.32.3 NSScrollViewMBS as NSScrollViewMBS	451
– 9.33.1 class DesktopUpDownArrows	453
* 9.33.3 NSStepperMBS as NSStepperMBS	453

	55
• 11 Cocoa Toolbar	775
– 18.1.1 class DesktopWindow	987
* 18.1.3 NSToolbarMBS as NSToolbarMBS	987

- **9 Cocoa Controls** 297
 - 9.34.1 class DisclosureTriangle 454
 - * 9.34.3 NSButtonMBS as NSButtonMBS 454

• 14 DiscRecording	815
– 14.1.1 class DRBurnMBS	815
* 14.1.3 abort	816
* 14.1.4 burnForDevice(device as DRDeviceMBS) as DRBurnMBS	816
* 14.1.5 Constructor(device as DRDeviceMBS)	816
* 14.1.6 device as DRDeviceMBS	816
* 14.1.7 DRBurnAppendableKey as string	816
* 14.1.8 DRBurnCompletionActionEject as string	817
* 14.1.9 DRBurnCompletionActionKey as string	817
* 14.1.10 DRBurnCompletionActionMount as string	817
* 14.1.11 DRBurnDoubleLayerL0DataZoneBlocksKey as string	817
* 14.1.12 DRBurnFailureActionEject as string	818
* 14.1.13 DRBurnFailureActionKey as string	818
* 14.1.14 DRBurnFailureActionNone as string	818
* 14.1.15 DRBurnOverwriteDiscKey as string	818
* 14.1.16 DRBurnRequestedSpeedKey as string	819
* 14.1.17 DRBurnStatusChangedNotification as string	819
* 14.1.18 DRBurnStrategyBDDAO as string	819
* 14.1.19 DRBurnStrategyCDSAO as string	819
* 14.1.20 DRBurnStrategyCDTAO as string	819
* 14.1.21 DRBurnStrategyDVDDAO as string	820
* 14.1.22 DRBurnStrategyIsRequiredKey as string	820
* 14.1.23 DRBurnStrategyKey as string	820
* 14.1.24 DRBurnTestingKey as string	821
* 14.1.25 DRBurnUnderrunProtectionKey as string	821
* 14.1.26 DRBurnVerifyDiscKey as string	821
* 14.1.27 DRCDTextKey as string	821
* 14.1.28 DRErrorStatusAdditionalSenseStringKey as string	822
* 14.1.29 DRErrorStatusErrorInfoStringKey as string	822
* 14.1.30 DRErrorStatusErrorKey as string	822
* 14.1.31 DRErrorStatusErrorStringKey as string	822
* 14.1.32 DRErrorStatusKey as string	823
* 14.1.33 DRErrorStatusSenseCodeStringKey as string	823
* 14.1.34 DRErrorStatusSenseKey as string	823
* 14.1.35 DRMediaCatalogNumberKey as string	823
* 14.1.36 DRStatusCurrentSessionKey as string	824
* 14.1.37 DRStatusCurrentSpeedKey as string	824
* 14.1.38 DRStatusCurrentTrackKey as string	824
* 14.1.39 DRStatusEraseTypeKey as string	824
* 14.1.40 DRStatusPercentCompleteKey as string	824
* 14.1.41 DRStatusProgressCurrentKPS as string	824

* 14.1.42 DRStatusProgressCurrentXFactor as string	825
* 14.1.43 DRStatusProgressInfoKey as string	825
* 14.1.44 DRStatusStateDone as string	825
* 14.1.45 DRStatusStateErasing as string	825
* 14.1.46 DRStatusStateFailed as string	825
* 14.1.47 DRStatusStateFinishing as string	826
* 14.1.48 DRStatusStateKey as string	826
* 14.1.49 DRStatusStateNone as string	826
* 14.1.50 DRStatusStatePreparing as string	826
* 14.1.51 DRStatusStateSessionClose as string	826
* 14.1.52 DRStatusStateSessionOpen as string	826
* 14.1.53 DRStatusStateTrackClose as string	827
* 14.1.54 DRStatusStateTrackOpen as string	827
* 14.1.55 DRStatusStateTrackWrite as string	827
* 14.1.56 DRStatusStateVerifying as string	827
* 14.1.57 DRStatusTotalSessionsKey as string	827
* 14.1.58 DRStatusTotalTracksKey as string	828
* 14.1.59 DRSynchronousBehaviorKey as string	828
* 14.1.60 status as dictionary	828
* 14.1.61 writeImageFile(ImageFile as FolderItem) as boolean	828
* 14.1.62 writeImageFile(ImagePath as String) as boolean	829
* 14.1.63 writeLayout(track as DRTrackMBS)	829
* 14.1.64 writeLayout(tracks() as DRTrackMBS)	829
* 14.1.66 appendable as boolean	829
* 14.1.67 BurnFailureAction as string	830
* 14.1.68 completionAction as string	830
* 14.1.69 DoubleLayerL0DataZoneBlocks as Double	830
* 14.1.70 MediaCatalogNumber as memoryblock	831
* 14.1.71 Overwrite as boolean	831
* 14.1.72 properties as dictionary	831
* 14.1.73 requestedBurnSpeed as single	832
* 14.1.74 Testing as boolean	832
* 14.1.75 UnderrunProtection as boolean	832
* 14.1.76 verifyDisc as boolean	832
– 14.2.1 class DRBurnProgressPanelMBS	834
* 14.2.3 beginProgressPanelForBurn(burn as DRBurnMBS, track as DRTrackMBS)	835
* 14.2.4 beginProgressPanelForBurn(burn as DRBurnMBS, tracks() as DRTrackMBS)	835
* 14.2.5 beginProgressPanelForImageFile(burn as DRBurnMBS, file as folderitem) as boolean	835
* 14.2.6 beginProgressPanelForImageFile(burn as DRBurnMBS, file as string) as boolean	836
* 14.2.7 beginProgressSheetForBurn(burn as DRBurnMBS, track as DRTrackMBS, docWindow as NSWindowMBS)	836

* 14.2.8	beginProgressSheetForBurn(burn as DRBurnMBS, tracks() as DRTrackMBS, docWindow as NSWindowMBS)	837
* 14.2.9	beginProgressSheetForImageFile(burn as DRBurnMBS, file as folderitem, docWindow as NSWindowMBS) as boolean	837
* 14.2.10	beginProgressSheetForImageFile(burn as DRBurnMBS, file as string, docWindow as NSWindowMBS) as boolean	838
* 14.2.11	Constructor	838
* 14.2.12	DRBurnProgressPanelDidFinishNotification as string	838
* 14.2.13	DRBurnProgressPanelWillBeginNotification as string	838
* 14.2.14	stopBurn	839
* 14.2.16	Description as string	839
* 14.2.17	VerboseProgressStatus as boolean	839
* 14.2.19	burnProgressPanelBurnDidFinish(burn as DRBurnMBS) as boolean	840
* 14.2.20	burnProgressPanelDidFinish	840
* 14.2.21	burnProgressPanelWillBegin	840
– 14.3.1	class DRBurnSetupPanelMBS	841
* 14.3.3	appendable	842
* 14.3.4	burnObject as DRBurnMBS	842
* 14.3.5	burnSpeed	842
* 14.3.6	completionAction	842
* 14.3.7	Constructor	843
* 14.3.8	DRBurnSetupPanelDefaultButtonDefaultTitle as string	843
* 14.3.9	expand	843
* 14.3.10	setCanSelectAppendableMedia(flag as boolean)	843
* 14.3.11	setCanSelectTestBurn(flag as boolean)	844
* 14.3.12	setDefaultButtonTitle(title as string)	844
* 14.3.13	testBurn	844
* 14.3.14	verifyBurn	844
– 14.4.1	class DRCDTextBlockMBS	845
* 14.4.3	Constructor	845
* 14.4.4	encoding as Integer	845
* 14.4.5	language as string	845
– 14.5.1	class DRDeviceMBS	846
* 14.5.3	acquireExclusiveAccess as boolean	846
* 14.5.4	acquireMediaReservation	846
* 14.5.5	bsdName as string	847
* 14.5.6	closeTray as boolean	847
* 14.5.7	Constructor	847
* 14.5.8	device(index as UInt32) as DRDeviceMBS	847
* 14.5.9	deviceCount as UInt32	848
* 14.5.10	deviceForBSDName(bsdName as string) as DRDeviceMBS	848

* 14.5.11 deviceForIORegistryEntryPath(path as string) as DRDeviceMBS	848
* 14.5.12 devices as DRDeviceMBS()	848
* 14.5.13 displayName as string	849
* 14.5.14 DRDeviceAppearedNotification as string	849
* 14.5.15 DRDeviceBurnSpeedBD1x as single	849
* 14.5.16 DRDeviceBurnSpeedCD1x as single	849
* 14.5.17 DRDeviceBurnSpeedDVD1x as single	849
* 14.5.18 DRDeviceBurnSpeedHDDVD1x as single	850
* 14.5.19 DRDeviceBurnSpeedMax as single	850
* 14.5.20 DRDeviceBurnSpeedsKey as string	850
* 14.5.21 DRDeviceCanTestWriteCDKey as string	850
* 14.5.22 DRDeviceCanTestWriteDVDKey as string	850
* 14.5.23 DRDeviceCanUnderrunProtectCDKey as string	851
* 14.5.24 DRDeviceCanUnderrunProtectDVDKey as string	851
* 14.5.25 DRDeviceCanWriteBDKey as string	851
* 14.5.26 DRDeviceCanWriteBDREKey as string	851
* 14.5.27 DRDeviceCanWriteBDRKey as string	851
* 14.5.28 DRDeviceCanWriteCDKey as string	851
* 14.5.29 DRDeviceCanWriteCDRawKey as string	852
* 14.5.30 DRDeviceCanWriteCDRKey as string	852
* 14.5.31 DRDeviceCanWriteCDRWKey as string	852
* 14.5.32 DRDeviceCanWriteCDSAOKeY as string	852
* 14.5.33 DRDeviceCanWriteCDTAOKeY as string	852
* 14.5.34 DRDeviceCanWriteCDTextKey as string	853
* 14.5.35 DRDeviceCanWriteDVDDAOKeY as string	853
* 14.5.36 DRDeviceCanWriteDVDKey as string	853
* 14.5.37 DRDeviceCanWriteDVDPlusRDoubleLayerKey as string	853
* 14.5.38 DRDeviceCanWriteDVDPlusRKey as string	853
* 14.5.39 DRDeviceCanWriteDVDPlusRWDoubleLayerKey as string	853
* 14.5.40 DRDeviceCanWriteDVDPlusRWKey as string	854
* 14.5.41 DRDeviceCanWriteDVDRAMKey as string	854
* 14.5.42 DRDeviceCanWriteDVDRDualLayerKey as string	854
* 14.5.43 DRDeviceCanWriteDVDRKey as string	854
* 14.5.44 DRDeviceCanWriteDVDRWDualLayerKey as string	854
* 14.5.45 DRDeviceCanWriteDVDRWKey as string	855
* 14.5.46 DRDeviceCanWriteHDDVDKey as string	855
* 14.5.47 DRDeviceCanWriteHDDVDRAMKey as string	855
* 14.5.48 DRDeviceCanWriteHDDVDRDualLayerKey as string	855
* 14.5.49 DRDeviceCanWriteHDDVDRKey as string	855
* 14.5.50 DRDeviceCanWriteHDDVDRWDualLayerKey as string	855
* 14.5.51 DRDeviceCanWriteHDDVDRWKey as string	856
* 14.5.52 DRDeviceCanWriteIndexPointsKey as string	856

* 14.5.53 DRDeviceCanWriteISRCKey as string	856
* 14.5.54 DRDeviceCanWriteKey as string	856
* 14.5.55 DRDeviceCurrentWriteSpeedKey as string	856
* 14.5.56 DRDeviceDisappearedNotification as string	857
* 14.5.57 DRDeviceFirmwareRevisionKey as string	857
* 14.5.58 DRDeviceIORegistryEntryPathKey as string	857
* 14.5.59 DRDeviceIsBusyKey as string	857
* 14.5.60 DRDeviceIsTrayOpenKey as string	857
* 14.5.61 DRDeviceLoadingMechanismCanEjectKey as string	858
* 14.5.62 DRDeviceLoadingMechanismCanInjectKey as string	858
* 14.5.63 DRDeviceLoadingMechanismCanOpenKey as string	858
* 14.5.64 DRDeviceMaximumWriteSpeedKey as string	858
* 14.5.65 DRDeviceMediaBlocksFreeKey as string	858
* 14.5.66 DRDeviceMediaBlocksOverwritableKey as string	858
* 14.5.67 DRDeviceMediaBlocksUsedKey as string	859
* 14.5.68 DRDeviceMediaBSDNameKey as string	859
* 14.5.69 DRDeviceMediaClassBD as string	859
* 14.5.70 DRDeviceMediaClassCD as string	859
* 14.5.71 DRDeviceMediaClassDVD as string	859
* 14.5.72 DRDeviceMediaClassHDDVD as string	860
* 14.5.73 DRDeviceMediaClassKey as string	860
* 14.5.74 DRDeviceMediaClassUnknown as string	860
* 14.5.75 DRDeviceMediaDoubleLayerL0DataZoneBlocksKey as string	860
* 14.5.76 DRDeviceMediaFreeSpaceKey as string	860
* 14.5.77 DRDeviceMediaInfoKey as string	861
* 14.5.78 DRDeviceMediaIsAppendableKey as string	861
* 14.5.79 DRDeviceMediaIsBlankKey as string	861
* 14.5.80 DRDeviceMediaIsErasableKey as string	861
* 14.5.81 DRDeviceMediaIsOverwritableKey as string	861
* 14.5.82 DRDeviceMediaIsReservedKey as string	862
* 14.5.83 DRDeviceMediaOverwritableSpaceKey as string	862
* 14.5.84 DRDeviceMediaSessionCountKey as string	862
* 14.5.85 DRDeviceMediaStateInTransition as string	862
* 14.5.86 DRDeviceMediaStateKey as string	863
* 14.5.87 DRDeviceMediaStateMediaPresent as string	863
* 14.5.88 DRDeviceMediaStateNone as string	863
* 14.5.89 DRDeviceMediaTrackCountKey as string	863
* 14.5.90 DRDeviceMediaTypeBDR as string	863
* 14.5.91 DRDeviceMediaTypeBDRE as string	864
* 14.5.92 DRDeviceMediaTypeBDROM as string	864
* 14.5.93 DRDeviceMediaTypeCDR as string	864
* 14.5.94 DRDeviceMediaTypeCDROM as string	864

* 14.5.95 DRDeviceMediaTypeCDRW as string	864
* 14.5.96 DRDeviceMediaTypeDVDPlusR as string	864
* 14.5.97 DRDeviceMediaTypeDVDPlusRDoubleLayer as string	865
* 14.5.98 DRDeviceMediaTypeDVDPlusRW as string	865
* 14.5.99 DRDeviceMediaTypeDVDPlusRWDoubleLayer as string	865
* 14.5.100 DRDeviceMediaTypeDVDR as string	865
* 14.5.101 DRDeviceMediaTypeDVDRAM as string	865
* 14.5.102 DRDeviceMediaTypeDVDRDualLayer as string	866
* 14.5.103 DRDeviceMediaTypeDVDROM as string	867
* 14.5.104 DRDeviceMediaTypeDVDRW as string	867
* 14.5.105 DRDeviceMediaTypeDVDRWDualLayer as string	867
* 14.5.106 DRDeviceMediaTypeHDDVDR as string	867
* 14.5.107 DRDeviceMediaTypeHDDVDRAM as string	867
* 14.5.108 DRDeviceMediaTypeHDDVDRDualLayer as string	867
* 14.5.109 DRDeviceMediaTypeHDDVDROM as string	868
* 14.5.110 DRDeviceMediaTypeHDDVDRW as string	868
* 14.5.111 DRDeviceMediaTypeHDDVDRWDualLayer as string	868
* 14.5.112 DRDeviceMediaTypeKey as string	868
* 14.5.113 DRDeviceMediaTypeUnknown as string	868
* 14.5.114 DRDeviceMediaUsedSpaceKey as string	869
* 14.5.115 DRDevicePhysicalInterconnectATAPI as string	869
* 14.5.116 DRDevicePhysicalInterconnectFibreChannel as string	869
* 14.5.117 DRDevicePhysicalInterconnectFireWire as string	869
* 14.5.118 DRDevicePhysicalInterconnectKey as string	869
* 14.5.119 DRDevicePhysicalInterconnectLocationExternal as string	869
* 14.5.120 DRDevicePhysicalInterconnectLocationInternal as string	870
* 14.5.121 DRDevicePhysicalInterconnectLocationKey as string	870
* 14.5.122 DRDevicePhysicalInterconnectLocationUnknown as string	870
* 14.5.123 DRDevicePhysicalInterconnectSCSI as string	870
* 14.5.124 DRDevicePhysicalInterconnectUSB as string	870
* 14.5.125 DRDeviceProductNameKey as string	871
* 14.5.126 DRDeviceStatusChangedNotification as string	871
* 14.5.127 DRDeviceSupportLevelAppleShipping as string	871
* 14.5.128 DRDeviceSupportLevelAppleSupported as string	871
* 14.5.129 DRDeviceSupportLevelKey as string	872
* 14.5.130 DRDeviceSupportLevelNone as string	872
* 14.5.131 DRDeviceSupportLevelUnsupported as string	872
* 14.5.132 DRDeviceSupportLevelVendorSupported as string	872
* 14.5.133 DRDeviceTrackInfoKey as string	872
* 14.5.134 DRDeviceTrackRefsKey as string	872
* 14.5.135 DRDeviceVendorNameKey as string	873
* 14.5.136 DRDeviceWriteBufferSizeKey as string	873

* 14.5.137 DRDeviceWriteCapabilitiesKey as string	873
* 14.5.138 ejectMedia as boolean	873
* 14.5.139 info as dictionary	873
* 14.5.140 ioRegistryEntryPath as string	874
* 14.5.141 isEqualToDevice(value as DRDeviceMBS) as boolean	874
* 14.5.142 isValid as boolean	874
* 14.5.143 mediaIsAppendable as boolean	874
* 14.5.144 mediaIsBlank as boolean	874
* 14.5.145 mediaIsBusy as boolean	875
* 14.5.146 mediaIsErasable as boolean	875
* 14.5.147 mediaIsOverwritable as boolean	875
* 14.5.148 mediaIsPresent as boolean	875
* 14.5.149 mediaIsReserved as boolean	875
* 14.5.150 mediaIsTransitioning as boolean	875
* 14.5.151 mediaSpaceFree as DRMSFMBS	876
* 14.5.152 mediaSpaceOverwritable as DRMSFMBS	876
* 14.5.153 mediaSpaceUsed as DRMSFMBS	876
* 14.5.154 mediaType as string	876
* 14.5.155 openTray as boolean	876
* 14.5.156 PhysicalInterconnect as string	876
* 14.5.157 PhysicalInterconnectLocation as string	877
* 14.5.158 releaseExclusiveAccess	877
* 14.5.159 releaseMediaReservation	877
* 14.5.160 status as dictionary	877
* 14.5.161 trayIsOpen as boolean	877
* 14.5.162 writesCD as boolean	878
* 14.5.163 writesDVD as boolean	878
– 14.6.1 class DREraseMBS	879
* 14.6.3 Constructor(device as DRDeviceMBS)	879
* 14.6.4 device as DRDeviceMBS	879
* 14.6.5 DREraseStatusChangedNotification as string	879
* 14.6.6 DREraseTypeComplete as string	879
* 14.6.7 DREraseTypeKey as string	880
* 14.6.8 DREraseTypeQuick as string	880
* 14.6.9 eraseForDevice(device as DRDeviceMBS) as DREraseMBS	880
* 14.6.10 start	880
* 14.6.11 status as dictionary	880
* 14.6.13 eraseType as string	881
* 14.6.14 properties as dictionary	881
– 14.7.1 class DREraseProgressPanelMBS	882
* 14.7.3 beginProgressPanelForErase(erase as DREraseMBS)	882

* 14.7.4	beginProgressSheetForErase(erase as DREraseMBS, docWindow as NSWindowMBS)	882
* 14.7.5	Constructor	882
* 14.7.6	DREraseProgressPanelDidFinishNotification as string	883
* 14.7.7	DREraseProgressPanelWillBeginNotification as string	883
* 14.7.9	Description as string	883
* 14.7.11	eraseProgressPanelDidFinish	884
* 14.7.12	eraseProgressPanelEraseDidFinish(erase as DREraseMBS) as boolean	884
* 14.7.13	eraseProgressPanelWillBegin	884
– 14.8.1	class DREraseSetupPanelMBS	885
* 14.8.3	Constructor	885
* 14.8.4	eraseObject as DREraseMBS	885
– 14.9.1	class DRFileMBS	886
* 14.9.3	Constructor	886
* 14.9.4	Constructor(name as string)	887
* 14.9.5	Constructor(name as string, data as memoryblock)	887
* 14.9.6	Constructor(path as folderitem)	887
* 14.9.7	DRLinkTypeFinderAlias as string	888
* 14.9.8	DRLinkTypeHardLink as string	888
* 14.9.9	DRLinkTypeSymbolicLink as string	888
* 14.9.10	fileWithPath(path as folderitem) as DRFileMBS	888
* 14.9.11	fileWithPath(path as string) as DRFileMBS	888
* 14.9.12	finderAliasPointingTo(original as DRFSObjectMBS, filesystem as string) as DRFileMBS	889
* 14.9.13	hardLinkPointingTo(original as DRFileMBS, filesystem as string) as DRFileMBS	889
* 14.9.14	linkWithLinkType(linkType as string, original as DRFSObjectMBS, filesystem as string) as DRFileMBS	889
* 14.9.15	symLinkPointingTo(original as DRFSObjectMBS, filesystem as string) as DRFileMBS	890
* 14.9.16	virtualFileWithName(name as string, data as memoryblock) as DRFileMBS	891
* 14.9.18	calculateSizeOfFile(fork as Integer, estimating as boolean) as uint64	891
* 14.9.19	cleanupFileAfterBurn	892
* 14.9.20	prepareFileForBurn as boolean	892
* 14.9.21	prepareFileForVerification as boolean	892
* 14.9.22	produceFile(fork as Integer, buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32) as uint32	892
– 14.10.1	class DRFolderMBS	894
* 14.10.3	addChild(child as DRFSObjectMBS)	894
* 14.10.4	children as DRFSObjectMBS()	894
* 14.10.5	Constructor	895
* 14.10.6	Constructor(name as string)	895

* 14.10.7 Constructor(path as folderitem)	895
* 14.10.8 count as Integer	895
* 14.10.9 folderWithPath(path as folderitem) as DRFolderMBS	896
* 14.10.10 folderWithPath(path as string) as DRFolderMBS	896
* 14.10.11 makeVirtual	896
* 14.10.12 removeChild(child as DRFSObjectMBS)	896
* 14.10.13 virtualFolderWithName(name as string) as DRFolderMBS	896
– 14.11.1 class DRFSObjectMBS	898
* 14.11.3 Constructor	898
* 14.11.4 DRAccessDate as string	898
* 14.11.5 DRAllFilesystems as string	898
* 14.11.6 DRAttributeModificationDate as string	899
* 14.11.7 DRBackupDate as string	899
* 14.11.8 DRContentModificationDate as string	899
* 14.11.9 DRCreationDate as string	899
* 14.11.10 DREffectiveDate as string	899
* 14.11.11 DRExpirationDate as string	899
* 14.11.12 DRHFSPlus as string	900
* 14.11.13 DRHFSPlusCatalogNodeID as string	900
* 14.11.14 DRHFSPlusTextEncodingHint as string	900
* 14.11.15 DRInvisible as string	900
* 14.11.16 DRISO9660 as string	901
* 14.11.17 DRISO9660LevelOne as string	901
* 14.11.18 DRISO9660LevelTwo as string	901
* 14.11.19 DRISO9660VersionNumber as string	901
* 14.11.20 DRJoliet as string	901
* 14.11.21 DRMacExtendedFinderFlags as string	902
* 14.11.22 DRMacFileCreator as string	902
* 14.11.23 DRMacFileType as string	902
* 14.11.24 DRMacFinderFlags as string	902
* 14.11.25 DRMacFinderHideExtension as string	902
* 14.11.26 DRMacIconLocation as string	903
* 14.11.27 DRMacScrollPosition as string	903
* 14.11.28 DRMacWindowBounds as string	903
* 14.11.29 DRMacWindowView as string	903
* 14.11.30 DRPosixFileMode as string	903
* 14.11.31 DRPosixGID as string	904
* 14.11.32 DRPosixUID as string	904
* 14.11.33 DRRecordingDate as string	904
* 14.11.34 DRUDF as string	904
* 14.11.35 DRUDFApplicationIdentifierSuffix as string	904

* 14.11.36 DRUDFExtendedFilePermissions as string	904
* 14.11.37 DRUDFInterchangeLevel as string	905
* 14.11.38 DRUDFMaxInterchangeLevel as string	905
* 14.11.39 DRUDFMaxVolumeSequenceNumber as string	905
* 14.11.40 DRUDFPrimaryVolumeDescriptorNumber as string	905
* 14.11.41 DRUDFRealTimeFile as string	906
* 14.11.42 DRUDFVersion102 as string	906
* 14.11.43 DRUDFVersion150 as string	906
* 14.11.44 DRUDFVolumeSequenceNumber as string	906
* 14.11.45 DRUDFVolumeSetIdentifier as string	906
* 14.11.46 DRUDFVolumeSetImplementationUse as string	906
* 14.11.47 DRUDFVolumeSetTimestamp as string	907
* 14.11.48 DRUDFWriteVersion as string	907
* 14.11.49 effectiveFilesystemMask as Integer	907
* 14.11.50 isVirtual as boolean	907
* 14.11.51 mangledNameForFilesystem(filesystem as string) as string	907
* 14.11.52 mangledNames as dictionary	908
* 14.11.53 parent as DRFolderMBS	908
* 14.11.54 propertiesForFilesystem(filesystem as string, mergeWithOtherFilesystems as boolean) as dictionary	908
* 14.11.55 propertyForKey(key as string, filesystem as string, mergeWithOtherFilesystems as boolean) as Variant	908
* 14.11.56 setProperties(Value as dictionary, filesystem as string)	909
* 14.11.57 setProperty(Value as Variant, key as string, filesystem as string)	909
* 14.11.58 sourcePath as string	910
* 14.11.60 baseName as string	910
* 14.11.61 explicitFilesystemMask as Integer	910
* 14.11.62 specificNameForFilesystem(filesystem as string) as string	910
* 14.11.63 specificNames as dictionary	911
– 14.12.1 class DRMSFMBS	912
* 14.12.3 Constructor	912
* 14.12.4 Constructor(frames as Integer)	912
* 14.12.5 Constructor(s as string)	912
* 14.12.6 description as string	913
* 14.12.7 descriptionWithFormat(format as string) as string	913
* 14.12.8 frames as Integer	914
* 14.12.9 isEqualToMSF(value as DRMSFMBS) as boolean	914
* 14.12.10 minutes as Integer	914
* 14.12.11 msf as DRMSFMBS	914
* 14.12.12 msfByAdding(value as DRMSFMBS) as DRMSFMBS	914
* 14.12.13 msfBySubtracting(value as DRMSFMBS) as DRMSFMBS	914
* 14.12.14 msfWithFrames(frames as Integer) as DRMSFMBS	915

	67
* 14.12.15 msfWithString(s as string) as DRMSFMBS	915
* 14.12.16 seconds as Integer	915
* 14.12.17 sectors as Integer	915
– 14.13.1 class DRSetupPanelMBS	916
* 14.13.3 cancel	916
* 14.13.4 close	916
* 14.13.5 Constructor	916
* 14.13.6 eject	917
* 14.13.7 ok	917
* 14.13.8 open	917
* 14.13.9 runSetupPanel as Integer	917
* 14.13.11 determineBestDevice(deviceA as DRDeviceMBS, deviceB as DRDeviceMBS) as DRDeviceMBS	918
* 14.13.12 DeviceContainsSuitableMedia(device as DRDeviceMBS, byref prompt as string) as boolean	918
* 14.13.13 DeviceCouldBeTarget(device as DRDeviceMBS) as boolean	918
* 14.13.14 DeviceSelectionChanged(device as DRDeviceMBS)	918
* 14.13.15 SetupPanelShouldHandleMediaReservations as boolean	919
– 14.14.1 class DRTrackMBS	920
* 14.14.3 Constructor	921
* 14.14.4 DRAbstractFile as string	921
* 14.14.5 DRApplicationIdentifier as string	921
* 14.14.6 DRAudioFourChannelKey as string	921
* 14.14.7 DRAudioPreEmphasisKey as string	922
* 14.14.8 DRBibliographicFile as string	922
* 14.14.9 DRBlockSize as string	922
* 14.14.10 DRBlockSizeKey as string	922
* 14.14.11 DRBlockTypeKey as string	923
* 14.14.12 DRCopyrightFile as string	923
* 14.14.13 DRDataFormKey as string	923
* 14.14.14 DRDataPreparer as string	923
* 14.14.15 DRDefaultDate as string	923
* 14.14.16 DRDVDCopyrightInfoKey as string	924
* 14.14.17 DRDVDTimestampKey as string	924
* 14.14.18 DRFreeBlocksKey as string	924
* 14.14.19 DRIndexPointsKey as string	924
* 14.14.20 DRISOLevel as string	924
* 14.14.21 DRISOMacExtensions as string	925
* 14.14.22 DRISORockRidgeExtensions as string	925
* 14.14.23 DRMaxBurnSpeedKey as string	925
* 14.14.24 DRNextWritableAddressKey as string	925

* 14.14.25 DRPreGapIsRequiredKey as string	925
* 14.14.26 DRPreGapLengthKey as string	926
* 14.14.27 DRPublisher as string	926
* 14.14.28 DRSCMSCopyrightFree as string	926
* 14.14.29 DRSCMSCopyrightProtectedCopy as string	926
* 14.14.30 DRSCMSCopyrightProtectedOriginal as string	927
* 14.14.31 DRSerialCopyManagementStateKey as string	927
* 14.14.32 DRSessionFormatKey as string	927
* 14.14.33 DRSessionNumberKey as string	927
* 14.14.34 DRSubchannelDataFormKey as string	928
* 14.14.35 DRSubchannelDataFormNone as string	928
* 14.14.36 DRSubchannelDataFormPack as string	928
* 14.14.37 DRSubchannelDataFormRaw as string	928
* 14.14.38 DRSuppressMacSpecificFiles as string	929
* 14.14.39 DRSystemIdentifier as string	929
* 14.14.40 DRTrackIsEmptyKey as string	929
* 14.14.41 DRTrackISRCKey as string	929
* 14.14.42 DRTrackLengthKey as string	930
* 14.14.43 DRTrackModeKey as string	930
* 14.14.44 DRTrackNumberKey as string	930
* 14.14.45 DRTrackPacketSizeKey as string	930
* 14.14.46 DRTrackPacketTypeFixed as string	930
* 14.14.47 DRTrackPacketTypeKey as string	931
* 14.14.48 DRTrackPacketTypeVariable as string	931
* 14.14.49 DRTrackStartAddressKey as string	931
* 14.14.50 DRTrackTypeClosed as string	931
* 14.14.51 DRTrackTypeIncomplete as string	931
* 14.14.52 DRTrackTypeInvisible as string	931
* 14.14.53 DRTrackTypeKey as string	932
* 14.14.54 DRTrackTypeReserved as string	932
* 14.14.55 DRVerificationTypeChecksum as string	932
* 14.14.56 DRVerificationTypeKey as string	932
* 14.14.57 DRVerificationTypeNone as string	932
* 14.14.58 DRVerificationTypeProduceAgain as string	933
* 14.14.59 DRVerificationTypeReceiveData as string	933
* 14.14.60 DRVolumeCheckedDate as string	933
* 14.14.61 DRVolumeCreationDate as string	933
* 14.14.62 DRVolumeEffectiveDate as string	933
* 14.14.63 DRVolumeExpirationDate as string	933
* 14.14.64 DRVolumeModificationDate as string	934
* 14.14.65 DRVolumeSet as string	934
* 14.14.66 estimateLength as UInt64	934

* 14.14.67 testProductionSpeedForInterval(seconds as Double) as Double	934
* 14.14.68 testProductionSpeedForLength(length as Integer) as Double	935
* 14.14.69 trackForAudioFile(path as folderitem) as DRTrackMBS	935
* 14.14.70 trackForAudioFile(path as string) as DRTrackMBS	936
* 14.14.71 trackForRootFolder(folder as DRFolderMBS) as DRTrackMBS	936
* 14.14.72 trackForRootFolder(folder as folderitem) as DRTrackMBS	937
* 14.14.74 BlockSize as Integer	937
* 14.14.75 BlockType as Integer	937
* 14.14.76 DataForm as Integer	937
* 14.14.77 length as DRMSFMBS	937
* 14.14.78 MaxBurnSpeed as Double	938
* 14.14.79 preGap as DRMSFMBS	938
* 14.14.80 PreGapIsRequired as boolean	938
* 14.14.81 PreGapLength as Double	938
* 14.14.82 properties as dictionary	939
* 14.14.83 SessionFormat as Integer	939
* 14.14.84 TrackISRC as memoryblock	939
* 14.14.85 TrackMode as Integer	939
* 14.14.86 VerificationType as string	940
* 14.14.88 cleanupTrackAfterBurn	940
* 14.14.89 cleanupTrackAfterVerification as boolean	940
* 14.14.90 estimateLengthOfTrack as uint64	940
* 14.14.91 prepareTrack(burn as DRBurnMBS) as boolean	941
* 14.14.92 prepareTrackForVerification as boolean	941
* 14.14.93 produceDataForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32) as uint32	941
* 14.14.94 producePreGapForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32) as uint32	942
* 14.14.95 verifyDataForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32) as boolean	943
* 14.14.96 verifyPreGapForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32) as boolean	943

• 9 Cocoa Controls	297
– 9.35.1 class Groupbox	455
* 9.35.3 NSBoxMBS as NSBoxMBS	455
– 9.36.1 class ImageWell	456
* 9.36.3 NSImageViewMBS as NSImageViewMBS	456
– 9.37.1 class KeyValueTypeMBS	457
* 9.37.3 Constructor	457
* 9.37.4 sortedArrayUsingDescriptor(values() as KeyValueTypeMBS, sortDescriptor as NSSortDescriptorMBS) as KeyValueTypeMBS()	457
* 9.37.5 sortedArrayUsingDescriptors(values() as KeyValueTypeMBS, sortDescriptor() as NSSortDescriptorMBS) as KeyValueTypeMBS()	457
* 9.37.7 Description as String	458
* 9.37.8 Handle as Integer	458
* 9.37.9 Tag as Variant	458
* 9.37.10 valueForKey(key as String) as Variant	458
* 9.37.12 Description as String	459
* 9.37.13 setValueForKey(key as string, value as Variant)	459
* 9.37.14 setValueForUndefinedKey(key as string, value as Variant)	459
* 9.37.15 valueForKey(key as string) as Variant	459
* 9.37.16 valueForUndefinedKey(key as string) as Variant	460

	71
• 12 Controls	801
– 12.3.1 class Listbox	803
* 12.3.3 HorizontalNSScrollerMBS as NSScrollerMBS	803
* 12.3.4 VerticalNSScrollerMBS as NSScrollerMBS	803

• 9 Cocoa Controls	297
– 9.38.1 class NSBoxMBS	461
* 9.38.3 borderRect as NSRectMBS	461
* 9.38.4 Constructor	461
* 9.38.5 Constructor(Handle as Integer)	462
* 9.38.6 Constructor(left as Double, top as Double, width as Double, height as Double)	462
* 9.38.7 setFrameFromContentFrame(contentFrame as NSRectMBS)	462
* 9.38.8 setTitleWithMnemonic(stringWithAmpersand as string)	463
* 9.38.9 sizeToFit	463
* 9.38.10 titleCell as NSCellMBS	464
* 9.38.11 titleRect as NSRectMBS	464
* 9.38.13 borderColor as NSColorMBS	464
* 9.38.14 borderType as Integer	464
* 9.38.15 borderWidth as Double	465
* 9.38.16 boxType as Integer	465
* 9.38.17 contentView as NSViewMBS	465
* 9.38.18 contentViewMargins as NSSizeMBS	465
* 9.38.19 cornerRadius as Double	466
* 9.38.20 fillColor as NSColorMBS	466
* 9.38.21 title as string	466
* 9.38.22 titleFont as NSFontMBS	466
* 9.38.23 titlePosition as Integer	467
* 9.38.24 Transparent as Boolean	467
– 9.39.1 control NSButtonControlMBS	469
* 9.39.3 AlternateTitle as String	469
* 9.39.4 BezelStyle as Integer	469
* 9.39.5 ButtonType as Integer	469
* 9.39.6 Title as String	470
* 9.39.7 View as NSButtonMBS	470
* 9.39.9 Action	470
* 9.39.10 BoundsChanged	470
* 9.39.11 Close	471
* 9.39.12 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	471
* 9.39.13 ContextualMenuAction(hitItem as MenuItem) as Boolean	471
* 9.39.14 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	471
* 9.39.15 EnableMenuItems	471
* 9.39.16 FrameChanged	472
* 9.39.17 GotFocus	472
* 9.39.18 LostFocus	472
* 9.39.19 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	472

	73
* 9.39.20 MouseDrag(x as Integer, y as Integer)	473
* 9.39.21 MouseUp(x as Integer, y as Integer)	473
* 9.39.22 Open	473
* 9.39.23 ScaleFactorChanged(NewFactor as Double)	473
* 9.39.24 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	474
– 9.40.1 class NSButtonMBS	475
* 9.40.3 Constructor	475
* 9.40.4 Constructor(Handle as Integer)	476
* 9.40.5 Constructor(left as Double, top as Double, width as Double, height as Double)	476
* 9.40.6 Constructor(Title as String, Image as NSImageMBS = nil, Type as Integer = 0)	476
* 9.40.7 setButtonType(buttonType as Integer)	477
* 9.40.8 setNextState	477
* 9.40.10 allowsMixedState as boolean	477
* 9.40.11 alternateImage as NSImageMBS	478
* 9.40.12 alternateTitle as string	478
* 9.40.13 attributedAlternateTitle as NSAttributedStringMBS	478
* 9.40.14 attributedTitle as NSAttributedStringMBS	478
* 9.40.15 backgroundColor as NSColorMBS	478
* 9.40.16 bezelColor as NSColorMBS	479
* 9.40.17 bezelStyle as Integer	479
* 9.40.18 hasDestructiveAction as Boolean	479
* 9.40.19 image as NSImageMBS	480
* 9.40.20 imageDimsWhenDisabled as Boolean	480
* 9.40.21 imageHugsTitle as Boolean	480
* 9.40.22 imagePosition as Integer	480
* 9.40.23 imageScaling as Integer	481
* 9.40.24 isBordered as boolean	481
* 9.40.25 isTransparent as boolean	481
* 9.40.26 keyEquivalent as string	482
* 9.40.27 keyEquivalentModifierMask as Integer	482
* 9.40.28 maxAcceleratorLevel as Integer	483
* 9.40.29 showsBorderOnlyWhileMouseInside as boolean	484
* 9.40.30 sound as Variant	484
* 9.40.31 SpringLoaded as Boolean	484
* 9.40.32 state as Integer	484
* 9.40.33 title as string	484
– 9.41.1 class NSClipViewMBS	487
* 9.41.3 autoscroll(theEvent as NSEventMBS) as boolean	488
* 9.41.4 constrainScrollPoint(newOrigin as NSPointMBS) as NSPointMBS	488
* 9.41.5 Constructor	488
* 9.41.6 Constructor(Handle as Integer)	489

* 9.41.7 Constructor(left as Double, top as Double, width as Double, height as Double)	489
* 9.41.8 documentRect as NSRectMBS	489
* 9.41.9 documentVisibleRect as NSRectMBS	490
* 9.41.10 reflectScrolledClipView(clipView as NSClipViewMBS)	490
* 9.41.11 scrollClipView(clipview as NSClipViewMBS, toPoint as NSPointMBS)	490
* 9.41.12 scrollToPoint(newOrigin as NSPointMBS)	490
* 9.41.13 viewBoundsChanged(notification as NSNotificationMBS)	490
* 9.41.14 viewFrameChanged(notification as NSNotificationMBS)	491
* 9.41.16 backgroundColor as NSColorMBS	491
* 9.41.17 copiesOnScroll as boolean	491
* 9.41.18 documentCursor as Variant	491
* 9.41.19 documentView as NSViewMBS	492
* 9.41.20 drawsBackground as boolean	492

	75
• 8 Cocoa Collection View	195
– 8.2.1 control NSCollectionViewControlMBS	217
* 8.2.3 performBatchUpdates(tag as variant)	218
* 8.2.5 ScrollView as NSScrollViewMBS	218
* 8.2.6 View as NSCollectionViewMBS	218
* 8.2.8 acceptDrop(draggingInfo as NSDraggingInfoMBS, indexPath as NSIndexPathMBS, drop- Operation as Integer) as Integer	219
* 8.2.9 BoundsChanged	219
* 8.2.10 cancelPrefetchingForItems(indexPaths() as NSIndexPathMBS)	219
* 8.2.11 canDragItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS) as Boolean	220
* 8.2.12 Close	220
* 8.2.13 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	220
* 8.2.14 ContextualMenuItemAction(hitItem as MenuItem) as Boolean	220
* 8.2.15 didChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer)	221
* 8.2.16 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	221
* 8.2.17 didDeselectItems(indexPaths() as NSIndexPathMBS)	221
* 8.2.18 didEndDisplayingItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)	222
* 8.2.19 didEndDisplayingSupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)	222
* 8.2.20 didSelectItems(indexPaths() as NSIndexPathMBS)	222
* 8.2.21 draggingImageForItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS, byref dragImageOffset as NSPointMBS) as NSImageMBS	223
* 8.2.22 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, dragOperation as Integer)	223
* 8.2.23 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, indexPaths() as NSIndexPathMBS)	224
* 8.2.24 EnableMenuItems	224
* 8.2.25 FrameChanged	224
* 8.2.26 GotFocus	224
* 8.2.27 insetForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as NSEdgeInsetsMBS	225
* 8.2.28 ItemApplyLayoutAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS)	225
* 8.2.29 ItemDidTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollection- ViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)	225
* 8.2.30 itemForRepresentedObject(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS	226
* 8.2.31 ItemPreferredLayoutAttributesFittingAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS) as NSCollectionViewLayoutAt- tributesMBS	226

* 8.2.32	ItemPrepareForReuse(item as NSCollectionViewItemMBS)	227
* 8.2.33	ItemWillTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)	227
* 8.2.34	LostFocus	227
* 8.2.35	minimumInteritemSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double	228
* 8.2.36	minimumLineSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double	228
* 8.2.37	MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	229
* 8.2.38	MouseDown(x as Integer, y as Integer)	229
* 8.2.39	MouseUp(x As Integer, y As Integer)	229
* 8.2.40	namesOfPromisedFilesDroppedAtDestination(dropURL as String, indexPaths() as NSIndexPathMBS) as String()	229
* 8.2.41	numberOfItemsInSection(section as Integer) as Integer	230
* 8.2.42	numberOfSections as Integer	230
* 8.2.43	Open	230
* 8.2.44	pasteboardWriterForItem(indexPath as NSIndexPathMBS) as NSPasteboardItemMBS	231
* 8.2.45	performBatchUpdatesCompleted(tag as variant, finished as boolean)	231
* 8.2.46	performBatchUpdatesWork(tag as variant)	231
* 8.2.47	prefetchItems(indexPaths() as NSIndexPathMBS)	232
* 8.2.48	referenceSizeForFooterInSection(layout as NSCollectionViewLayoutMBS, section as Integer) as NSSizeMBS	232
* 8.2.49	referenceSizeForHeaderInSection(layout as NSCollectionViewLayoutMBS, section as Integer) as NSSizeMBS	232
* 8.2.50	ScaleFactorChanged(NewFactor as double)	233
* 8.2.51	shouldChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer) as NSIndexPathMBS()	233
* 8.2.52	shouldDeselectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()	234
* 8.2.53	shouldSelectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()	234
* 8.2.54	sizeForItemAtIndexPath(layout as NSCollectionViewLayoutMBS, indexPath as NSIndexPathMBS) as NSSizeMBS	235
* 8.2.55	transitionLayout(fromLayout as NSCollectionViewLayoutMBS, toLayout as NSCollectionViewLayoutMBS) as NSCollectionViewTransitionLayoutMBS	235
* 8.2.56	updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)	236
* 8.2.57	validateDrop(draggingInfo as NSDraggingInfoMBS, byref proposedIndexPath as NSIndexPathMBS, byref dropOperation as Integer) as Integer	236
* 8.2.58	viewForSupplementaryElement(kind as String, indexPath as NSIndexPathMBS) as NSViewMBS	237
* 8.2.59	willDisplayItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)	237
* 8.2.60	willDisplaySupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)	238
* 8.2.61	willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	238

* 8.2.62 writeItems(indexPaths() as NSIndexPathMBS, Pasteboard as NSPasteboardMBS) as Boolean	238
– 8.3.1 class NSCollectionViewFlowLayoutInvalidationContextMBS	240
* 8.3.3 Constructor	240
* 8.3.5 invalidateFlowLayoutAttributes as Boolean	240
* 8.3.6 invalidateFlowLayoutDelegateMetrics as Boolean	241
– 8.4.1 class NSCollectionViewFlowLayoutMBS	242
* 8.4.3 collapseSectionAtIndex(sectionIndex as Integer)	242
* 8.4.4 Constructor	242
* 8.4.5 ElementKindSectionFooter as String	243
* 8.4.6 ElementKindSectionHeader as String	243
* 8.4.7 expandSectionAtIndex(sectionIndex as Integer)	243
* 8.4.9 estimatedItemSize as NSSizeMBS	243
* 8.4.10 footerReferenceSize as NSSizeMBS	243
* 8.4.11 headerReferenceSize as NSSizeMBS	244
* 8.4.12 itemSize as NSSizeMBS	244
* 8.4.13 minimumInteritemSpacing as Double	244
* 8.4.14 minimumLineSpacing as Double	245
* 8.4.15 scrollDirection as Integer	245
* 8.4.16 sectionFootersPinToVisibleBounds as Boolean	245
* 8.4.17 sectionHeadersPinToVisibleBounds as Boolean	246
* 8.4.18 sectionInset as NSEdgeInsetsMBS	246
* 8.4.19 sectionAtIndexIsCollapsed(sectionIndex as Integer) as Boolean	246
– 8.5.1 class NSCollectionViewGridLayoutMBS	248
* 8.5.3 backgroundColors as NSColorMBS()	248
* 8.5.4 Constructor	249
* 8.5.5 setBackgroundColors(Colors() as NSColorMBS = nil)	249
* 8.5.7 margins as NSEdgeInsetsMBS	249
* 8.5.8 maximumItemSize as NSSizeMBS	250
* 8.5.9 maximumNumberOfColumns as Integer	250
* 8.5.10 maximumNumberOfRows as Integer	250
* 8.5.11 minimumInteritemSpacing as Double	250
* 8.5.12 minimumItemSize as NSSizeMBS	251
* 8.5.13 minimumLineSpacing as Double	251
– 8.6.1 class NSCollectionViewItemMBS	252
* 8.6.3 Constructor	252
* 8.6.4 copy as NSCollectionViewItemMBS	252
* 8.6.6 collectionView as NSCollectionViewMBS	252
* 8.6.7 highlightState as Integer	253
* 8.6.8 identifier as String	253
* 8.6.9 imageView as NSImageViewMBS	253

* 8.6.10 selected as Boolean	253
* 8.6.11 textField as NSTextFieldMBS	253
– 8.7.1 class NSCollectionViewLayoutAttributesMBS	255
* 8.7.3 Constructor	255
* 8.7.4 copy as NSCollectionViewLayoutAttributesMBS	255
* 8.7.5 ElementKindInterItemGapIndicator as String	255
* 8.7.6 layoutAttributesForDecorationView(decorationViewKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS	256
* 8.7.7 layoutAttributesForInterItemGapBefore(indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS	256
* 8.7.8 layoutAttributesForItem(indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS	256
* 8.7.9 layoutAttributesForSupplementaryView(elementKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS	257
* 8.7.11 alpha as Double	257
* 8.7.12 frame as NSSizeMBS	258
* 8.7.13 Hidden as Boolean	258
* 8.7.14 indexPath as NSIndexPathMBS	258
* 8.7.15 representedElementCategory as Integer	258
* 8.7.16 representedElementKind as String	258
* 8.7.17 size as NSSizeMBS	259
* 8.7.18 zIndex as Integer	259
– 8.8.1 class NSCollectionViewLayoutInvalidationContextMBS	260
* 8.8.3 Constructor	260
* 8.8.4 invalidateDecorationElementsOfKind(elementKind as String, indexPaths() as NSIndexPathMBS)	260
* 8.8.5 invalidatedItemIndexPaths as NSIndexPathMBS()	261
* 8.8.6 invalidateItemsAtIndexPaths(indexPaths() as NSIndexPathMBS)	261
* 8.8.7 invalidateSupplementaryElementsOfKind(elementKind as String, indexPaths() as NSIndexPathMBS)	261
* 8.8.9 contentOffsetAdjustment as NSPointMBS	261
* 8.8.10 contentSizeAdjustment as NSSizeMBS	262
* 8.8.11 invalidateDataSourceCounts as Boolean	262
* 8.8.12 invalidatedDecorationIndexPaths as Dictionary	262
* 8.8.13 invalidatedSupplementaryIndexPaths as Dictionary	263
* 8.8.14 invalidateEverything as Boolean	263
– 8.9.1 class NSCollectionViewLayoutMBS	264
* 8.9.3 Constructor	264
* 8.9.4 invalidateLayout	264
* 8.9.5 invalidateLayoutWithContext(context as NSCollectionViewLayoutInvalidationContextMBS)	264
* 8.9.6 NewInvalidationContext as NSCollectionViewLayoutInvalidationContextMBS	265

	79
* 8.9.7 NewLayoutAttributes as NSCollectionViewLayoutAttributesMBS	265
* 8.9.9 collectionView as NSCollectionViewMBS	265
– 8.10.1 class NSCollectionViewMBS	266
* 8.10.3 backgroundColors as NSColorMBS()	266
* 8.10.4 Constructor	266
* 8.10.5 Constructor(Handle as Integer)	267
* 8.10.6 Constructor(left as double, top as double, width as double, height as double)	267
* 8.10.7 deleteItems(indexPaths() as NSIndexPathMBS)	267
* 8.10.8 deleteSections(sections as NSIndexSetMBS)	268
* 8.10.9 deselectAll	268
* 8.10.10 deselectItems(indexPaths() as NSIndexPathMBS)	269
* 8.10.11 frameForItem(Index as Integer) as NSRectMBS	269
* 8.10.12 frameForItem(Index as Integer, numberOfItems as Integer) as NSRectMBS	269
* 8.10.13 indexPathForItem(indexPath as NSCollectionViewItemMBS) as NSIndexPathMBS	270
* 8.10.14 indexPathForItem(x as double, y as double) as NSIndexPathMBS	270
* 8.10.15 indexPathsForVisibleItems as NSIndexPathMBS()	271
* 8.10.16 indexPathsForVisibleSupplementaryElementsOfKind(elementKind as String) as NSIndexPathMBS()	271
* 8.10.17 insertItems(indexPaths() as NSIndexPathMBS)	271
* 8.10.18 insertSections(sections as NSIndexSetMBS)	272
* 8.10.19 item(index as Integer) as NSCollectionViewItemMBS	272
* 8.10.20 item(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS	272
* 8.10.21 layoutAttributesForItem(indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS	273
* 8.10.22 layoutAttributesForSupplementaryElementOfKind(kind as String, indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS	273
* 8.10.23 makeItem(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS	273
* 8.10.24 makeSupplementaryViewOfKind(elementKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewSectionHeaderViewMBS	274
* 8.10.25 moveItem(indexPath as NSIndexPathMBS, toIndexPath as NSIndexPathMBS)	274
* 8.10.26 moveSection(section as Integer, toSection as Integer)	275
* 8.10.27 numberOfItemsInSection(section as Integer) as Integer	275
* 8.10.28 reloadData	275
* 8.10.29 reloadData(indexPaths() as NSIndexPathMBS)	276
* 8.10.30 reloadData(sections as NSIndexSetMBS)	276
* 8.10.31 scrollToItems(indexPaths() as NSIndexPathMBS, scrollPosition as Integer)	276
* 8.10.32 selectAll	277
* 8.10.33 selection as NSIndexPathMBS()	277
* 8.10.34 selectItems(indexPaths() as NSIndexPathMBS, scrollPosition as Integer)	277
* 8.10.35 setBackgroundColors(Colors() as NSColorMBS = nil)	278
* 8.10.36 setDraggingSourceOperationMask(mask as integer, local as boolean)	278

* 8.10.37	setSelection(indexPaths() as NSIndexPathMBS)	278
* 8.10.38	supplementaryViewForElementKind(elementKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewSectionHeaderViewMBS	279
* 8.10.39	toggleSectionCollapse(sender as NSViewMBS)	279
* 8.10.40	visibleItems as NSCollectionViewItemMBS()	279
* 8.10.41	visibleSupplementaryViewsOfKind(elementKind as String) as NSViewMBS()	279
* 8.10.43	allowsEmptySelection as Boolean	280
* 8.10.44	allowsMultipleSelection as Boolean	280
* 8.10.45	animator as NSCollectionViewMBS	280
* 8.10.46	backgroundView as NSViewMBS	281
* 8.10.47	backgroundViewScrollsWithContent as Boolean	281
* 8.10.48	collectionViewLayout as NSCollectionViewLayoutMBS	282
* 8.10.49	firstResponder as Boolean	282
* 8.10.50	numberOfSections as Integer	282
* 8.10.51	selectable as Boolean	283
– 8.11.1	class NSCollectionViewSectionHeaderViewMBS	285
* 8.11.3	Constructor	285
* 8.11.4	Constructor(Handle as Integer)	285
* 8.11.5	Constructor(left as double, top as double, width as double, height as double)	286
* 8.11.7	sectionCollapseButton as NSButtonMBS	286
– 8.12.1	class NSCollectionViewTransitionLayoutMBS	287
* 8.12.3	Constructor	287
* 8.12.4	Constructor(currentLayout as NSCollectionViewLayoutMBS, nextLayout as NSCollectionViewLayoutMBS)	287
* 8.12.6	currentLayout as NSCollectionViewLayoutMBS	288
* 8.12.7	nextLayout as NSCollectionViewLayoutMBS	288
* 8.12.8	transitionProgress as Double	288
* 8.12.9	valueForAnimatedKey(AnimatedKey as String) as Double	288
– 8.13.1	class NSCollectionViewUpdateItemMBS	290
* 8.13.3	Constructor	290
* 8.13.5	indexPathAfterUpdate as NSIndexPathMBS	290
* 8.13.6	indexPathBeforeUpdate as NSIndexPathMBS	290
* 8.13.7	updateAction as Integer	291

	81
• 9 Cocoa Controls	297
– 9.42.1 control NSComboBoxControlMBS	494
* 9.42.3 View as NSComboBoxMBS	494
* 9.42.5 Action	494
* 9.42.6 BoundsChanged	495
* 9.42.7 Close	495
* 9.42.8 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	495
* 9.42.9 ContextualMenuAction(hitItem as MenuItem) as Boolean	495
* 9.42.10 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	495
* 9.42.11 EnableMenuItems	496
* 9.42.12 FrameChanged	496
* 9.42.13 GotFocus	496
* 9.42.14 LostFocus	496
* 9.42.15MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	496
* 9.42.16 MouseDrag(x as Integer, y as Integer)	497
* 9.42.17 MouseUp(x As Integer, y As Integer)	497
* 9.42.18 Open	497
* 9.42.19 ScaleFactorChanged(NewFactor as double)	497
* 9.42.20 TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	498
* 9.42.21 TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	498
* 9.42.22 TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	498
* 9.42.23 textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	499
* 9.42.24 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean	499
* 9.42.25 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	499
– 9.43.1 class NSComboBoxMBS	500
* 9.43.3 addItemWithObjectValue(value as Variant)	500
* 9.43.4 Constructor	500
* 9.43.5 Constructor(Handle as Integer)	500
* 9.43.6 Constructor(left as Double, top as Double, width as Double, height as Double)	501
* 9.43.7 deselectItemAtIndex(index as Integer)	501
* 9.43.8 indexOfItemWithObjectValue(value as Variant) as Integer	502
* 9.43.9 indexOfSelectedItem as Integer	502
* 9.43.10 noteNumberOfItemsChanged	502
* 9.43.11 numberOfItems as Integer	503
* 9.43.12 reloadData	503
* 9.43.13 removeAllItems	503
* 9.43.14 removeItemAtIndex(index as Integer)	503
* 9.43.15 removeItemWithObjectValue(value as Variant)	503

* 9.43.16	scrollItemAtIndexToTop(index as Integer)	504
* 9.43.17	scrollItemAtIndexToVisible(index as Integer)	504
* 9.43.18	selectItemAtIndex(index as Integer)	504
* 9.43.19	selectItemWithObjectValue(value as Variant)	504
* 9.43.21	completes as boolean	505
* 9.43.22	hasVerticalScroller as boolean	505
* 9.43.23	intercellSpacing as NSSizeMBS	505
* 9.43.24	isButtonBordered as boolean	505
* 9.43.25	itemHeight as Double	506
* 9.43.26	numberOfVisibleItems as Integer	506
* 9.43.27	usesDataSource as boolean	506
– 9.44.1	control NSDatePickerControlMBS	507
* 9.44.3	AcceptTabs as Boolean	507
* 9.44.4	View as NSDatePickerMBS	507
* 9.44.6	Action	507
* 9.44.7	BoundsChanged	508
* 9.44.8	Close	508
* 9.44.9	ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	508
* 9.44.10	ContextualMenuItemAction(hitItem as MenuItem) as Boolean	508
* 9.44.11	didCloseContextualMenu(menu as NSMenuItemMBS, NSEvent as NSEventMBS)	508
* 9.44.12	EnableMenuItems	509
* 9.44.13	FrameChanged	509
* 9.44.14	GotFocus	509
* 9.44.15	LostFocus	509
* 9.44.16	MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	509
* 9.44.17	MouseDown(x as Integer, y as Integer)	510
* 9.44.18	MouseUp(x as Integer, y as Integer)	510
* 9.44.19	Open	510
* 9.44.20	ScaleFactorChanged(NewFactor as Double)	510
* 9.44.21	willShowContextualMenu(menu as NSMenuItemMBS, NSEvent as NSEventMBS)	511
– 9.45.1	class NSDatePickerMBS	512
* 9.45.3	Constructor	512
* 9.45.4	Constructor(Handle as Integer)	512
* 9.45.5	Constructor(left as Double, top as Double, width as Double, height as Double)	513
* 9.45.7	backgroundColor as NSColorMBS	513
* 9.45.8	Bezeled as Boolean	513
* 9.45.9	Bordered as Boolean	514
* 9.45.10	calendar as NSCalendarMBS	514
* 9.45.11	datePickerElements as Integer	514
* 9.45.12	datePickerMode as Integer	514

* 9.45.13 datePickerStyle as Integer	514
* 9.45.14 dateTimeValue as dateTime	515
* 9.45.15 dateValue as date	515
* 9.45.16 drawsBackground as Boolean	515
* 9.45.17 locale as NSLocaleMBS	515
* 9.45.18 maxDate as date	515
* 9.45.19 maxDateTime as dateTime	516
* 9.45.20 minDate as date	516
* 9.45.21 minDateTime as dateTime	516
* 9.45.22 textColor as NSColorMBS	516
* 9.45.23 timeInterval as Double	516
* 9.45.24 timeZone as NSTimeZoneMBS	517

- **15 Drag & Drop** 947
 - ?? Globals ??
 - * 15.1.1 InstallDragImageMBS 947
 - * 15.1.2 SetNextDragImageMBS(Img as NSImageMBS, DragItemCount as Integer = 1) 947

	85
• 10 Cocoa Networking	753
– 10.1.1 class NSHTTPCookieMBS	753
* 10.1.3 Constructor(properties as dictionary)	754
* 10.1.4 cookiesWithResponseHeaderFields(headerFields as dictionary, URL as string) as NSHTTPCookieMBS()	755
* 10.1.5 cookieWithProperties(dic as dictionary) as NSHTTPCookieMBS	755
* 10.1.6 NSHTTPCookieComment as string	756
* 10.1.7 NSHTTPCookieCommentURL as string	756
* 10.1.8 NSHTTPCookieDiscard as string	757
* 10.1.9 NSHTTPCookieDomain as string	757
* 10.1.10 NSHTTPCookieExpires as string	757
* 10.1.11 NSHTTPCookieMaximumAge as string	757
* 10.1.12 NSHTTPCookieName as string	757
* 10.1.13 NSHTTPCookieOriginURL as string	758
* 10.1.14 NSHTTPCookiePath as string	758
* 10.1.15 NSHTTPCookiePort as string	758
* 10.1.16 NSHTTPCookieSecure as string	758
* 10.1.17 NSHTTPCookieValue as string	758
* 10.1.18 NSHTTPCookieVersion as string	759
* 10.1.19 portList as Integer()	759
* 10.1.20 requestHeaderFieldsWithCookies(cookies() as NSHTTPCookieMBS) as dictionary	760
* 10.1.22 comment as string	761
* 10.1.23 commentURL as string	761
* 10.1.24 domain as string	762
* 10.1.25 expiresDate as date	762
* 10.1.26 expiresDateTime as dateTime	763
* 10.1.27 Handle as Integer	763
* 10.1.28 isHTTPOnly as boolean	763
* 10.1.29 isSecure as boolean	764
* 10.1.30 isSessionOnly as boolean	764
* 10.1.31 name as string	765
* 10.1.32 path as string	765
* 10.1.33 properties as dictionary	765
* 10.1.34 value as string	766
* 10.1.35 version as Integer	767
– 10.2.1 class NSHTTPCookieStorageMBS	768
* 10.2.3 Constructor	768
* 10.2.4 cookies as NSHTTPCookieMBS()	769
* 10.2.5 cookiesForURL(URL as string) as NSHTTPCookieMBS()	769
* 10.2.6 cookiesToArray(cookies() as NSHTTPCookieMBS) as Integer	770

* 10.2.7 deleteCookie(cookie as NSHTTPCookieMBS)	770
* 10.2.8 NSHTTPCookieManagerAcceptPolicyChangedNotification as string	770
* 10.2.9 NSHTTPCookieManagerCookiesChangedNotification as string	771
* 10.2.10 removeCookiesSinceDate(d as date)	771
* 10.2.11 removeCookiesSinceDate(d as dateTime)	771
* 10.2.12 setCookie(cookie as NSHTTPCookieMBS)	771
* 10.2.13 setCookies(cookies() as NSHTTPCookieMBS, URL as string, mainDocumentURL as string)	772
* 10.2.14 sharedHTTPCookieStorage as NSHTTPCookieStorageMBS	772
* 10.2.16 Handle as Integer	772
* 10.2.17 cookieAcceptPolicy as Integer	773

	87
• 9 Cocoa Controls	297
– 9.46.1 class <code>NSImageViewMBS</code>	518
* 9.46.3 Constructor	518
* 9.46.4 Constructor(Handle as Integer)	518
* 9.46.5 Constructor(left as Double, top as Double, width as Double, height as Double)	519
* 9.46.7 <code>allowsCutCopyPaste</code> as Boolean	519
* 9.46.8 <code>animates</code> as Boolean	520
* 9.46.9 <code>image</code> as <code>NSImageMBS</code>	520
* 9.46.10 <code>imageAlignment</code> as Integer	520
* 9.46.11 <code>imageFrameStyle</code> as Integer	520
* 9.46.12 <code>imageScaling</code> as Integer	521
* 9.46.13 <code>isEditable</code> as Boolean	521

• 8 Cocoa Collection View	195
– 8.14.1 class NSIndexPathMBS	292
* 8.14.3 compare(other as NSIndexPathMBS) as Integer	292
* 8.14.4 Constructor(Index as Integer)	292
* 8.14.5 Constructor(Indexes()) as Integer	292
* 8.14.6 copy as NSIndexPathMBS	293
* 8.14.7 indexAtPosition(position as Integer) as Integer	293
* 8.14.8 indexes as Integer()	293
* 8.14.9 indexPathByAddingIndex(index as Integer) as NSIndexPathMBS	293
* 8.14.10 indexPathByRemovingLastIndex as NSIndexPathMBS	293
* 8.14.11 indexPathForItem(item as Integer, section as Integer = 0) as NSIndexPathMBS	294
* 8.14.12 indexPathWithIndex(index as Integer) as NSIndexPathMBS	294
* 8.14.13 indexPathWithIndexes(indexes() as Integer) as NSIndexPathMBS	294
* 8.14.15 item as Integer	294
* 8.14.16 length as Integer	295
* 8.14.17 section as Integer	295

	89
• 7 Cocoa	171
– 7.1.1 class NSLevelIndicatorMBS	171
* 7.1.3 Constructor	172
* 7.1.4 Constructor(Handle as Integer)	172
* 7.1.5 Constructor(left as Double, top as Double, width as Double, height as Double)	172
* 7.1.6 rectOfTickMarkAtIndex(index as Integer) as NSRectMBS	173
* 7.1.7 tickMarkValueAtIndex(index as Integer) as Double	173
* 7.1.9 criticalValue as Double	173
* 7.1.10 levelIndicatorStyle as Integer	173
* 7.1.11 maxValue as Double	174
* 7.1.12 minValue as Double	174
* 7.1.13 numberOfMajorTickMarks as Integer	174
* 7.1.14 numberOfTickMarks as Integer	174
* 7.1.15 tickMarkPosition as Integer	174
* 7.1.16 warningValue as Double	175

• 9 Cocoa Controls	297
– 9.47.1 control NSOutlineControlMBS	524
* 9.47.3 AcceptTabs as Boolean	524
* 9.47.4 allowsColumnReordering as Boolean	524
* 9.47.5 allowsColumnResizing as Boolean	525
* 9.47.6 allowsColumnSelection as Boolean	525
* 9.47.7 allowsEmptySelection as Boolean	525
* 9.47.8 allowsMultipleSelection as Boolean	525
* 9.47.9 autohidesScrollers as Boolean	526
* 9.47.10 hasHorizontalScroller as Boolean	526
* 9.47.11 hasVerticalScroller as Boolean	526
* 9.47.12 ScrollView as NSScrollViewMBS	526
* 9.47.13 View as NSOutlineViewMBS	526
* 9.47.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean	527
* 9.47.16 BoundsChanged	527
* 9.47.17 childOffItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	527
* 9.47.18 Close	528
* 9.47.19 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)	528
* 9.47.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	528
* 9.47.21 concludeDragOperation(info as NSDraggingInfoMBS)	528
* 9.47.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	529
* 9.47.23 ContextualMenuAction(hitItem as MenuItem) as Boolean	529
* 9.47.24 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS	529
* 9.47.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	530
* 9.47.26 didClickTableColumn(tableColumn as NSTableColumnMBS)	530
* 9.47.28 didDragTableColumn(tableColumn as NSTableColumnMBS)	530
* 9.47.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	531
* 9.47.30 didTile	531
* 9.47.31 DoubleClick	531
* 9.47.32 draggingEnded(info as NSDraggingInfoMBS)	531
* 9.47.33 draggingExited(info as NSDraggingInfoMBS)	531
* 9.47.34 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	532
* 9.47.35 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)	532
* 9.47.36 EnableMenuItems	533

* 9.47.37 FrameChanged	533
* 9.47.38 GotFocus	533
* 9.47.39 heightForRowByItem(item as NSOutlineViewItemMBS) as Double	533
* 9.47.40 isGroupItem(item as NSOutlineViewItemMBS) as Boolean	534
* 9.47.41 isItemExpandable(item as NSOutlineViewItemMBS) as Boolean	534
* 9.47.42 ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	535
* 9.47.43 ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	535
* 9.47.44 itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS	535
* 9.47.45 ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	536
* 9.47.46 ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	536
* 9.47.47 LeftMouseDown(e as NSEventMBS) as Boolean	536
* 9.47.48 LeftMouseDragged(e as NSEventMBS) as Boolean	536
* 9.47.49 LeftMouseUp(e as NSEventMBS) as Boolean	536
* 9.47.50 LostFocus	537
* 9.47.51 MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean	537
* 9.47.52 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	537
* 9.47.53 MouseDrag(x as Integer, y as Integer)	537
* 9.47.54 MouseUp(x as Integer, y as Integer)	538
* 9.47.55 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()	538
* 9.47.56 nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS	538
* 9.47.57 numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	539
* 9.47.58 objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant	539
* 9.47.59 Open	539
* 9.47.60 OtherMouseDown(e as NSEventMBS) as Boolean	539
* 9.47.61 OtherMouseDragged(e as NSEventMBS) as Boolean	540
* 9.47.62 OtherMouseUp(e as NSEventMBS) as Boolean	540
* 9.47.63 pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS	540
* 9.47.64 persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant	540
* 9.47.65 RightMouseDown(e as NSEventMBS) as Boolean	541
* 9.47.66 RightMouseDragged(e as NSEventMBS) as Boolean	541
* 9.47.67 RightMouseUp(e as NSEventMBS) as Boolean	541
* 9.47.68 rowViewForItem(item as NSOutlineViewItemMBS) as NSTableRowViewMBS	541
* 9.47.69 ScaleFactorChanged(NewFactor as Double)	542
* 9.47.70 SelectionDidChange(notification as NSNotificationMBS)	542

* 9.47.71 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	542
* 9.47.72 SelectionIsChanging(notification as NSNotificationMBS)	543
* 9.47.73 selectionShouldChangeInOutlineView as Boolean	543
* 9.47.74 setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)	543
* 9.47.75 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean	543
* 9.47.76 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean	544
* 9.47.77 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean	544
* 9.47.78 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean	545
* 9.47.79 shouldReorderColumn(columnIndex as Integer, newIndex as Integer) as Boolean	545
* 9.47.80 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean	545
* 9.47.81 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean	546
* 9.47.82 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOut- lineViewItemMBS) as Boolean	546
* 9.47.83 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean	546
* 9.47.84 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean	547
* 9.47.85 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean	547
* 9.47.86 sizeToFitWidthOfColumn(column as Integer) as Double	547
* 9.47.87 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)	548
* 9.47.88 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean	548
* 9.47.89 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean	548
* 9.47.90 tooltipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTa- bleColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String	548
* 9.47.91 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String	549
* 9.47.92 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)	549
* 9.47.93 validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer	550
* 9.47.94 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS	550
* 9.47.95 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)	551
* 9.47.96 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)	551
* 9.47.98 willTile	552
* 9.47.99 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean	552

	93
• 6 AVFoundation	169
– 9.47.1 control NSOutlineControlMBS	524
* 9.47.27 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	530
* 9.47.97 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	551

• 9 Cocoa Controls	297
– 9.47.1 control NSOutlineControlMBS	524
* 9.47.3 AcceptTabs as Boolean	524
* 9.47.4 allowsColumnReordering as Boolean	524
* 9.47.5 allowsColumnResizing as Boolean	525
* 9.47.6 allowsColumnSelection as Boolean	525
* 9.47.7 allowsEmptySelection as Boolean	525
* 9.47.8 allowsMultipleSelection as Boolean	525
* 9.47.9 autohidesScrollers as Boolean	526
* 9.47.10 hasHorizontalScroller as Boolean	526
* 9.47.11 hasVerticalScroller as Boolean	526
* 9.47.12 ScrollView as NSScrollViewMBS	526
* 9.47.13 View as NSOutlineViewMBS	526
* 9.47.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean	527
* 9.47.16 BoundsChanged	527
* 9.47.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	527
* 9.47.18 Close	528
* 9.47.19 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)	528
* 9.47.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	528
* 9.47.21 concludeDragOperation(info as NSDraggingInfoMBS)	528
* 9.47.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	529
* 9.47.23 ContextualMenuAction(hitItem as MenuItem) as Boolean	529
* 9.47.24 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS	529
* 9.47.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	530
* 9.47.26 didClickTableColumn(tableColumn as NSTableColumnMBS)	530
* 9.47.28 didDragTableColumn(tableColumn as NSTableColumnMBS)	530
* 9.47.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	531
* 9.47.30 didTile	531
* 9.47.31 DoubleClick	531
* 9.47.32 draggingEnded(info as NSDraggingInfoMBS)	531
* 9.47.33 draggingExited(info as NSDraggingInfoMBS)	531
* 9.47.34 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	532
* 9.47.35 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)	532
* 9.47.36 EnableMenuItems	533

* 9.47.37 FrameChanged	533
* 9.47.38 GotFocus	533
* 9.47.39 heightOfRowByItem(item as NSOutlineViewItemMBS) as Double	533
* 9.47.40 isGroupItem(item as NSOutlineViewItemMBS) as Boolean	534
* 9.47.41 isItemExpandable(item as NSOutlineViewItemMBS) as Boolean	534
* 9.47.42 ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	535
* 9.47.43 ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	535
* 9.47.44 itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS	535
* 9.47.45 ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	536
* 9.47.46 ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	536
* 9.47.47 LeftMouseDown(e as NSEventMBS) as Boolean	536
* 9.47.48 LeftMouseDragged(e as NSEventMBS) as Boolean	536
* 9.47.49 LeftMouseUp(e as NSEventMBS) as Boolean	536
* 9.47.50 LostFocus	537
* 9.47.51 MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean	537
* 9.47.52 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	537
* 9.47.53 MouseDrag(x as Integer, y as Integer)	537
* 9.47.54 MouseUp(x as Integer, y as Integer)	538
* 9.47.55 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()	538
* 9.47.56 nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS	538
* 9.47.57 numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	539
* 9.47.58 objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant	539
* 9.47.59 Open	539
* 9.47.60 OtherMouseDown(e as NSEventMBS) as Boolean	539
* 9.47.61 OtherMouseDragged(e as NSEventMBS) as Boolean	540
* 9.47.62 OtherMouseUp(e as NSEventMBS) as Boolean	540
* 9.47.63 pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS	540
* 9.47.64 persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant	540
* 9.47.65 RightMouseDown(e as NSEventMBS) as Boolean	541
* 9.47.66 RightMouseDragged(e as NSEventMBS) as Boolean	541
* 9.47.67 RightMouseUp(e as NSEventMBS) as Boolean	541
* 9.47.68 rowViewForItem(item as NSOutlineViewItemMBS) as NSTableRowViewMBS	541
* 9.47.69 ScaleFactorChanged(NewFactor as Double)	542
* 9.47.70 SelectionDidChange(notification as NSNotificationMBS)	542

* 9.47.71 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	542
* 9.47.72 SelectionIsChanging(notification as NSNotificationMBS)	543
* 9.47.73 selectionShouldChangeInOutlineView as Boolean	543
* 9.47.74 setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)	543
* 9.47.75 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean	543
* 9.47.76 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean	544
* 9.47.77 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean	544
* 9.47.78 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean	545
* 9.47.79 shouldReorderColumn(columnIndex as Integer, newIndex as Integer) as Boolean	545
* 9.47.80 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean	545
* 9.47.81 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean	546
* 9.47.82 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOut- lineViewItemMBS) as Boolean	546
* 9.47.83 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean	546
* 9.47.84 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean	547
* 9.47.85 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean	547
* 9.47.86 sizeToFitWidthOfColumn(column as Integer) as Double	547
* 9.47.87 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)	548
* 9.47.88 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean	548
* 9.47.89 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean	548
* 9.47.90 tooltipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTa- bleColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String	548
* 9.47.91 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String	549
* 9.47.92 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)	549
* 9.47.93 validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer	550
* 9.47.94 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS	550
* 9.47.95 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)	551
* 9.47.96 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)	551
* 9.47.98 willTile	552
* 9.47.99 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean	552

	97
• 6 AVFoundation	169
– 9.47.1 control NSOutlineControlMBS	524
* 9.47.27 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	530
* 9.47.97 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	551

• 9 Cocoa Controls	297
– 9.47.1 control NSOutlineControlMBS	524
* 9.47.3 AcceptTabs as Boolean	524
* 9.47.4 allowsColumnReordering as Boolean	524
* 9.47.5 allowsColumnResizing as Boolean	525
* 9.47.6 allowsColumnSelection as Boolean	525
* 9.47.7 allowsEmptySelection as Boolean	525
* 9.47.8 allowsMultipleSelection as Boolean	525
* 9.47.9 autohidesScrollers as Boolean	526
* 9.47.10 hasHorizontalScroller as Boolean	526
* 9.47.11 hasVerticalScroller as Boolean	526
* 9.47.12 ScrollView as NSScrollViewMBS	526
* 9.47.13 View as NSOutlineViewMBS	526
* 9.47.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean	527
* 9.47.16 BoundsChanged	527
* 9.47.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	527
* 9.47.18 Close	528
* 9.47.19 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)	528
* 9.47.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	528
* 9.47.21 concludeDragOperation(info as NSDraggingInfoMBS)	528
* 9.47.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	529
* 9.47.23 ContextualMenuAction(hitItem as MenuItem) as Boolean	529
* 9.47.24 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS	529
* 9.47.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	530
* 9.47.26 didClickTableColumn(tableColumn as NSTableColumnMBS)	530
* 9.47.28 didDragTableColumn(tableColumn as NSTableColumnMBS)	530
* 9.47.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	531
* 9.47.30 didTile	531
* 9.47.31 DoubleClick	531
* 9.47.32 draggingEnded(info as NSDraggingInfoMBS)	531
* 9.47.33 draggingExited(info as NSDraggingInfoMBS)	531
* 9.47.34 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	532
* 9.47.35 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)	532
* 9.47.36 EnableMenuItems	533

* 9.47.37	FrameChanged	533
* 9.47.38	GotFocus	533
* 9.47.39	heightOfRowByItem(item as NSOutlineViewItemMBS) as Double	533
* 9.47.40	isGroupItem(item as NSOutlineViewItemMBS) as Boolean	534
* 9.47.41	isItemExpandable(item as NSOutlineViewItemMBS) as Boolean	534
* 9.47.42	ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	535
* 9.47.43	ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	535
* 9.47.44	itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS	535
* 9.47.45	ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	536
* 9.47.46	ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)	536
* 9.47.47	MouseDown(e as NSEventMBS) as Boolean	536
* 9.47.48	MouseDownDragged(e as NSEventMBS) as Boolean	536
* 9.47.49	MouseDownUp(e as NSEventMBS) as Boolean	536
* 9.47.50	LostFocus	537
* 9.47.51	MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean	537
* 9.47.52	MouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	537
* 9.47.53	MouseDownDrag(x as Integer, y as Integer)	537
* 9.47.54	MouseDownUp(x as Integer, y as Integer)	538
* 9.47.55	namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()	538
* 9.47.56	nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS	538
* 9.47.57	numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	539
* 9.47.58	objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant	539
* 9.47.59	Open	539
* 9.47.60	OtherMouseDown(e as NSEventMBS) as Boolean	539
* 9.47.61	OtherMouseDownDragged(e as NSEventMBS) as Boolean	540
* 9.47.62	OtherMouseDownUp(e as NSEventMBS) as Boolean	540
* 9.47.63	pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS	540
* 9.47.64	persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant	540
* 9.47.65	RightMouseDown(e as NSEventMBS) as Boolean	541
* 9.47.66	RightMouseDownDragged(e as NSEventMBS) as Boolean	541
* 9.47.67	RightMouseDownUp(e as NSEventMBS) as Boolean	541
* 9.47.68	rowViewForItem(item as NSOutlineViewItemMBS) as NSTableRowViewMBS	541
* 9.47.69	ScaleFactorChanged(NewFactor as Double)	542
* 9.47.70	SelectionDidChange(notification as NSNotificationMBS)	542

- * 9.47.71 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS 542
- * 9.47.72 SelectionIsChanging(notification as NSNotificationMBS) 543
- * 9.47.73 selectionShouldChangeInOutlineView as Boolean 543
- * 9.47.74 setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant) 543
- * 9.47.75 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean 543
- * 9.47.76 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean 544
- * 9.47.77 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 544
- * 9.47.78 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean 545
- * 9.47.79 shouldReorderColumn(columnIndex as Integer, newColumnIndex as Integer) as Boolean 545
- * 9.47.80 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean 545
- * 9.47.81 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean 546
- * 9.47.82 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 546
- * 9.47.83 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean 546
- * 9.47.84 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean 547
- * 9.47.85 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean 547
- * 9.47.86 sizeToFitWidthOfColumn(column as Integer) as Double 547
- * 9.47.87 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS) 548
- * 9.47.88 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 548
- * 9.47.89 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean 548
- * 9.47.90 toolTipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String 548
- * 9.47.91 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String 549
- * 9.47.92 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS) 549
- * 9.47.93 validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer 550
- * 9.47.94 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS 550
- * 9.47.95 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 551
- * 9.47.96 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) 551
- * 9.47.98 willTile 552

* 9.47.99 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean	552
– 9.48.1 class NSOutlineViewItemMBS	553
* 9.48.3 Constructor	553
* 9.48.4 sortedArrayUsingDescriptor(values() as NSOutlineViewItemMBS, sortDescriptor as NSSortDescriptorMBS) as NSOutlineViewItemMBS()	553
* 9.48.5 sortedArrayUsingDescriptors(values() as NSOutlineViewItemMBS, sortDescriptor() as NSSortDescriptorMBS) as NSOutlineViewItemMBS()	553
* 9.48.7 Description as String	554
* 9.48.8 Handle as Integer	554
* 9.48.9 valueForKey(key as String) as Variant	554
* 9.48.11 Description as String	554
* 9.48.12 setValueForKey(key as string, value as Variant)	555
* 9.48.13 setValueForUndefinedKey(key as string, value as Variant)	555
* 9.48.14 valueForKey(key as string) as Variant	555
* 9.48.15 valueForKeyForUndefinedKey(key as string) as Variant	556
– 9.49.1 class NSOutlineViewMBS	557
* 9.49.3 child(index as Integer, offItem as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	557
* 9.49.4 childIndexForItem(item as NSOutlineViewItemMBS) as Integer	558
* 9.49.5 collapseItem(item as NSOutlineViewItemMBS)	558
* 9.49.6 collapseItem(item as NSOutlineViewItemMBS, collapseChildren as Boolean)	558
* 9.49.7 Constructor	559
* 9.49.8 Constructor(Handle as Integer)	559
* 9.49.9 Constructor(left as Double, top as Double, width as Double, height as Double)	559
* 9.49.10 expandItem(item as NSOutlineViewItemMBS)	560
* 9.49.11 expandItem(item as NSOutlineViewItemMBS, expandChildren as Boolean)	560
* 9.49.12 frameOfOutlineCellAtRow(row as Integer) as NSRectMBS	560
* 9.49.13 insertItemsAtIndexes(indexes as NSIndexSetMBS, Parent as NSOutlineViewItemMBS, animationOptions as Integer)	561
* 9.49.14 insertRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)	561
* 9.49.15 isExpandable(item as NSOutlineViewItemMBS) as Boolean	561
* 9.49.16 isItemExpanded(item as NSOutlineViewItemMBS) as Boolean	562
* 9.49.17 itemAtRow(row as Integer) as NSOutlineViewItemMBS	562
* 9.49.18 levelForItem(item as NSOutlineViewItemMBS) as Integer	562
* 9.49.19 levelForRow(row as Integer) as Integer	562
* 9.49.20 moveItemAtIndex(oldIndex as Integer, oldParent as NSOutlineViewItemMBS, newIndex as Integer, newParent as NSOutlineViewItemMBS)	562
* 9.49.21 moveRowAtIndex(oldIndex as Integer, newIndex as Integer)	563
* 9.49.22 NSOutlineViewColumnDidMoveNotification as String	563
* 9.49.23 NSOutlineViewColumnDidResizeNotification as String	563
* 9.49.24 NSOutlineViewDisclosureButtonKey as String	564

* 9.49.25 NSOutlineViewItemDidCollapseNotification as String	564
* 9.49.26 NSOutlineViewItemDidExpandNotification as String	564
* 9.49.27 NSOutlineViewItemWillCollapseNotification as String	564
* 9.49.28 NSOutlineViewItemWillExpandNotification as String	565
* 9.49.29 NSOutlineViewSelectionDidChangeNotification as String	565
* 9.49.30 NSOutlineViewSelectionIsChangingNotification as String	565
* 9.49.31 NSOutlineViewShowHideButtonKey as String	565
* 9.49.32 numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer	566
* 9.49.33 parentForItem(item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS	566
* 9.49.34 reloadItem(item as NSOutlineViewItemMBS)	566
* 9.49.35 reloadItem(item as NSOutlineViewItemMBS, reloadChildren as Boolean)	566
* 9.49.36 removeItemsAtIndexes(indexes as NSIndexSetMBS, Parent as NSOutlineViewItemMBS, animationOptions as Integer)	567
* 9.49.37 removeRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)	567
* 9.49.38 rowForItem(item as NSOutlineViewItemMBS) as Integer	567
* 9.49.39 setDropItem(item as NSOutlineViewItemMBS, dropChildIndex as Integer)	568
* 9.49.41 autosizesOutlineColumn as Boolean	568
* 9.49.42 autosaveExpandedItems as Boolean	568
* 9.49.43 indentationMarkerFollowsCell as Boolean	569
* 9.49.44 indentationPerLevel as Double	569
* 9.49.45 outlinetableColumn as NSTableColumnMBS	569
* 9.49.46 stronglyReferencesItems as Boolean	569
* 9.49.47 userInterfaceLayoutDirection as Integer	570

	103
• 7 Cocoa	171
– 7.2.1 class NSPopoverMBS	176
* 7.2.3 available as boolean	176
* 7.2.4 Close	177
* 7.2.5 Constructor	177
* 7.2.6 Destructor	177
* 7.2.7 isShown as boolean	177
* 7.2.8 NSPopoverCloseReasonDetachToWindow as string	177
* 7.2.9 NSPopoverCloseReasonKey as string	178
* 7.2.10 NSPopoverCloseReasonStandard as string	178
* 7.2.11 NSPopoverDidCloseNotification as string	178
* 7.2.12 NSPopoverDidShowNotification as string	178
* 7.2.13 NSPopoverWillCloseNotification as string	178
* 7.2.14 NSPopoverWillShowNotification as string	179
* 7.2.15 performClose	179
* 7.2.16 showRelativeToRect(positioningRect as NSRectMBS, view as NSViewMBS, edge as Integer)	179
* 7.2.18 Handle as Integer	180
* 7.2.19 Tag as Variant	180
* 7.2.20 animates as boolean	180
* 7.2.21 behavior as Integer	180
* 7.2.22 contentSize as NSSizeMBS	180
* 7.2.23 contentViewController as NSViewControllerMBS	181
* 7.2.24 positioningRect as NSRectMBS	181
* 7.2.26 detachableWindowForPopover as NSWindowMBS	181
* 7.2.27 popoverDidClose(notification as NSNotificationMBS)	182
* 7.2.28 popoverDidDetach	182
* 7.2.29 popoverDidShow(notification as NSNotificationMBS)	182
* 7.2.30 popoverShouldClose as boolean	182
* 7.2.31 popoverShouldDetach as boolean	183
* 7.2.32 popoverWillClose(notification as NSNotificationMBS)	183
* 7.2.33 popoverWillShow(notification as NSNotificationMBS)	183

• 9 Cocoa Controls	297
– 9.50.1 control NSPopUpButtonControlMBS	571
* 9.50.3 View as NSPopUpButtonMBS	571
* 9.50.5 Action	571
* 9.50.6 BoundsChanged	571
* 9.50.7 Close	572
* 9.50.8 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	572
* 9.50.9 ContextualMenuItemAction(hitItem as MenuItem) as Boolean	572
* 9.50.10 didCloseContextualMenu(menu as NSMenuItemMBS, NSEvent as NSEventMBS)	572
* 9.50.11 EnableMenuItems	572
* 9.50.12 FrameChanged	573
* 9.50.13 GotFocus	573
* 9.50.14 LostFocus	573
* 9.50.15MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	573
* 9.50.16 MouseDrag(x as Integer, y as Integer)	574
* 9.50.17 MouseUp(x as Integer, y as Integer)	574
* 9.50.18 Open	574
* 9.50.19 ScaleFactorChanged(NewFactor as Double)	574
* 9.50.20 willShowContextualMenu(menu as NSMenuItemMBS, NSEvent as NSEventMBS)	574
– 9.51.1 class NSPopUpButtonMBS	576
* 9.51.3 addItemWithTitle(titles() as string)	576
* 9.51.4 addItemWithTitle(title as string)	577
* 9.51.5 Constructor	577
* 9.51.6 Constructor(Handle as Integer)	578
* 9.51.7 Constructor(left as Double, top as Double, width as Double, height as Double)	578
* 9.51.8 Constructor(left as Double, top as Double, width as Double, height as Double, pull-Down as boolean)	579
* 9.51.9 indexOfItem(item as NSMenuItemMBS) as Integer	579
* 9.51.10 indexOfItemWithTag(tag as Integer) as Integer	579
* 9.51.11 indexOfItemWithTitle(title as string) as Integer	580
* 9.51.12 indexOfSelectedItem as Integer	580
* 9.51.13 insertItemWithTitle(title as string, atIndex as Integer)	580
* 9.51.14 itemAtIndex(index as Integer) as NSMenuItemMBS	580
* 9.51.15 itemWithTitle(title as string) as NSMenuItemMBS	581
* 9.51.16 lastItem as NSMenuItemMBS	581
* 9.51.17 removeAllItems	581
* 9.51.18 removeItemAtIndex(index as Integer)	581
* 9.51.19 removeItemWithTitle(title as string)	582
* 9.51.20 selectItem(item as NSMenuItemMBS)	582
* 9.51.21 selectItemAtIndex(index as Integer)	582

	105
* 9.51.22 selectItemWithTag(tag as Integer) as boolean	582
* 9.51.23 selectItemWithTitle(title as string)	583
* 9.51.24 setTitle(title as string)	583
* 9.51.25 synchronizeTitleAndSelectedItem	583
* 9.51.26 titleOfSelectedItem as string	583
* 9.51.28 arrowPosition as Integer	583
* 9.51.29 autoenablesItems as boolean	584
* 9.51.30 menu as NSMenuItemMBS	584
* 9.51.31 numberOfItems as Integer	584
* 9.51.32 pullsDown as boolean	584
* 9.51.33 selectedItem as NSMenuItemMBS	585
– 9.52.1 class NSProgressIndicatorMBS	586
* 9.52.3 Constructor	586
* 9.52.4 Constructor(Handle as Integer)	587
* 9.52.5 Constructor(left as Double, top as Double, width as Double, height as Double)	587
* 9.52.6 incrementBy(delta as Double)	587
* 9.52.7 sizeToFit	588
* 9.52.8 startAnimation	588
* 9.52.9 stopAnimation	588
* 9.52.11 controlSize as Integer	588
* 9.52.12 controlTint as Integer	588
* 9.52.13 doubleValue as Double	589
* 9.52.14 isBezeled as boolean	589
* 9.52.15 isDisplayedWhenStopped as boolean	589
* 9.52.16 isIndeterminate as boolean	589
* 9.52.17 maxValue as Double	589
* 9.52.18 minValue as Double	590
* 9.52.19 style as Integer	590
* 9.52.20 usesThreadedAnimation as boolean	590
– 9.53.1 class NSScrollerMBS	592
* 9.53.3 checkSpaceForParts	592
* 9.53.4 Constructor	593
* 9.53.5 Constructor(Handle as Integer)	593
* 9.53.6 Constructor(left as Double, top as Double, width as Double, height as Double)	593
* 9.53.7 drawArrow(Arrow as Integer, highlight as boolean)	594
* 9.53.8 drawKnob	594
* 9.53.9 drawKnobSlotInRect(slotRect as NSRectMBS, highlight as boolean)	594
* 9.53.10 drawParts	594
* 9.53.11 highlight(flag as boolean)	595
* 9.53.12 hitPart as Integer	595
* 9.53.13 isCompatibleWithOverlayScrollers as boolean	595

* 9.53.14	NSPreferredScrollerStyleDidChangeNotification as string	595
* 9.53.15	preferredScrollerStyle as Integer	595
* 9.53.16	rectForPart(part as Integer) as NSRectMBS	596
* 9.53.17	scrollerWidth as Double	596
* 9.53.18	scrollerWidthForControlSize(controlSize as Integer) as Double	597
* 9.53.19	setFloatValue(aFloat as Double, proportion as Double)	597
* 9.53.20	testPart(p as NSPointMBS) as Integer	597
* 9.53.21	trackKnob(theEvent as NSEventMBS)	597
* 9.53.22	trackScrollButtons(theEvent as NSEventMBS)	597
* 9.53.23	usableParts as Integer	598
* 9.53.25	arrowsPosition as Integer	598
* 9.53.26	controlSize as Integer	598
* 9.53.27	controlTint as Integer	598
* 9.53.28	knobProportion as Double	599
* 9.53.29	knobStyle as Integer	599
* 9.53.30	scrollerStyle as Integer	599
– 9.54.1	class NSScrollViewMBS	602
* 9.54.3	Constructor	602
* 9.54.4	Constructor(Handle as Integer)	603
* 9.54.5	Constructor(left as Double, top as Double, width as Double, height as Double)	603
* 9.54.6	flashScrollers	604
* 9.54.7	reflectScrolledClipView(clipView as NSClipViewMBS)	604
* 9.54.8	tile	604
* 9.54.10	autohidesScrollers as boolean	605
* 9.54.11	backgroundColor as NSColorMBS	605
* 9.54.12	borderType as Integer	605
* 9.54.13	contentSize as NSSizeMBS	605
* 9.54.14	contentView as NSClipViewMBS	605
* 9.54.15	documentCursor as Variant	606
* 9.54.16	documentView as NSViewMBS	606
* 9.54.17	documentVisibleRect as NSRectMBS	606
* 9.54.18	drawsBackground as boolean	606
* 9.54.19	FindBarPosition as Integer	606
* 9.54.20	FindBarView as NSViewMBS	607
* 9.54.21	FindBarVisible as Boolean	607
* 9.54.22	hasHorizontalRuler as boolean	607
* 9.54.23	hasHorizontalScroller as boolean	607
* 9.54.24	hasVerticalRuler as boolean	607
* 9.54.25	hasVerticalScroller as boolean	608
* 9.54.26	horizontalLineScroll as Double	608
* 9.54.27	horizontalPageScroll as Double	608

	107
* 9.54.28 horizontalScrollElasticity as Integer	608
* 9.54.29 horizontalScroller as NSScrollerMBS	609
* 9.54.30 lineScroll as Double	609
* 9.54.31 pageScroll as Double	609
* 9.54.32 rulersVisible as boolean	610
* 9.54.33 scrollerKnobStyle as Integer	610
* 9.54.34 scrollerStyle as Integer	610
* 9.54.35 scrollsDynamically as boolean	611
* 9.54.36 usesPredominantAxisScrolling as boolean	611
* 9.54.37 verticalLineScroll as Double	611
* 9.54.38 verticalPageScroll as Double	612
* 9.54.39 verticalScrollElasticity as Integer	612
* 9.54.40 verticalScroller as NSScrollerMBS	612
– 9.55.1 class NSSliderMBS	614
* 9.55.3 acceptsFirstMouse(event as NSEventMBS) as boolean	614
* 9.55.4 closestTickMarkValueToValue(value as Double) as Double	614
* 9.55.5 Constructor	615
* 9.55.6 Constructor(Handle as Integer)	615
* 9.55.7 Constructor(left as Double, top as Double, width as Double, height as Double)	615
* 9.55.8 indexOfTickMarkAtPoint(p as NSPointMBS) as Integer	616
* 9.55.9 indexOfTickMarkAtPoint(x as Double, y as Double) as Integer	616
* 9.55.10 rectOfTickMarkAtIndex(index as Integer) as NSRectMBS	616
* 9.55.11 tickMarkValueAtIndex(index as Integer) as Double	616
* 9.55.13 allowsTickMarkValuesOnly as boolean	617
* 9.55.14 altIncrementValue as Double	617
* 9.55.15 image as NSImageMBS	617
* 9.55.16 isVertical as Integer	617
* 9.55.17 knobThickness as Double	618
* 9.55.18 maxValue as Double	618
* 9.55.19 minValue as Double	618
* 9.55.20 numberOfTickMarks as Integer	618
* 9.55.21 sliderType as Integer	619
* 9.55.22 tickMarkPosition as Integer	619
* 9.55.23 title as string	619
* 9.55.24 titleCell as NSCellMBS	619
* 9.55.25 titleColor as NSColorMBS	620
* 9.55.26 titleFont as NSFontMBS	620
* 9.55.27 trackFillColor as NSColorMBS	620

- **17 Statusitem** 985
 - 17.1.1 class `NSStatusBarButtonMBS` 985
 - * 17.1.3 Available as boolean 985
 - * 17.1.5 `appearsDisabled` as Boolean 986

	109
• 9 Cocoa Controls	297
– 9.56.1 class NSStepperMBS	621
* 9.56.3 Constructor	621
* 9.56.4 Constructor(Handle as Integer)	621
* 9.56.5 Constructor(left as Double, top as Double, width as Double, height as Double)	622
* 9.56.7 autorepeat as boolean	622
* 9.56.8 increment as Double	622
* 9.56.9 maxValue as Double	623
* 9.56.10 minValue as Double	623
* 9.56.11 valueWraps as boolean	623
– 9.57.1 control NSSwitchControlMBS	624
* 9.57.3 State as Boolean	624
* 9.57.4 View as NSSwitchMBS	624
* 9.57.6 Action	625
* 9.57.7 BoundsChanged	625
* 9.57.8 Close	625
* 9.57.9 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	625
* 9.57.10 ContextualMenuAction(hitItem as MenuItem) as Boolean	625
* 9.57.11 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	626
* 9.57.12 EnableMenuItems	626
* 9.57.13 FrameChanged	626
* 9.57.14 GotFocus	626
* 9.57.15 LostFocus	626
* 9.57.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	627
* 9.57.17 MouseDrag(x as Integer, y as Integer)	627
* 9.57.18 MouseUp(x As Integer, y As Integer)	627
* 9.57.19 Open	627
* 9.57.20 ScaleFactorChanged(NewFactor as double)	628
* 9.57.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	628
– 9.58.1 class NSSwitchMBS	629
* 9.58.3 Constructor	629
* 9.58.4 Constructor(Handle as Integer)	630
* 9.58.5 Constructor(left as double, top as double, width as double, height as double)	630
* 9.58.7 state as Boolean	631
– 9.59.1 class NSTableColumnMBS	632
* 9.59.3 Constructor(identifier as string)	632
* 9.59.4 dataCellForRow(row as Integer) as NSCellMBS	632
* 9.59.5 sizeToFit	633
* 9.59.7 dataCell as NSCellMBS	633

* 9.59.8 Editable as boolean	633
* 9.59.9 headerCell as NSTableHeaderCellMBS	633
* 9.59.10 headerToolTip as string	633
* 9.59.11 Hidden as boolean	634
* 9.59.12 identifier as string	634
* 9.59.13 maxWidth as Double	634
* 9.59.14 minWidth as Double	634
* 9.59.15 Resizable as boolean	634
* 9.59.16 resizingMask as Integer	635
* 9.59.17 sortDescriptorPrototype as NSSortDescriptorMBS	635
* 9.59.18 tableView as NSTableViewMBS	635
* 9.59.19 title as String	635
* 9.59.20 width as Double	635
– 9.60.1 control NSTableControlMBS	637
* 9.60.3 AcceptTabs as Boolean	637
* 9.60.4 allowsColumnReordering as Boolean	638
* 9.60.5 allowsColumnResizing as Boolean	638
* 9.60.6 allowsColumnSelection as Boolean	638
* 9.60.7 allowsEmptySelection as Boolean	638
* 9.60.8 allowsMultipleSelection as Boolean	639
* 9.60.9 autohidesScrollers as Boolean	639
* 9.60.10 disableCellEvents as Boolean	639
* 9.60.11 disableViewEvents as Boolean	639
* 9.60.12 hasHorizontalScroller as Boolean	640
* 9.60.13 hasVerticalScroller as Boolean	640
* 9.60.14 ScrollView as NSScrollViewMBS	640
* 9.60.15 View as NSTableViewMBS	640
* 9.60.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean	640
* 9.60.18 BoundsChanged	641
* 9.60.19 Close	641
* 9.60.20 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	641
* 9.60.21 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	641
* 9.60.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	642
* 9.60.23 ContextualMenuAction(hitItem as MenuItem) as Boolean	642
* 9.60.24 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	642
* 9.60.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	642
* 9.60.26 didClickTableColumn(tableColumn as NSTableColumnMBS)	643
* 9.60.28 didDragTableColumn(tableColumn as NSTableColumnMBS)	643

	111
* 9.60.29 didRemoveRowView(rowView as NSTableViewMBS, row as Integer)	643
* 9.60.30 didTile	644
* 9.60.31 DoubleClick	644
* 9.60.32 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	644
* 9.60.33 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)	644
* 9.60.34 EnableMenuItems	645
* 9.60.35 FrameChanged	645
* 9.60.36 GotFocus	645
* 9.60.37 heightForRow(row as Int64) as Double	645
* 9.60.38 isGroupRow(row as Int64) as boolean	646
* 9.60.39 LeftMouseDown(e as NSEventMBS) as boolean	646
* 9.60.40 LeftMouseDragged(e as NSEventMBS) as boolean	646
* 9.60.41 LeftMouseUp(e as NSEventMBS) as boolean	647
* 9.60.42 LostFocus	647
* 9.60.43 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	647
* 9.60.44 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	648
* 9.60.45 MouseDrag(x as Integer, y as Integer)	648
* 9.60.46 MouseUp(x as Integer, y as Integer)	648
* 9.60.47 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()	648
* 9.60.48 nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64	649
* 9.60.49 numberOfRowsInTableView as Integer	649
* 9.60.50 objectValue(column as NSTableColumnMBS, row as Integer) as Variant	649
* 9.60.51 Open	649
* 9.60.52 OtherMouseDown(e as NSEventMBS) as boolean	650
* 9.60.53 OtherMouseDragged(e as NSEventMBS) as boolean	650
* 9.60.54 OtherMouseUp(e as NSEventMBS) as boolean	650
* 9.60.55 pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS	650
* 9.60.56 RightMouseDown(e as NSEventMBS) as boolean	651
* 9.60.57 RightMouseDragged(e as NSEventMBS) as boolean	651
* 9.60.58 RightMouseUp(e as NSEventMBS) as boolean	651
* 9.60.59 rowActionsForRow(row as Integer, edge as Integer) as NSTableViewRowActionMBS()	651
* 9.60.60 rowViewForRow(row as Integer) as NSTableViewRowMBS	652
* 9.60.61 scaleFactorChanged(NewFactor as Double)	652
* 9.60.62 SelectionDidChange(notification as NSNotificationMBS)	652
* 9.60.63 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	652
* 9.60.64 SelectionIsChanging(notification as NSNotificationMBS)	653
* 9.60.65 selectionShouldChangeInTableView as boolean	653

- * 9.60.66 setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer) as boolean 653
- * 9.60.67 shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean 654
- * 9.60.68 shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64) as boolean 654
- * 9.60.69 shouldSelectRow(row as Int64) as boolean 654
- * 9.60.70 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean 655
- * 9.60.71 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean 655
- * 9.60.72 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean 655
- * 9.60.73 shouldTypeSelectForEvent(e as NSEventMBS, searchString as string) as Boolean 656
- * 9.60.74 sizeToFitWidthOfColumn(column as Int64) as Double 656
- * 9.60.75 sortDescriptorsDidChange(oldDescriptors()) as NSSortDescriptorMBS 657
- * 9.60.76 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean 657
- * 9.60.77 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean 657
- * 9.60.78 toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string 657
- * 9.60.79 typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string 658
- * 9.60.80 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS) 658
- * 9.60.81 validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer) as Integer 659
- * 9.60.82 view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS 659
- * 9.60.83 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) 659
- * 9.60.85 willTile 660
- * 9.60.86 writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS) as boolean 660

	113
• 6 AVFoundation	169
– 9.60.1 control NSTableControlMBS	637
* 9.60.27 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	643
* 9.60.84 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	660

• 9 Cocoa Controls	297
– 9.60.1 control NSTableControlMBS	637
* 9.60.3 AcceptTabs as Boolean	637
* 9.60.4 allowsColumnReordering as Boolean	638
* 9.60.5 allowsColumnResizing as Boolean	638
* 9.60.6 allowsColumnSelection as Boolean	638
* 9.60.7 allowsEmptySelection as Boolean	638
* 9.60.8 allowsMultipleSelection as Boolean	639
* 9.60.9 autohidesScrollers as Boolean	639
* 9.60.10 disableCellEvents as Boolean	639
* 9.60.11 disableViewEvents as Boolean	639
* 9.60.12 hasHorizontalScroller as Boolean	640
* 9.60.13 hasVerticalScroller as Boolean	640
* 9.60.14 ScrollView as NSScrollViewMBS	640
* 9.60.15 View as NSTableViewMBS	640
* 9.60.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean	640
* 9.60.18 BoundsChanged	641
* 9.60.19 Close	641
* 9.60.20 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	641
* 9.60.21 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	641
* 9.60.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	642
* 9.60.23 ContextualMenuAction(hitItem as MenuItem) as Boolean	642
* 9.60.24 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	642
* 9.60.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	642
* 9.60.26 didClickTableColumn(tableColumn as NSTableColumnMBS)	643
* 9.60.28 didDragTableColumn(tableColumn as NSTableColumnMBS)	643
* 9.60.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	643
* 9.60.30 didTile	644
* 9.60.31 DoubleClick	644
* 9.60.32 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	644
* 9.60.33 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)	644
* 9.60.34 EnableMenuItems	645
* 9.60.35 FrameChanged	645
* 9.60.36 GotFocus	645
* 9.60.37 heightOfRow(row as Int64) as Double	645
* 9.60.38 isGroupRow(row as Int64) as boolean	646

	115
* 9.60.39 LeftMouseDown(e as NSEventMBS) as boolean	646
* 9.60.40 LeftMouseDownDragged(e as NSEventMBS) as boolean	646
* 9.60.41 LeftMouseDownUp(e as NSEventMBS) as boolean	647
* 9.60.42 LostFocus	647
* 9.60.43 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	647
* 9.60.44 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	648
* 9.60.45 MouseDrag(x as Integer, y as Integer)	648
* 9.60.46 MouseUp(x as Integer, y as Integer)	648
* 9.60.47 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()	648
* 9.60.48 nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64	649
* 9.60.49 numberOfRowsInTableView as Integer	649
* 9.60.50 objectValue(column as NSTableColumnMBS, row as Integer) as Variant	649
* 9.60.51 Open	649
* 9.60.52 OtherMouseDown(e as NSEventMBS) as boolean	650
* 9.60.53 OtherMouseDownDragged(e as NSEventMBS) as boolean	650
* 9.60.54 OtherMouseDownUp(e as NSEventMBS) as boolean	650
* 9.60.55 pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS	650
* 9.60.56 RightMouseDown(e as NSEventMBS) as boolean	651
* 9.60.57 RightMouseDownDragged(e as NSEventMBS) as boolean	651
* 9.60.58 RightMouseDownUp(e as NSEventMBS) as boolean	651
* 9.60.59 rowActionsForRow(row as Integer, edge as Integer) as NSTableViewRowActionMBS()	651
* 9.60.60 rowViewForRow(row as Integer) as NSTableRowViewMBS	652
* 9.60.61 scaleFactorChanged(NewFactor as Double)	652
* 9.60.62 SelectionDidChange(notification as NSNotificationMBS)	652
* 9.60.63 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	652
* 9.60.64 SelectionIsChanging(notification as NSNotificationMBS)	653
* 9.60.65 selectionShouldChangeInTableView as boolean	653
* 9.60.66 setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)	653
* 9.60.67 shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean	654
* 9.60.68 shouldReorderColumn(columnIndex as Int64, newIndex as Int64) as boolean	654
* 9.60.69 shouldSelectRow(row as Int64) as boolean	654
* 9.60.70 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean	655
* 9.60.71 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean	655
* 9.60.72 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean	655

- * 9.60.73 `shouldTypeSelectForEvent(e as NSEventMBS, searchString as string)` as Boolean 656
- * 9.60.74 `sizeToFitWidthOfColumn(column as Int64)` as Double 656
- * 9.60.75 `sortDescriptorsDidChange(oldDescriptors())` as `NSSortDescriptorMBS` 657
- * 9.60.76 `textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS)` as boolean 657
- * 9.60.77 `textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS)` as boolean 657
- * 9.60.78 `toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS)` as string 657
- * 9.60.79 `typeSelectString(tableColumn as NSTableColumnMBS, row as Int64)` as string 658
- * 9.60.80 `updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)` 658
- * 9.60.81 `validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer)` as Integer 659
- * 9.60.82 `view(tableColumn as NSTableColumnMBS, row as Integer)` as `NSViewMBS` 659
- * 9.60.83 `willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)` 659
- * 9.60.85 `willTile` 660
- * 9.60.86 `writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS)` as boolean 660

	117
• 6 AVFoundation	169
– 9.60.1 control NSTableControlMBS	637
* 9.60.27 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	643
* 9.60.84 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	660

• 9 Cocoa Controls	297
– 9.60.1 control NSTableControlMBS	637
* 9.60.3 AcceptTabs as Boolean	637
* 9.60.4 allowsColumnReordering as Boolean	638
* 9.60.5 allowsColumnResizing as Boolean	638
* 9.60.6 allowsColumnSelection as Boolean	638
* 9.60.7 allowsEmptySelection as Boolean	638
* 9.60.8 allowsMultipleSelection as Boolean	639
* 9.60.9 autohidesScrollers as Boolean	639
* 9.60.10 disableCellEvents as Boolean	639
* 9.60.11 disableViewEvents as Boolean	639
* 9.60.12 hasHorizontalScroller as Boolean	640
* 9.60.13 hasVerticalScroller as Boolean	640
* 9.60.14 ScrollView as NSScrollViewMBS	640
* 9.60.15 View as NSTableViewMBS	640
* 9.60.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean	640
* 9.60.18 BoundsChanged	641
* 9.60.19 Close	641
* 9.60.20 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	641
* 9.60.21 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)	641
* 9.60.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	642
* 9.60.23 ContextualMenuAction(hitItem as MenuItem) as Boolean	642
* 9.60.24 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	642
* 9.60.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	642
* 9.60.26 didClickTableColumn(tableColumn as NSTableColumnMBS)	643
* 9.60.28 didDragTableColumn(tableColumn as NSTableColumnMBS)	643
* 9.60.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	643
* 9.60.30 didTile	644
* 9.60.31 DoubleClick	644
* 9.60.32 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	644
* 9.60.33 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)	644
* 9.60.34 EnableMenuItems	645
* 9.60.35 FrameChanged	645
* 9.60.36 GotFocus	645
* 9.60.37 heightOfRow(row as Int64) as Double	645
* 9.60.38 isGroupRow(row as Int64) as boolean	646

	119
* 9.60.39 LeftMouseDown(e as NSEventMBS) as boolean	646
* 9.60.40 LeftMouseDragged(e as NSEventMBS) as boolean	646
* 9.60.41 LeftMouseUp(e as NSEventMBS) as boolean	647
* 9.60.42 LostFocus	647
* 9.60.43 MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	647
* 9.60.44 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	648
* 9.60.45 MouseDrag(x as Integer, y as Integer)	648
* 9.60.46 MouseUp(x as Integer, y as Integer)	648
* 9.60.47 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()	648
* 9.60.48 nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64	649
* 9.60.49 numberOfRowsInTableView as Integer	649
* 9.60.50 objectValue(column as NSTableColumnMBS, row as Integer) as Variant	649
* 9.60.51 Open	649
* 9.60.52 OtherMouseDown(e as NSEventMBS) as boolean	650
* 9.60.53 OtherMouseDragged(e as NSEventMBS) as boolean	650
* 9.60.54 OtherMouseUp(e as NSEventMBS) as boolean	650
* 9.60.55 pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS	650
* 9.60.56 RightMouseDown(e as NSEventMBS) as boolean	651
* 9.60.57 RightMouseDragged(e as NSEventMBS) as boolean	651
* 9.60.58 RightMouseUp(e as NSEventMBS) as boolean	651
* 9.60.59 rowActionsForRow(row as Integer, edge as Integer) as NSTableViewRowActionMBS()	651
* 9.60.60 rowViewForRow(row as Integer) as NSTableRowViewMBS	652
* 9.60.61 scaleFactorChanged(NewFactor as Double)	652
* 9.60.62 SelectionDidChange(notification as NSNotificationMBS)	652
* 9.60.63 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	652
* 9.60.64 SelectionIsChanging(notification as NSNotificationMBS)	653
* 9.60.65 selectionShouldChangeInTableView as boolean	653
* 9.60.66 setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)	653
* 9.60.67 shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean	654
* 9.60.68 shouldReorderColumn(columnIndex as Int64, newIndex as Int64) as boolean	654
* 9.60.69 shouldSelectRow(row as Int64) as boolean	654
* 9.60.70 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean	655
* 9.60.71 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean	655
* 9.60.72 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean	655

* 9.60.73	shouldTypeSelectForEvent(e as NSEventMBS, searchString as string) as Boolean	656
* 9.60.74	sizeToFitWidthOfColumn(column as Int64) as Double	656
* 9.60.75	sortDescriptorsDidChange(oldDescriptors()) as NSSortDescriptorMBS)	657
* 9.60.76	textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean	657
* 9.60.77	textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean	657
* 9.60.78	toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string	657
* 9.60.79	typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string	658
* 9.60.80	updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)	658
* 9.60.81	validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer) as Integer	659
* 9.60.82	view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS	659
* 9.60.83	willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)	659
* 9.60.85	willTile	660
* 9.60.86	writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS) as boolean	660
– 9.61.1	class NSTableDataSourceMBS	662
* 9.61.3	Close	662
* 9.61.4	numberOfRowsInTableView as Integer	662
* 9.61.5	objectValue(column as NSTableColumnMBS, row as Integer) as Variant	662
* 9.61.6	setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)	663
* 9.61.7	sortDescriptorsDidChange(oldDescriptors()) as NSSortDescriptorMBS)	663
– 9.62.1	class NSTableHeaderCellMBS	664
* 9.62.3	drawSortIndicatorWithFrame(cellFrame as NSRectMBS, inView as NSViewMBS, ascending as boolean, priority as Integer)	664
* 9.62.4	sortIndicatorRectForBounds(r as NSRectMBS) as NSRectMBS	664
– 9.63.1	class NSTableHeaderViewMBS	665
* 9.63.3	columnAtPoint(point as NSPointMBS) as Integer	665
* 9.63.4	Constructor	665
* 9.63.5	Constructor(Handle as Integer)	666
* 9.63.6	Constructor(left as Double, top as Double, width as Double, height as Double)	666
* 9.63.7	draggedColumn as Integer	666
* 9.63.8	draggedDistance as Double	667
* 9.63.9	headerRectOfColumn(Column as Integer) as NSRectMBS	667
* 9.63.10	resizedColumn as Integer	667
* 9.63.12	tableView as NSTableViewMBS	667
– 9.64.1	class NSTableRowViewMBS	668
* 9.64.3	Constructor	668

	121
* 9.64.5 backgroundColor as NSColorMBS	668
* 9.64.6 emphasized as Boolean	668
* 9.64.7 Floating as Boolean	669
* 9.64.8 groupRowStyle as Boolean	669
* 9.64.9 NextRowSelected as Boolean	669
* 9.64.10 PreviousRowSelected as Boolean	669
* 9.64.11 selected as Boolean	669
* 9.64.12 selectionHighlightStyle as Integer	670
– 9.65.1 class NSTableViewMBS	671
* 9.65.3 addTableColumn(column as NSTableColumnMBS)	671
* 9.65.4 beginUpdates	672
* 9.65.5 canDragRowsWithIndexes(rowIndexes as NSIndexSetMBS, mouseDownPoint as NSPointMBS) as Boolean	672
* 9.65.6 columnAtPoint(p as NSPointMBS) as Integer	672
* 9.65.7 columnAtPoint(x as Double, y as Double) as Integer	673
* 9.65.8 columnForView(view as NSViewMBS) as Integer	673
* 9.65.9 columnIndexesInRect(rect as NSRectMBS) as NSIndexSetMBS	673
* 9.65.10 columnWithIdentifier(identifier as string) as Integer	673
* 9.65.11 Constructor	674
* 9.65.12 Constructor(Handle as Integer)	674
* 9.65.13 Constructor(left as Double, top as Double, width as Double, height as Double)	674
* 9.65.14 deselectAll	675
* 9.65.15 deselectColumn(column as Integer)	675
* 9.65.16 deselectRow(row as Integer)	675
* 9.65.17 Destructor	675
* 9.65.18 dragImageForRowsWithIndexes(dragRows as NSIndexSetMBS, tableColumns() as NSTableColumnMBS, theEvent as NSEventMBS, byref dragImageOffset as NSPointMBS) as NSImageMBS	675
* 9.65.19 edit(column as Integer, row as Integer, selectit as boolean)	676
* 9.65.20 endUpdates	676
* 9.65.21 frameOfCellAtColumnRow(column as Integer, row as Integer) as NSRectMBS	676
* 9.65.22 hiddenRowIndexes as NSIndexSetMBS	677
* 9.65.23 hideRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)	677
* 9.65.24 insertRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)	677
* 9.65.25 isColumnSelected(column as Integer) as boolean	678
* 9.65.26 isRowSelected(row as Integer) as boolean	678
* 9.65.27 moveColumn(column as Integer, toIndex as Integer)	678
* 9.65.28 moveRowAtIndex(oldIndex as Integer, newIndex as Integer)	678
* 9.65.29 noteHeightOfRowsWithIndexesChanged(indexSet as NSIndexSetMBS)	679
* 9.65.30 noteNumberOfRowsChanged	679
* 9.65.31 rectOfColumn(column as Integer) as NSRectMBS	679
* 9.65.32 rectOfRow(row as Integer) as NSRectMBS	680

* 9.65.33 reloadData	680
* 9.65.34 reloadData(rowIndexes as NSIndexSetMBS, columnIndexes as NSIndexSetMBS)	680
* 9.65.35 removeRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)	681
* 9.65.36 removeTableColumn(column as NSTableColumnMBS)	681
* 9.65.37 rowAtPoint(p as NSPointMBS) as Integer	681
* 9.65.38 rowAtPoint(x as Double, y as Double) as Integer	682
* 9.65.39 rowForView(view as NSViewMBS) as Integer	682
* 9.65.40 rowsInRect(rect as NSRectMBS) as NSRangeMBS	682
* 9.65.41 rowViewAtRow(row as Integer, makeIfNecessary as Boolean) as NSViewMBS	683
* 9.65.42 scrollColumnToVisible(column as Integer)	683
* 9.65.43 scrollRowToVisible(row as Integer)	683
* 9.65.44 ScrollToLine(Line as Integer, Animated as Boolean)	684
* 9.65.45 selectAll	684
* 9.65.46 selectColumnIndexes(indexes as NSIndexSetMBS, extend as boolean)	684
* 9.65.47 selectedColumnIndexes as NSIndexSetMBS	684
* 9.65.48 selectedRowIndexes as NSIndexSetMBS	684
* 9.65.49 selectRowIndexes(indexes as NSIndexSetMBS, extend as boolean)	685
* 9.65.50 setDraggingSourceOperationMask(mask as Integer, isLocal as Boolean)	685
* 9.65.51 setDropRow(row as Integer, dropOperation as Integer)	685
* 9.65.52 setSortDescriptor(sortDescriptor as NSSortDescriptorMBS)	685
* 9.65.53 setSortDescriptors(sortDescriptors() as NSSortDescriptorMBS)	686
* 9.65.54 sizeLastColumnToFit	686
* 9.65.55 sizeToFit	686
* 9.65.56 sortDescriptors as NSSortDescriptorMBS()	686
* 9.65.57 tableColumns as NSTableColumnMBS()	687
* 9.65.58 tableColumnWithIdentifier(identifier as string) as NSTableColumnMBS	687
* 9.65.59 tile	687
* 9.65.60 unhideRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)	687
* 9.65.61 viewAtColumn(column as Integer, row as Integer, makeIfNecessary as Boolean) as NSViewMBS	687
* 9.65.63 allowsColumnReordering as boolean	688
* 9.65.64 allowsColumnResizing as boolean	688
* 9.65.65 allowsColumnSelection as boolean	688
* 9.65.66 allowsEmptySelection as boolean	689
* 9.65.67 allowsMultipleSelection as boolean	689
* 9.65.68 allowsTypeSelect as boolean	689
* 9.65.69 autosaveName as string	689
* 9.65.70 autosaveTableColumns as boolean	689
* 9.65.71 backgroundColor as NSColorMBS	690
* 9.65.72 clickedColumn as Integer	690
* 9.65.73 clickedRow as Integer	690
* 9.65.74 columnAutoresizingStyle as Integer	691

	123
* 9.65.75 cornerView as NSViewMBS	691
* 9.65.76 dataSource as NSTableDataSourceMBS	691
* 9.65.77 draggingDestinationFeedbackStyle as Integer	691
* 9.65.78 editedColumn as Integer	692
* 9.65.79 editedRow as Integer	692
* 9.65.80 effectiveRowSizeStyle as Integer	692
* 9.65.81 effectiveStyle as Integer	692
* 9.65.82 floatsGroupRows as Boolean	692
* 9.65.83 focusedColumn as Integer	693
* 9.65.84 gridColor as NSColorMBS	693
* 9.65.85 gridStyleMask as Integer	693
* 9.65.86 headerView as NSTableHeaderViewMBS	694
* 9.65.87 highlightedtableColumn as NSTableColumnMBS	694
* 9.65.88 intercellSpacing as NSSizeMBS	694
* 9.65.89 numberOfColumns as Integer	694
* 9.65.90 numberOfRows as Integer	694
* 9.65.91 numberOfSelectedColumns as Integer	695
* 9.65.92 numberOfSelectedRows as Integer	695
* 9.65.93 rowActionsVisible as Boolean	695
* 9.65.94 rowHeight as Double	695
* 9.65.95 rowSizeStyle as Integer	695
* 9.65.96 selectedColumn as Integer	696
* 9.65.97 selectedRow as Integer	696
* 9.65.98 selectionHighlightStyle as Integer	696
* 9.65.99 style as Integer	697
* 9.65.100 usesAlternatingRowBackgroundColors as boolean	697
* 9.65.101 usesStaticContents as Boolean	697
* 9.65.102 verticalMotionCanBeginDrag as boolean	698
* 9.65.103 indicatorImageInTableColumn(column as NSTableColumnMBS) as NSImageMBS	698
* 9.65.105 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)	699
* 9.65.106 ColumnDidResize(notification as NSNotificationMBS, column as NSTableColumnMBS, index as Integer)	699
* 9.65.107 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS	699
* 9.65.108 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)	699
* 9.65.109 didClickTableColumn(tableColumn as NSTableColumnMBS)	700
* 9.65.110 didDragTableColumn(tableColumn as NSTableColumnMBS)	700
* 9.65.111 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)	700
* 9.65.112 DoubleClick	700
* 9.65.113 heightOfRow(row as Int64) as Double	701
* 9.65.114 isGroupRow(row as Int64) as boolean	701

* 9.65.115	mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)	701
* 9.65.116	nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64	702
* 9.65.117	rowViewForRow(row as Integer) as NSTableRowViewMBS	702
* 9.65.118	SelectionDidChange(notification as NSNotificationMBS)	702
* 9.65.119	selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS	703
* 9.65.120	SelectionIsChanging(notification as NSNotificationMBS)	703
* 9.65.121	selectionShouldChangeInTableView as boolean	703
* 9.65.122	shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean	704
* 9.65.123	shouldReorderColumn(columnIndex as Int64, newIndex as Int64) as boolean	704
* 9.65.124	shouldSelectRow(row as Int64) as boolean	704
* 9.65.125	shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean	705
* 9.65.126	shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean	705
* 9.65.127	shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean	706
* 9.65.128	shouldTypeSelectForEvent(e as NSEventMBS, searchString as string) as Boolean	706
* 9.65.129	sizeToFitWidthOfColumn(column as Int64) as Double	706
* 9.65.130	textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean	707
* 9.65.131	textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean	707
* 9.65.132	toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string	707
* 9.65.133	typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string	708
* 9.65.134	view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS	708
* 9.65.135	willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)	709
– 9.66.1	class NSTableViewRowActionMBS	714
* 9.66.3	available as boolean	714
* 9.66.4	Constructor(Style as Integer, Title as String)	714
* 9.66.6	BackgroundColor as NSColorMBS	715
* 9.66.7	Handle as Integer	715
* 9.66.8	Image as NSImageMBS	715
* 9.66.9	Style as Integer	715
* 9.66.10	Title as String	716
* 9.66.12	Action(row as Integer)	716
– 9.67.1	control NSTokenFieldControlMBS	717
* 9.67.3	View as NSTokenFieldMBS	717

	125
* 9.67.5 BoundsChanged	717
* 9.67.6 Close	717
* 9.67.7 completionsForSubstring(substring as string, tokenIndex as Integer, byref selectedIndex as Integer) as Variant()	718
* 9.67.8 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	718
* 9.67.9 ContextualMenuAction(hitItem as MenuItem) as Boolean	718
* 9.67.10 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	718
* 9.67.11 displayStringForRepresentedObject(representedObject as Variant) as string	719
* 9.67.12 editingStringForRepresentedObject(representedObject as Variant) as string	719
* 9.67.13 EnableMenuItems	719
* 9.67.14 FrameChanged	719
* 9.67.15 GotFocus	720
* 9.67.16 hasMenuForRepresentedObject(representedObject as Variant) as boolean	720
* 9.67.17 LostFocus	720
* 9.67.18 menuForRepresentedObject(representedObject as Variant) as NSMenuMBS	720
* 9.67.19MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	721
* 9.67.20 MouseDrag(x as Integer, y as Integer)	721
* 9.67.21 MouseUp(x as Integer, y as Integer)	721
* 9.67.22 Open	721
* 9.67.23 readFromPasteboard(pboard as NSPasteboardMBS) as Variant()	722
* 9.67.24 representedObjectForEditingString(editingString as string) as Variant	722
* 9.67.25 ScaleFactorChanged(NewFactor as Double)	722
* 9.67.26 shouldAddObjects(tokens() as Variant, index as Integer) as Variant()	722
* 9.67.27 styleForRepresentedObject(representedObject as Variant) as Integer	723
* 9.67.28 TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	723
* 9.67.29 TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	724
* 9.67.30 TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)	724
* 9.67.31 textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	724
* 9.67.32 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean	725
* 9.67.33 tokenFieldAction	725
* 9.67.34 tokenFieldTextShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	725
* 9.67.35 tokenFieldTextShouldEndEditing(fieldEditor as NSTextMBS) as boolean	725
* 9.67.36 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)	725
* 9.67.37 writeRepresentedObjects(objects() as Variant, pboard as NSPasteboardMBS) as boolean	726

• 7 Cocoa	171
– 7.3.1 class NSTokenFieldMBS	185
* 7.3.3 Constructor	185
* 7.3.4 Constructor(Handle as Integer)	186
* 7.3.5 Constructor(left as Double, top as Double, width as Double, height as Double)	186
* 7.3.6 defaultCompletionDelay as Double	186
* 7.3.7 defaultTokenizingCharacterSet as NSCharacterSetMBS	187
* 7.3.8 objects as Variant()	187
* 7.3.9 setObjects(objects() as Variant)	187
* 7.3.11 completionDelay as Double	187
* 7.3.12 tokenizingCharacterSet as NSCharacterSetMBS	187
* 7.3.13 tokenStyle as Integer	188
* 7.3.15 completionsForSubstring(substring as string, tokenIndex as Integer, byref selectedIndex as Integer) as Variant()	188
* 7.3.16 displayStringForRepresentedObject(representedObject as Variant) as string	188
* 7.3.17 editingStringForRepresentedObject(representedObject as Variant) as string	188
* 7.3.18 hasMenuForRepresentedObject(representedObject as Variant) as boolean	189
* 7.3.19 menuForRepresentedObject(representedObject as Variant) as NSMenuItemMBS	189
* 7.3.20 readFromPasteboard(pboard as NSPasteboardMBS) as Variant()	189
* 7.3.21 representedObjectForEditingString(editingString as string) as Variant	189
* 7.3.22 shouldAddObjects(tokens() as Variant, index as Integer) as Variant()	190
* 7.3.23 styleForRepresentedObject(representedObject as Variant) as Integer	190
* 7.3.24 tokenFieldAction	190
* 7.3.25 tokenFieldTextShouldBeginEditing(fieldEditor as NSTextMBS) as boolean	191
* 7.3.26 tokenFieldTextShouldEndEditing(fieldEditor as NSTextMBS) as boolean	191
* 7.3.27 writeRepresentedObjects(objects() as Variant, pboard as NSPasteboardMBS) as boolean	191

	127
• 11 Cocoa Toolbar	775
– 11.3.1 class NSToolbarItemGroupMBS	782
* 11.3.3 Constructor(itemIdentifier as string)	782
* 11.3.4 SetSubItems(items() as NSToolbarItemMBS)	782
* 11.3.5 subitems as NSToolbarItemMBS()	782
– 11.4.1 class NSToolbarItemMBS	784
* 11.4.3 Constructor(Handle as Integer)	784
* 11.4.4 Constructor(itemIdentifier as string)	784
* 11.4.5 NSToolbarCloudSharingItemIdentifier as string	785
* 11.4.6 NSToolbarCustomizeToolbarItemIdentifier as string	785
* 11.4.7 NSToolbarFlexibleSpaceItemIdentifier as string	785
* 11.4.8 NSToolbarPrintItemIdentifier as string	785
* 11.4.9 NSToolbarSeparatorItemIdentifier as string	785
* 11.4.10 NSToolbarShowColorsItemIdentifier as string	786
* 11.4.11 NSToolbarShowFontsItemIdentifier as string	786
* 11.4.12 NSToolbarSidebarTrackingSeparatorItemIdentifier as string	786
* 11.4.13 NSToolbarSpaceItemIdentifier as string	786
* 11.4.14 NSToolbarToggleSidebarItemIdentifier as string	786
* 11.4.15 validate	786
* 11.4.17 allowsDuplicatesInToolbar as boolean	787
* 11.4.18 autovalidates as boolean	787
* 11.4.19 Bordered as Boolean	787
* 11.4.20 ClassName as String	788
* 11.4.21 ClassPath as String	788
* 11.4.22 Enabled as boolean	788
* 11.4.23 Handle as Integer	788
* 11.4.24 image as NSImageMBS	788
* 11.4.25 itemIdentifier as string	789
* 11.4.26 label as string	789
* 11.4.27 MaxSize as NSSizeMBS	789
* 11.4.28 menuFormRepresentation as NSMenuItemMBS	789
* 11.4.29 MinSize as NSSizeMBS	789
* 11.4.30 Navigational as Boolean	790
* 11.4.31 paletteLabel as string	790
* 11.4.32 tag as Integer	790
* 11.4.33 Title as String	790
* 11.4.34 toolbar as NSToolbarMBS	791
* 11.4.35 toolTip as string	791
* 11.4.36 view as NSViewMBS	791
* 11.4.37 visibilityPriority as Integer	791
– 11.5.1 class NSToolbarMBS	793

* 11.5.3 Constructor(Handle as Integer)	793
* 11.5.4 Constructor(Identifier as string)	793
* 11.5.5 insertItemWithIdentifier(identifier as string, atIndex as Integer)	794
* 11.5.6 items as NSToolbarItemMBS()	794
* 11.5.7 NSToolbarDidRemoveItemNotification as string	794
* 11.5.8 NSToolbarWillAddItemNotification as string	794
* 11.5.9 removeItemAtIndex(index as Integer)	794
* 11.5.10 runCustomizationPalette	795
* 11.5.11 validateVisibleItems	795
* 11.5.12 visibleItems as NSToolbarItemMBS()	795
* 11.5.14 allowsExtensionItems as Boolean	796
* 11.5.15 allowsUserCustomization as boolean	796
* 11.5.16 autosavesConfiguration as boolean	796
* 11.5.17 centeredItemIdentifier as String	796
* 11.5.18 configurationDictionary as dictionary	797
* 11.5.19 configurationDictionaryData as Memoryblock	797
* 11.5.20 customizationPaletteIsRunning as boolean	797
* 11.5.21 displayMode as Integer	798
* 11.5.22 fullScreenAccessoryView as NSViewMBS	798
* 11.5.23 fullScreenAccessoryViewMaxHeight as Double	798
* 11.5.24 fullScreenAccessoryViewMinHeight as Double	798
* 11.5.25 Handle as Integer	798
* 11.5.26 identifier as string	799
* 11.5.27 selectedItemIdentifier as string	799
* 11.5.28 showsBaselineSeparator as boolean	799
* 11.5.29 sizeMode as Integer	799
* 11.5.30 toolbarView as NSViewMBS	800
* 11.5.31 visible as boolean	800

	129
• 9 Cocoa Controls	297
– 9.68.1 control NSViewControlMBS	727
* 9.68.3 View as NSViewMBS	727
* 9.68.5 acceptsFirstMouse(e as NSEventMBS) as boolean	727
* 9.68.6 acceptsFirstResponder as boolean	728
* 9.68.7 becomeFirstResponder as boolean	728
* 9.68.8 beginGestureWithEvent(e as NSEventMBS) as boolean	728
* 9.68.9 canBecomeKeyView as boolean	728
* 9.68.10 Close	728
* 9.68.11 Closed	728
* 9.68.12 concludeDragOperation(sender as NSDraggingInfoMBS)	729
* 9.68.13 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	729
* 9.68.14 ContextualMenuItemAction(hitItem as MenuItem) as Boolean	729
* 9.68.15 draggingEnded(sender as NSDraggingInfoMBS)	730
* 9.68.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer	730
* 9.68.17 draggingExited(sender as NSDraggingInfoMBS)	730
* 9.68.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)	731
* 9.68.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	731
* 9.68.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer	731
* 9.68.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)	732
* 9.68.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer	732
* 9.68.23 drawFocusRingMask(g as NSGraphicsMBS) as boolean	733
* 9.68.24 DrawRect(g as NSGraphicsMBS, left as Double, top as Double, width as Double, height as Double)	733
* 9.68.25 EnableMenuItems	733
* 9.68.26 endGestureWithEvent(e as NSEventMBS) as boolean	733
* 9.68.27 focusRingMaskBounds as NSRectMBS	734
* 9.68.28 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean	734
* 9.68.29 isOpaque as boolean	734
* 9.68.30 keyDown(e as NSEventMBS) as boolean	734
* 9.68.31 keyUp(e as NSEventMBS) as boolean	735
* 9.68.32 magnifyWithEvent(e as NSEventMBS) as boolean	735
* 9.68.33 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuItemMBS) as NSMenuItemMBS	735
* 9.68.34 mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	736
* 9.68.35 mouseDownCanMoveWindow as boolean	736

* 9.68.36	mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	736
* 9.68.37	mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean	736
* 9.68.38	mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean	736
* 9.68.39	mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean	737
* 9.68.40	mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	737
* 9.68.41	Open	737
* 9.68.42	Opened	737
* 9.68.43	otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	737
* 9.68.44	otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	738
* 9.68.45	otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	738
* 9.68.46	performDragOperation(sender as NSDraggingInfoMBS) as boolean	738
* 9.68.47	prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean	738
* 9.68.48	pressureChange(e as NSEventMBS) as boolean	739
* 9.68.49	resignFirstResponder as boolean	739
* 9.68.50	rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean	739
* 9.68.51	rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean	739
* 9.68.52	rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean	740
* 9.68.53	rotateWithEvent(e as NSEventMBS) as boolean	740
* 9.68.54	ScaleFactorChanged(NewFactor as Double)	740
* 9.68.55	scrollWheel(e as NSEventMBS) as boolean	740
* 9.68.56	swipeWithEvent(e as NSEventMBS) as boolean	740
* 9.68.57	updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)	741
* 9.68.58	viewDidMoveToWindow	741
* 9.68.59	wantsPeriodicDraggingUpdates as boolean	741

	131
• 7 Cocoa	171
– ?? Globals	??
* 7.4.1 CenterResizeAddWindowMBS(win as window)	193
* 7.4.2 CenterResizeInstallMBS	193
* 7.4.3 CenterResizeRemoveWindowMBS(win as window)	193

- **9 Cocoa Controls** 297
 - 9.69.1 class PopupMenu 742
 - * 9.69.3 NSButtonMBS as NSButtonMBS 742
 - * 9.69.4 NSPopUpButtonMBS as NSPopUpButtonMBS 742

	133
• 9 Cocoa Controls	297
– 12.4.1 class ProgressBar	804
* 12.4.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS	804
– 9.70.1 class ProgressWheel	744
* 9.70.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS	744
– 9.71.1 class PushButton	745
* 9.71.3 NSButtonMBS as NSButtonMBS	745

• 16 Quartz Composer	949
– 16.2.1 class QCCompositionMBS	953
* 16.2.3 compositionWithData(data as MemoryBlock) as QCCompositionMBS	953
* 16.2.4 compositionWithData(data as string) as QCCompositionMBS	954
* 16.2.5 compositionWithFile(file as folderitem) as QCCompositionMBS	954
* 16.2.6 compositionWithFile(path as string) as QCCompositionMBS	955
* 16.2.7 Constructor	955
* 16.2.8 copy as QCCompositionMBS	955
* 16.2.9 getAttributes as dictionary	955
* 16.2.10 inputKeys as string()	956
* 16.2.11 outputKeys as string()	956
* 16.2.12 protocols as string()	956
* 16.2.13 QCCompositionAttributeBuiltInKey as string	956
* 16.2.14 QCCompositionAttributeCategoryKey as string	956
* 16.2.15 QCCompositionAttributeCopyrightKey as string	957
* 16.2.16 QCCompositionAttributeDescriptionKey as string	957
* 16.2.17 QCCompositionAttributeHasConsumersKey as string	957
* 16.2.18 QCCompositionAttributeIsTimeDependentKey as string	958
* 16.2.19 QCCompositionAttributeNameKey as string	958
* 16.2.20 QCCompositionCategoryDistortion as string	958
* 16.2.21 QCCompositionCategoryStylize as string	958
* 16.2.22 QCCompositionCategoryUtility as string	959
* 16.2.23 QCCompositionInputAudioPeakKey as string	959
* 16.2.24 QCCompositionInputAudioSpectrumKey as string	959
* 16.2.25 QCCompositionInputDestinationImageKey as string	959
* 16.2.26 QCCompositionInputImageKey as string	959
* 16.2.27 QCCompositionInputPaceKey as string	960
* 16.2.28 QCCompositionInputPreviewModeKey as string	960
* 16.2.29 QCCompositionInputPrimaryColorKey as string	960
* 16.2.30 QCCompositionInputScreenImageKey as string	960
* 16.2.31 QCCompositionInputSecondaryColorKey as string	960
* 16.2.32 QCCompositionInputSourceImageKey as string	960
* 16.2.33 QCCompositionInputTrackInfoKey as string	961
* 16.2.34 QCCompositionInputTrackPositionKey as string	961
* 16.2.35 QCCompositionInputTrackSignalKey as string	961
* 16.2.36 QCCompositionInputXKey as string	961
* 16.2.37 QCCompositionInputYKey as string	961
* 16.2.38 QCCompositionOutputImageKey as string	962
* 16.2.39 QCCompositionOutputWebPageURLKey as string	962
* 16.2.40 QCCompositionProtocolGraphicAnimation as string	962
* 16.2.41 QCCompositionProtocolGraphicTransition as string	962

	135
* 16.2.42 QCCompositionProtocolImageFilter as string	963
* 16.2.43 QCCompositionProtocolMusicVisualizer as string	963
* 16.2.44 QCCompositionProtocolScreenSaver as string	963
* 16.2.46 Description as String	964
* 16.2.47 Handle as Integer	964
* 16.2.48 identifier as String	964
* 16.2.49 Name as String	964
– 16.3.1 class QCCompositionRepositoryMBS	966
* 16.3.3 allCompositions as QCCompositionMBS()	966
* 16.3.4 Compositions(protocols() as String = nil, attributes as Dictionary = nil) as QCCompositionMBS()	966
* 16.3.5 compositionWithIdentifier(identifier as string) as QCCompositionMBS	967
* 16.3.6 Constructor	967
* 16.3.7 loadPlugIn(file as folderitem) as Boolean	968
* 16.3.8 loadPlugIn(path as string) as Boolean	968
* 16.3.9 QCCompositionRepositoryDidUpdateNotification as string	968
* 16.3.10 sharedCompositionRepository as QCCompositionRepositoryMBS	969
* 16.3.12 Handle as Integer	969
– 16.4.1 control QCViewControlMBS	970
* 16.4.3 View as QCViewMBS	970
* 16.4.5 BoundsChanged	970
* 16.4.6 Close	970
* 16.4.7 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean	971
* 16.4.8 ContextualMenuItemAction(hitItem as MenuItem) as Boolean	971
* 16.4.9 didCloseContextualMenu(menu as NSMenuItem, NSEvent as NSEventMBS)	971
* 16.4.10 DidStartRendering	971
* 16.4.11 DidStopRendering	971
* 16.4.12 EnableMenuItems	971
* 16.4.13 FrameChanged	972
* 16.4.14 GotFocus	972
* 16.4.15 LostFocus	972
* 16.4.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean	972
* 16.4.17 MouseDrag(x as Integer, y as Integer)	973
* 16.4.18 MouseUp(x as Integer, y as Integer)	973
* 16.4.19 Open	973
* 16.4.20 ScaleFactorChanged(NewFactor as Double)	973
* 16.4.21 willShowContextualMenu(menu as NSMenuItem, NSEvent as NSEventMBS)	974
– 16.5.1 class QCViewMBS	975
* 16.5.3 Constructor	975
* 16.5.4 Constructor(Handle as Integer)	976

* 16.5.5 Constructor(left as Double, top as Double, width as Double, height as Double)	976
* 16.5.6 erase	976
* 16.5.7 getAttributes as dictionary	977
* 16.5.8 inputKeys as string()	977
* 16.5.9 loadComposition(composition as QCCompositionMBS) as boolean	977
* 16.5.10 loadCompositionFromData(data as MemoryBlock) as boolean	977
* 16.5.11 loadCompositionFromData(data as string) as boolean	978
* 16.5.12 loadCompositionFromFile(file as folderitem) as boolean	978
* 16.5.13 loadCompositionFromFile(filepath as string) as boolean	978
* 16.5.14 outputKeys as string()	979
* 16.5.15 pauseRendering	979
* 16.5.16 resumeRendering	979
* 16.5.17 setValueForInputKey(value as Variant, key as string) as boolean	979
* 16.5.18 startRendering as boolean	979
* 16.5.19 stopRendering	980
* 16.5.20 unloadComposition	980
* 16.5.21 valueForInputKey(key as string) as Variant	980
* 16.5.22 valueForOutputKey(key as string) as Variant	980
* 16.5.24 autostartsRendering as boolean	981
* 16.5.25 eraseColor as NSColorMBS	981
* 16.5.26 eventForwardingMask as Integer	981
* 16.5.27 isPausedRendering as boolean	981
* 16.5.28 isRendering as boolean	982
* 16.5.29 loadedComposition as QCCompositionMBS	982
* 16.5.30 maxRenderingFrameRate as Double	982
* 16.5.31 snapshotImage as NSImageMBS	983

	137
• 13 CoreGraphics	805
– 13.1.1 class QuartzFilterManagerMBS	805
* 13.1.3 filterPanel as NSPanelMBS	805
* 13.1.4 filters as QuartzFilterMBS()	806
* 13.1.5 filtersInDomains(domains() as string) as QuartzFilterMBS()	806
* 13.1.6 filterView as QuartzFilterViewMBS	806
* 13.1.7 selectedFilter as QuartzFilterMBS	806
* 13.1.8 selectFilter(filter as QuartzFilterMBS) as boolean	807
* 13.1.10 Handle as Integer	807
* 13.1.12 didAddFilter(filter as QuartzFilterMBS)	807
* 13.1.13 didModifyFilter(filter as QuartzFilterMBS)	807
* 13.1.14 didRemoveFilter(filter as QuartzFilterMBS)	807
* 13.1.15 didSelectFilter(filter as QuartzFilterMBS)	807
– 13.2.1 class QuartzFilterMBS	809
* 13.2.3 applyToContext(CGContextHandle as Integer) as boolean	810
* 13.2.4 Constructor	810
* 13.2.5 localizedName as string	810
* 13.2.6 quartzFilterWithFile(file as folderitem) as QuartzFilterMBS	811
* 13.2.7 quartzFilterWithURL(url as string) as QuartzFilterMBS	811
* 13.2.8 removeFromContext(CGContextHandle as Integer)	811
* 13.2.9 url as string	811
* 13.2.11 Handle as Integer	812
– 13.3.1 class QuartzFilterViewMBS	813
* 13.3.3 Constructor	813
* 13.3.4 Constructor(Handle as Integer)	813
* 13.3.5 Constructor(left as Double, top as Double, width as Double, height as Double)	814
* 13.3.6 sizeToFit	814

• 9 Cocoa Controls	297
– 9.72.1 class Radiobutton	746
* 9.72.3 NSButtonMBS as NSButtonMBS	746
– 9.73.1 class ScrollBar	747
* 9.73.3 NSScrollerMBS as NSScrollerMBS	747
– 9.74.1 class Separator	748
* 9.74.3 NSBoxMBS as NSBoxMBS	748
– 9.75.1 class Slider	749
* 9.75.3 NSSliderMBS as NSSliderMBS	749
– 9.76.1 class TextArea	750
* 9.76.3 NSScrollViewMBS as NSScrollViewMBS	750
– 9.77.1 class UpDownArrows	752
* 9.77.3 NSStepperMBS as NSStepperMBS	752

	139
• 11 Cocoa Toolbar	775
– 18.2.1 class Window	988
* 18.2.3 NSToolbarMBS as NSToolbarMBS	988

Chapter 2

List of all classes

• BevelButton	297
• CanvasGesturesMBS	298
• Checkbox	304
• ComboBox	305
• CustomNSScrollerMBS	306
• CustomNSTokenFieldMBS	321
• CustomNSToolbarItemMBS	775
• CustomNSToolbarMBS	777
• CustomNSViewMBS	335
• DateTimePicker	351
• DesktopBevelButton	352
• DesktopButton	353
• DesktopCheckBox	354
• DesktopComboBox	355
• DesktopDateTimePicker	356
• DesktopDisclosureTriangle	357
• DesktopGroupBox	358
• DesktopImageViewer	359
• DesktopListbox	801

• DesktopPopupMenu	443
• DesktopProgressbar	802
• DesktopProgressWheel	445
• DesktopRadioButton	446
• DesktopRadioGroup	447
• DesktopScrollBar	448
• DesktopSeparator	449
• DesktopSlider	450
• DesktopTextArea	451
• DesktopUpDownArrows	453
• DesktopWindow	987
• DisclosureTriangle	454
• DRBurnMBS	815
• DRBurnProgressPanelMBS	834
• DRBurnSetupPanelMBS	841
• DRCDTextBlockMBS	845
• DRDeviceMBS	846
• DREraseMBS	879
• DREraseProgressPanelMBS	882
• DREraseSetupPanelMBS	885
• DRFileMBS	886
• DRFolderMBS	894
• DRFSObjectMBS	898
• DRMSFMBS	912
• DRSetupPanelMBS	916
• DRTrackMBS	920
• GroupBox	455
• ImageWell	456
• KeyValueCodingMBS	457

	143
• Listbox	803
• NSBoxMBS	461
• NSButtonMBS	475
• NSClipViewMBS	487
• NSCollectionViewFlowLayoutInvalidationContextMBS	240
• NSCollectionViewFlowLayoutMBS	242
• NSCollectionViewGridLayoutMBS	248
• NSCollectionViewItemMBS	252
• NSCollectionViewLayoutAttributesMBS	255
• NSCollectionViewLayoutInvalidationContextMBS	260
• NSCollectionViewLayoutMBS	264
• NSCollectionViewMBS	266
• NSCollectionViewSectionHeaderViewMBS	285
• NSCollectionViewTransitionLayoutMBS	287
• NSCollectionViewUpdateItemMBS	290
• NSComboBoxMBS	500
• NSDatePickerMBS	512
• NSHTTPCookieMBS	753
• NSHTTPCookieStorageMBS	768
• NSImageViewMBS	518
• NSIndexPathMBS	292
• NSLevelIndicatorMBS	171
• NSOutlineViewItemMBS	553
• NSOutlineViewMBS	557
• NSPopoverMBS	176
• NSPopUpButtonMBS	576
• NSProgressIndicatorMBS	586
• NSScrollerMBS	592
• NSScrollViewMBS	602

• NSSliderMBS	614
• NSSearchBarButtonMBS	985
• NSStepperMBS	621
• NSSwitchMBS	629
• NSTableColumnMBS	632
• NSTableDataSourceMBS	662
• NSTableHeaderCellMBS	664
• NSTableHeaderViewMBS	665
• NSTableRowViewMBS	668
• NSTableViewMBS	671
• NSTableViewRowActionMBS	714
• NSTokenFieldMBS	185
• NSToolbarItemGroupMBS	782
• NSToolbarItemMBS	784
• NSToolbarMBS	793
• PopupMenu	742
• ProgressBar	804
• ProgressWheel	744
• PushButton	745
• QCCompositionMBS	953
• QCCompositionRepositoryMBS	966
• QCViewMBS	975
• QuartzFilterManagerMBS	805
• QuartzFilterMBS	809
• QuartzFilterViewMBS	813
• Radiobutton	746
• ScrollBar	747
• Separator	748
• Slider	749
• TextArea	750
• UpDownArrows	752
• Window	988

Chapter 3

List of all controls

• DesktopNSButtonControlMBS	360
• DesktopNSCollectionViewControlMBS	195
• DesktopNSComboBoxControlMBS	365
• DesktopNSDatePickerControlMBS	370
• DesktopNSOutlineControlMBS	374
• DesktopNSPopUpButtonControlMBS	402
• DesktopNSSwitchControlMBS	406
• DesktopNSTableControlMBS	410
• DesktopNSTokenFieldControlMBS	434
• DesktopQCViewControlMBS	949
• NSButtonControlMBS	469
• NSCollectionViewControlMBS	217
• NSComboBoxControlMBS	494
• NSDatePickerControlMBS	507
• NSOutlineControlMBS	524
• NSPopUpButtonControlMBS	571
• NSSwitchControlMBS	624
• NSTableControlMBS	637
• NSTokenFieldControlMBS	717
• NSViewControlMBS	727
• QCViewControlMBS	970

Chapter 4

List of all global methods

- 7.4.1 CenterResizeAddWindowMBS(win as window) 193
- 7.4.2 CenterResizeInstallMBS 193
- 7.4.3 CenterResizeRemoveWindowMBS(win as window) 193
- 15.1.1 InstallDragImageMBS 947
- 15.1.2 SetNextDragImageMBS(Img as NSImageMBS, DragItemCount as Integer = 1) 947

Chapter 5

Addressbook

5.1 control ABPeoplePickerViewControlMBS

5.1.1 control ABPeoplePickerViewControlMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use CNContactPickerMBS class instead. **Function:** The control to host a people picker view.

Notes: Shows people and groups from Addressbook and allows picking them.

Blog Entries

- [Saying goodbye to AddressBook framework](#)
- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 15.0](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr9](#)

Xojo Developer Magazine

- [13.2, page 8: News](#)

5.1.2 Properties

5.1.3 View as ABPeoplePickerViewMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The view used.

Notes: Access properties of the view hosted in the control via this variable.
(Read only property)

5.1.4 Events

5.1.5 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

5.1.6 Close

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The control is about to close.

5.1.7 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

5.1.8 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

5.1.9 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

5.1.10 DisplayedPropertyDidChange

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Called when the displayed property in the record list is changed.

5.1.11 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event where you can enable menu items.

5.1.12 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

5.1.13 GotFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function: The control itself got focus.

Notes: This only fires if the control itself got focus and not a sub control.

5.1.14 GroupDoubleClick

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The event to be invoked when a group is double-clicked.

5.1.15 GroupSelectionDidChange

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Called when the selection in the group list is changed.

5.1.16 LostFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function: The control lost focus.

Notes: This only fires if the control itself lost focus and not a sub control.

5.1.17MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

5.1.18 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

5.1.19 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

5.1.20 NameDoubleClick

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The event to be invoked when a name is double-clicked.

5.1.21 NameSelectionDidChange

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Called when the selection in the name list is changed.

5.1.22 Open

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The control is about to be created and you can initialize it.

5.1.23 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

5.1.24 ValueSelectionDidChange

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Called when the selection in a multivalue property is changed.

5.1.25 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

5.2 class ABPeoplePickerViewMBS

5.2.1 class ABPeoplePickerViewMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use `CNContactViewControllerMBS` class instead. **Function:** This is a view for Mac applications to select people from the address-book.

Notes: The `ABPeoplePickerView` class allows you to customize the behavior of people-picker views in an application's user interface.

Use `CocoaControlMBS` control to put this view on a window.

You can embed this view in a `CustomNSViewMBS` to get more events for mouse and keyboard. Subclass of the `NSViewMBS` class.

Blog Entries

- [Saying goodbye to AddressBook framework](#)
- [MBS Real Studio Plugins, version 13.0pr7](#)
- [MBS Real Studio Plugins, version 12.3pr10](#)

5.2.2 Methods

5.2.3 ABPeoplePickerDisplayedPropertyDidChangeNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names.

Notes: Use this string with `NSNotificationObserverMBS` class. Or use the `DisplayedPropertyDidChange` event.

Posted when the displayed property in the record list is changed.

5.2.4 ABPeoplePickerGroupSelectionDidChangeNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names.

Notes: Use this string with `NSNotificationObserverMBS` class. Or use the `GroupSelectionDidChange` event. Posted when the selection in the group list is changed.

5.2.5 ABPeoplePickerNameSelectionDidChangeNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names.

Notes: Use this string with NSNotificationObserverMBS class. Or use the NameSelectionDidChange event. Posted when the selection in the name list is changed.

5.2.6 ABPeoplePickerValueSelectionDidChangeNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names.

Notes: Use this string with NSNotificationObserverMBS class. Or use the ValueSelectionDidChange event. Posted when the selection in a multivalue property is changed.

5.2.7 addProperty(PropertyName as string)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Adds a property to the group of properties whose values are shown in the record list.

5.2.8 clearSearchField

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Clears the search field and resets the list of displayed records.

5.2.9 Constructor

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new people picker view object.

See also:

- 5.2.10 Constructor(Handle as Integer) 156
- 5.2.11 Constructor(left as Double, top as Double, width as Double, height as Double) 156

5.2.10 Constructor(Handle as Integer)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new people picker view object.

Notes: Pass a handle to a Cocoa ABPeoplePickerView object.

See also:

- 5.2.9 Constructor 155
- 5.2.11 Constructor(left as Double, top as Double, width as Double, height as Double) 156

5.2.11 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new people picker view object.

Notes: Pass rectangle for new control.

See also:

- 5.2.9 Constructor 155
- 5.2.10 Constructor(Handle as Integer) 156

5.2.12 deselectAll

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Deselects all selected groups, records, and values in multivalue properties.

5.2.13 deselectGroup(group as ABGroupMBS)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Deselects a group selected in the group list.

5.2.14 deselectIdentifier(identifier as string, person as ABPersonMBS)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Deselects a value selected in a multivalue property.

Notes: identifier: The identifier of the value that will be deselected.

person: The person whose value will be deselected.

5.2.15 `deselectRecord(record as ABRecordMBS)`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Deselects a record selected in the record list.

Notes: record: The record to deselect.

5.2.16 `editInAddressBook`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Launches Address Book to edit the item selected in the people picker.

5.2.17 `properties as string()`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Returns an array of the properties whose values are shown in the record list.

5.2.18 `removeProperty(PropertyName as string)`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Removes a property from the group of properties whose values are shown in the record list.

5.2.19 `selectedGroups as ABGroupMBS()`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The groups selected in the group list.

Notes: The selected groups are returned as an array of ABGroupMBS objects.

Available in Mac OS X v10.6 and later.

5.2.20 `selectedIdentifiersForPerson(person as ABPersonMBS) as string()`

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Returns the identifiers of the selected values in a multivalue property.

Notes: person: The person whose identifiers for selected values will be returned.

Returns nil if the property displayed is a single-value property.

5.2.21 selectedRecords as ABRecordMBS()

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The selection in the records list.

Notes: The selection is returned as an array of ABGroupMBS or ABPersonMBS objects.
Available in Mac OS X v10.6 and later.

5.2.22 selectedValues as Variant()

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Returns an array of all the values selected in the displayed multivalue property.

5.2.23 selectGroup(group as ABGroupMBS, byExtendingSelection as boolean)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Selects a group or a set of groups in the group list.

Notes: group: The group to be selected, or to be added to the current selection.
byExtendingSelection: True to extend the current selection; otherwise, false.

5.2.24 selectIdentifier(identifier as string, person as ABPersonMBS, byExtendingSelection as boolean)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Selects a value or a set of values in a multivalue property.

Notes: identifier: The identifier to be selected, or to be added to the current selection.
person: The person that the value to be selected is associated with.
byExtendingSelection: True to extend the current selection; otherwise, false.

5.2.25 selectInAddressBook

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Launches Address Book and selects the item selected in the people picker.

5.2.26 selectRecord(group as ABRecordMBS, byExtendingSelection as boolean)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Selects a record or a set of records in the record list.

Notes: record: The record to be selected, or to be added to the current selection.

byExtendingSelection: True to extend the current selection; otherwise, false.

5.2.27 Properties**5.2.28 accessoryView as NSViewMBS**

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The view that is placed to the left of the search field.

Notes: Available in Mac OS X v10.6 and later.

If accessory is nil, the accessory view is removed.

(Read and Write property)

5.2.29 allowsGroupSelection as boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that specifies whether the user can select entire groups in the group column.

Notes: If true, the user can to select entire groups. If false, the user is required to select at least one person in the group.

Available in Mac OS X v10.6 and later.

(Read and Write property)

5.2.30 allowsMultipleSelection as boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that specifies whether multiple groups, records, or values of multivalued properties can be selected at a time.

Notes: Available in Mac OS X v10.6 and later.

(Read and Write property)

5.2.31 `autosaveName` as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The name under which the column positions and the filter selection are saved.

Notes: Available in Mac OS X v10.6 and later.

(Read and Write property)

5.2.32 `displayedProperty` as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The property currently displayed in the record list.

Notes: Available in Mac OS X v10.6 and later.

(Read and Write property)

5.2.33 `valueSelectionBehavior` as Integer

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The current selection behavior.

Notes: The default behavior is `ABSingleValueSelection`.

Available in Mac OS X v10.6 and later.

(Read and Write property)

5.2.34 `columnTitleForProperty(propertyName as string)` as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The title of a custom property.

Notes: (Read and Write computed property)

5.2.35 Events

5.2.36 `DisplayedPropertyDidChange`

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Called when the displayed property in the record list is changed.

5.2.37 GroupDoubleClick

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: The event to be invoked when a group is double-clicked.

5.2.38 GroupSelectionDidChange

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Called when the selection in the group list is changed.

5.2.39 NameDoubleClick

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: The event to be invoked when a name is double-clicked.

5.2.40 NameSelectionDidChange

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Called when the selection in the name list is changed.

5.2.41 ValueSelectionDidChange

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Called when the selection in a multivalue property is changed.

5.2.42 Constants

Constants for the Selection Behavior type.

Constant	Value	Description
ABMultipleValueSelection	2	The user can select multiple values.
ABNoValueSelection	0	The user cannot select individual values.
ABSingleValueSelection	1	The user can select a single value.

5.3 control ABPersonViewControlMBS

5.3.1 control ABPersonViewControlMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use `CNContactViewControllerMBS` class instead. **Function:** The control to host a Person view.

Notes: Shows people from Addressbook and allows editing.

Blog Entries

- [Saying goodbye to AddressBook framework](#)
- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 15.0](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr9](#)

5.3.2 Methods

5.3.3 retainObject

Plugin Version: 16.0, Platform: macOS, Targets: Desktop only.

Function: Retains the viewer object.

Notes: This method is to workaroud a bug which your app may see in OS X 10.11. So to avoid crashes with releasing an view, you can retain it an extra time.

5.3.4 Properties

5.3.5 View as ABPersonViewMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The view used.

Notes: Access properties of the view hosted in the control via this variable. (Read only property)

5.3.6 Events

5.3.7 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

5.3.8 Close

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The control is about to close.

5.3.9 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

5.3.10 ContextualMenuItemAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuItem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

5.3.11 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

5.3.12 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event where you can enable menu items.

5.3.13 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

5.3.14 GotFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function: The control itself got focus.

Notes: This only fires if the control itself got focus and not a sub control.

5.3.15 LostFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function: The control lost focus.

Notes: This only fires if the control itself lost focus and not a sub control.

5.3.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

5.3.17 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

5.3.18 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

5.3.19 Open

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The control is about to be created and you can initialize it.

5.3.20 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

5.3.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

5.4 class ABPersonViewMBS

5.4.1 class ABPersonViewMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Deprecated: This item is deprecated and should no longer be used. You can use `CNContactViewControllerMBS` instead. **Function:** `ABPersonViewMBS` provides a view for displaying and editing `ABPersonMBS` objects in your user interface.

Notes: Available on Mac OS X 10.7 or later.

Please also check the documentation from Apple for the `ABPersonView` class.

You can embed this view in a `CustomNSViewMBS` to get more events for mouse and keyboard. Subclass of the `NSViewMBS` class.

Blog Entries

- [Saying goodbye to AddressBook framework](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr12](#)
- [MBS Real Studio Plugins, version 13.0pr7](#)
- [Lion features for Real Studio](#)
- [Lion arrived](#)

5.4.2 Methods

5.4.3 available as Boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Whether this class is available.

Example:

```
msgbox "ABPersonView available: "+str(ABPersonViewMBS.available)
```

Notes: Returns true on Mac OS X 10.7 or later.

5.4.4 Constructor

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new people picker view object.

See also:

5.4. CLASS ABPERSONVIEWMBS	167
• 5.4.5 Constructor(Handle as Integer)	167
• 5.4.6 Constructor(left as Double, top as Double, width as Double, height as Double)	167

5.4.5 Constructor(Handle as Integer)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new people picker view object.

Notes: Pass a handle to a Cocoa ABPersonView object.

See also:

- 5.4.4 Constructor 166
- 5.4.6 Constructor(left as Double, top as Double, width as Double, height as Double) 167

5.4.6 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new people picker view object.

Notes: Pass rectangle for new control.

See also:

- 5.4.4 Constructor 166
- 5.4.5 Constructor(Handle as Integer) 167

5.4.7 Properties

5.4.8 editing as Boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that determines whether the receiver is in editing mode.

Notes: When true, ABPersonView includes additional controls to manipulate person properties.

(Read and Write property)

5.4.9 person as ABPersonMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: An ABPersonMBS record for display.

Notes: Raises if person originates from ABAddressBook's sharedAddressBook. Use ABAddressBookMBS

constructor.

Person must exist in an ABAddressBook created and manipulated on the main thread only.

When person is nil, displays an empty selection state.

(Read and Write property)

5.4.10 shouldShowLinkedPeople as Boolean

Plugin Version: 12.3, Platform: macOS, Targets: Desktop only.

Function: Indicates whether the view should include information for linked contacts in addition to the set contact.

Notes: If true, information is included from linked contacts. If false, only the information on this person is shown.

Available in Mac OS X 10.8 and newer.

(Read and Write property)

Chapter 6

AVFoundation

Chapter 7

Cocoa

7.1 class NSLevelIndicatorMBS

7.1.1 class NSLevelIndicatorMBS

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: NSLevelIndicatorMBS is a subclass of NSControlMBS that displays a value on a linear scale.

Notes: Level indicators provide a visual representation of a level or amount of something, using discrete values. While similar to NSSlider, it provides a more customized visual feedback to the user. Level indicators do not have a “knob” indicating the current setting or allowing the user to adjust settings. The supported indicator styles include:

- A capacity style level indicator. The continuous mode for this style is often used to indicate conditions such as how much data is on hard disk. The discrete mode is similar to audio level indicators in audio playback applications. You can specify both a warning value and a critical value that provides additional visual feedback to the user.
- A ranking style level indicator. This is similar to the star ranking displays provided in iTunes and iPhoto. You can also specify your own ranking image.
- A relevancy style level indicator. This style is used to display the relevancy of a search result, for example in Mail.

NSLevelIndicator uses an NSLevelIndicatorCell to implement much of the control’s functionality. NSLevelIndicator provides cover methods for most of NSLevelIndicatorCell’s methods, which invoke the corresponding cell method.

Subclass of the NSControlMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 12.2pr6](#)

7.1.2 Methods

7.1.3 Constructor

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new level indicator with size 100/100 and position 0/0

Example:

```
dim t as new NSLevelIndicatorMBS
```

Notes: On success the handle property is not zero.

See also:

- 7.1.4 Constructor(Handle as Integer) 172
- 7.1.5 Constructor(left as Double, top as Double, width as Double, height as Double) 172

7.1.4 Constructor(Handle as Integer)

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSLevelIndicator handle.

Example:

```
dim t as new NSLevelIndicatorMBS(0, 0, 100, 100)
dim v as new NSLevelIndicatorMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSLevelIndicator and the plugin retains this handle.

See also:

- 7.1.3 Constructor 172
- 7.1.5 Constructor(left as Double, top as Double, width as Double, height as Double) 172

7.1.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new level indicator with the given size and position.

Example:

`dim x as new NSLevelIndicatorMBS(0, 0, 100, 100)`

Notes: On success the handle property is not zero.
See also:

- 7.1.3 Constructor 172
- 7.1.4 Constructor(Handle as Integer) 172

7.1.6 `rectOfTickMarkAtIndex(index as Integer) as NSRectMBS`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Returns the bounding rectangle of the tick mark identified by index (the minimum-value tick mark is at index 0).

Notes: If no tick mark is associated with index, the method raises a `NSExcptionMBS`.

7.1.7 `tickMarkValueAtIndex(index as Integer) as Double`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Returns the receiver's value represented by the tick mark at index (the minimum-value tick mark has an index of 0).

7.1.8 Properties

7.1.9 `criticalValue as Double`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The critical value.

Notes: (Read and Write computed property)

7.1.10 `levelIndicatorStyle as Integer`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Set style of the indicator.

Notes: (Read and Write computed property)

7.1.11 `maxValue` as Double

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The maximum value the level indicator can represent.

Notes: (Read and Write computed property)

7.1.12 `minValue` as Double

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The minimum value.

Notes: (Read and Write computed property)

7.1.13 `numberOfMajorTickMarks` as Integer

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The number of major tick marks displayed.

Notes: The count must be less than or equal to the number of tick marks returned by `numberOfTickMarks`. For example, if the number of tick marks is 11 and you specify 3 major tick marks, the resulting level indicator will display 3 major tickmarks alternating with 8 minor tick marks.

(Read and Write computed property)

7.1.14 `numberOfTickMarks` as Integer

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The number of tick marks displayed by the receiver (which include those assigned to the minimum and maximum values) to count.

Notes: By default, this value is 0, and no tick marks appear. The number of tick marks assigned to a slider, along with the slider's minimum and maximum values, determines the values associated with the tick marks.

(Read and Write computed property)

7.1.15 `tickMarkPosition` as Integer

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Where tick marks appear relative to the indicator.

Notes: This method has no effect if no tick marks have been assigned (that is, `numberOfTickMarks` returns 0).

(Read and Write computed property)

7.1.16 warningValue as Double

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The warning value.

Notes: (Read and Write computed property)

7.1.17 Constants

Style Constants

Constant	Value	Description
NSContinuousCapacityLevelIndicatorStyle	1	A style that is often used to indicate conditions such as how much free space is left on a hard disk.
NSDiscreteCapacityLevelIndicatorStyle	2	A style similar to audio level indicators in iTunes.
NSRatingLevelIndicatorStyle	3	A style similar to the star ranking displays provided in iTunes and iBooks.
NSRelevancyLevelIndicatorStyle	0	A style similar to the rank column displayed when searching in Mail.

Tick mark Constants

Constant	Value	Description
NSTickMarkAbove	1	Tick marks above (for horizontal sliders).
NSTickMarkBelow	0	Tick marks below (for horizontal sliders); the default for horizontal sliders.
NSTickMarkLeft	0	Tick marks to the left (for vertical sliders); the default for vertical sliders.
NSTickMarkRight	1	Tick marks to the right (for vertical sliders).

7.2 class NSPopoverMBS

7.2.1 class NSPopoverMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The Xojo class for a pop-over.

Notes: A popover is a unit of content that is positioned relative to some other content on the screen. An anchor is used to express the relation between these two units of content. Each popover has an appearance that specifies its visual characteristics, as well as a behavior that determines which user interactions will cause the popover to close. A transient popover is closed in response to most user interactions, whereas a semi-transient popovers is closed when the user interacts with the window containing the popover's positioning view. Popovers with application-defined behavior are not usually closed on the developer's behalf. AppKit automatically positions each popover relative to its positioning view and moves the popover whenever its positioning view moves. A positioning rectangle within the positioning view can be specified for additional granularity. Popovers can be detached to become a separate window when they are dragged by implementing the appropriate delegate method.

Please also check the documentation from Apple for the NSPopover class.
Available in Mac OS X v10.7 and later.

Blog Entries

- [News from the MBS Xojo Plugins Version 23.2](#)
- [News from the MBS Xojo Plugins Version 22.4](#)
- [MBS Xojo Plugins in version 22.4](#)
- [MBS Xojo Plugins, version 22.4pr1](#)
- [MBS Xojo Plugins, version 20.3pr7](#)
- [Lion features for Real Studio](#)

7.2.2 Methods

7.2.3 available as boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Whether pop-overs are available.

Example:

```
msgbox "NSPopover available: "+str(NSPopoverMBS.available)
```

Notes: Returns true on Mac OS X 10.7.

7.2.4 Close

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Forces the popover to close without calling events.

Notes: Any popovers nested within the popovers will also receive a close message. When a window is closed in response to the close message being sent, all of its popovers will be closed. The popover will animate out when closed (unless the animates property is set to false).

7.2.5 Constructor

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new popover.

7.2.6 Destructor

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The destructor.

7.2.7 isShown as boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Whether popover is visible.

Notes: True if the popover is being shown, false otherwise. The popover is considered to be shown from the point when showRelativeToRect is invoked until the popover is closed in response to an invocation of either close or performClose.

7.2.8 NSPopoverCloseReasonDetachToWindow as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the constant for the close reason.

Notes: Specifies that the popover has been closed because it is being detached to a window; a possible value for NSPopoverCloseReasonKey.

7.2.9 NSPopoverCloseReasonKey as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the constants for the close notification parameter dictionary.

Notes: Specifies the close reason. Currently used only as the userInfo key for the NSPopoverWillCloseNotification.

7.2.10 NSPopoverCloseReasonStandard as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the constant for the close reason.

Notes: Specifies that the popover is being closed in a standard way; a possible value for NSPopoverCloseReasonKey.

7.2.11 NSPopoverDidCloseNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names you can use with NSNotificationObserverMBS class.

Notes: Sent after the popover has finished animating offscreen. This notification has the same user info keys as NSPopoverWillCloseNotification.

7.2.12 NSPopoverDidShowNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names you can use with NSNotificationObserverMBS class.

Notes: Sent after the popover has finished animating onscreen.

7.2.13 NSPopoverWillCloseNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names you can use with NSNotificationObserverMBS class.

Notes: Sent before the popover is closed. The userInfo key NSPopoverCloseReasonKey specifies the reason for closing. It can currently be either NSPopoverCloseReasonStandard or NSPopoverCloseReasonDetachToWindow, although more reasons for closing may be added in the future.

7.2.14 NSPopoverWillShowNotification as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: One of the notification names you can use with NSNotificationObserverMBS class.

Notes: Sent before the popover is shown.

7.2.15 performClose

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Attempts to close the popover.

Notes: The popover will not be closed if popoverShouldClose event returns false. The operation will fail if it is displaying a nested popover, or if it has a child window. A window will attempt to close its popovers when it receives a performClose message. The popover will animate out when closed (unless the animates property is set to false).

7.2.16 showRelativeToRect(positioningRect as NSRectMBS, view as NSViewMBS, edge as Integer)

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Shows the popover positioned relative to positioningRect of positioningView (see the description of positioningRect).

Notes: The common case is to pass positioningView.bounds for positioningRect, in which case the popover will be placed adjacent to the positioningView and there is no need to update positioningRect (AppKit will detect the the bounds of the positioning view was specified and automatically update the positioningView). preferredEdge is a hint to AppKit about the desired placement of the anchor of the popover towards the positioningRect, and is with respect to the isFlipped state of the positioningView. Also, if positioningRect is an empty rect, the view.bounds will automatically be used. The current (but not guaranteed) behavior is that AppKit will place the anchor towards the preferredEdge of the positioningRect unless such a placement would cause the popover not to fit on the screen of positioningView. If the anchor cannot be placed towards the preferredEdge, AppKit will (in the current implementation) attempt to place the anchor on the opposite side of the positioningRect. If that cannot be done, AppKit will attempt to place the anchor on a remaining sides of the popover, and failing that will center the popover on the screen, causing it to (at least temporarily) lose its anchor. The popover will animate onscreen and eventually animate offscreen when it is closed (unless the property animates is set to false). This method will throw a NSInvalidArgumentException if view is nil or if view is not in a window, or if the popover's behavior is NSPopoverBehaviorSemitransient and the popover's positioningView is in a popover or child window. It will throw a NSInternalInconsistencyException if the popover's content view controller (or the view controller's view) is nil. If the popover is already being shown, this method will update to be associated with the new view and positioningRect passed. If the positioning view is not visible, this method does nothing.

7.2.17 Properties

7.2.18 Handle as Integer

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The internal reference to the NSPopover object.

Notes: (Read and Write property)

7.2.19 Tag as Variant

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The tag.

Notes: You can use this property for whatever you like.

e.g. to keep the reference to the NSViewControllerMBS object.

(Read and Write property)

7.2.20 animates as boolean

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Should the popover be animated when it shows, closes, or appears to transition to a detachable window.

Notes: This property also controls whether the popover animates when the content view or content size changes. AppKit does not guarantee which behaviors will be animated or that this property will be respected; it is regarded as a hint. The default value is true.

(Read and Write computed property)

7.2.21 behavior as Integer

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The behavior of the popover.

Notes: The default behavior is NSPopoverBehaviorApplicationDefined. See the declaration of NSPopoverBehavior* constants for more information about popover behaviors.

(Read and Write computed property)

7.2.22 contentSize as NSSizeMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The content size of the popover.

Notes: The popover's content size is set to match the size of the content view when the content view controller is set. Changes to the content size of the popover will animate while the popover is shown (provided animates is true).

(Read and Write computed property)

7.2.23 contentViewController as NSViewControllerMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The view controller that manages the content of the popover.

Notes: Please use with NSViewControllerMBS class.

The default value is nil. You must set the content view controller of the popover to a non-nil value before the popover is shown. Changes to the popover's content view controller while the popover is shown will animate (provided animates is true).

(Read and Write computed property)

7.2.24 positioningRect as NSRectMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The positioning rectangle.

Notes: Popovers are positioned relative to a positioning view and are automatically moved when the location or size of the positioning view changes. Sometimes it is desirable to position popovers relative to a rectangle within the positioning view. In this case, you must update the positioningRect binding whenever this rectangle changes, or use the positioningRect binding so AppKit can re-position the popover when appropriate.

(Read and Write computed property)

7.2.25 Events

7.2.26 detachableWindowForPopover as NSWindowMBS

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Detaches the popover creating a window containing the content.

Notes: You return a window instance to which the popover should be detached.

You should not remove the popover's content view as part of your implementation of this method.

The popover and the detachable window may be shown at the same time and therefore cannot share a

content view (or a content view controller).

If the popover and the detachable window should have the same content, you should define the content with a separate view and use a view controller to instantiate separate copies of the content for the popover and the detachable window.

The popover will animate to appear as though it morphs into the detachable window (unless the popover animates property is set to false. The exact animation used is not guaranteed.

If you do not implement this event or it returns nil, the default behavior is that the popup is not detached.

7.2.27 popoverDidClose(notification as NSNotificationMBS)

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Invoked when the popover did close.

Notes: Invoked on the delegate when the NSPopoverDidCloseNotification notification is sent.

7.2.28 popoverDidDetach

Plugin Version: 22.4, Platform: macOS, Targets: .

Function: Indicates that a popover has been released while it's in an implicitly detached state.

Notes: This event is not called when the popover's detached window is returned by detachableWindowForPopover.

7.2.29 popoverDidShow(notification as NSNotificationMBS)

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Invoked when the popover has been shown.

Notes: Invoked on the delegate when the NSPopoverDidShowNotification notification is sent.

7.2.30 popoverShouldClose as boolean

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Allows a delegate to override a close request.

Notes: Return true if the popover should close, false otherwise.

The popover invokes this method on its delegate whenever it is about to close. This gives you a chance to override the close.

If you do not implement this method the default behavior is that the popover will close.

7.2.31 popoverShouldDetach as boolean

Plugin Version: 22.4, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a popover should detach from its positioning view and become a separate window.

Notes: If you don't implement this event, it returns false by default. If you return true from this event, but you don't implement detachableWindowForPopover or you implement it to return nil, a detachable window is created with the popover's contentViewController.

An automatically created window has the same appearance as the detached popover. For example, if the popover's contentViewController has a title, it will be bound to and displayed as the title of the detached window. When a popover is released in a detached state, it calls popoverDidDetach event. When a detached popover is closed, calls to popoverShouldClose, popoverWillClose, and popoverDidClose, in addition to the related notifications, specify the reason NSPopoverCloseReasonStandard.

7.2.32 popoverWillClose(notification as NSNotificationMBS)

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Invoked when the popover is about to close.

Notes: Invoked on the delegate when the NSPopoverWillCloseNotification notification is sent.

7.2.33 popoverWillShow(notification as NSNotificationMBS)

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Invoked when the popover will show.

Notes: Invoked on the delegate when the NSPopoverWillShowNotification notification is sent.

7.2.34 Constants

Edge constants

Constant	Value	Description
MaxXEdge	2	right
MaxYEdge	3	top (Cocoa coordinates have 0 on bottom of screen)
MinXEdge	0	left
MinYEdge	1	bottom (Cocoa coordinates have 0 on bottom of screen)

Behavior modes

Constant	Value	Description
NSPopoverBehaviorApplicationDefined	0	Your application assumes responsibility for closing the popover. AppKit still close the popover in a limited number of circumstances. For instance, AppKit will attempt to close the popover when the window of its positioning view is closed. The exact interactions in which AppKit will close the popover are not guaranteed. You may consider implementing <code>-cancel:</code> to close the popover when the escape key is pressed.
NSPopoverBehaviorSemitransient	2	AppKit will close the popover when the user interacts with user interface elements in the window containing the popover's positioning view. Semi-transparent popovers cannot be shown relative to views in other popovers, nor can they be shown relative to views in child windows. The exact interactions that cause semi-transparent popovers to close are not specified.
NSPopoverBehaviorTransient	1	AppKit will close the popover when the user interacts with a user interface element outside the popover. Note that interacting with menus or panels that become key only when needed will not cause a transient popover to close. The exact interactions that will cause transient popovers to close are not specified.

7.3 class NSTokenFieldMBS

7.3.1 class NSTokenFieldMBS

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: NSTokenField is a subclass of NSTextField that provides tokenized editing similar to the address field in the Mail application.

Notes: NSTokenField uses an NSTokenFieldCell to implement much of the control's functionality. NSTokenField provides cover methods for most methods of NSTokenFieldCell, which invoke the corresponding cell method.

In MBS Plugin the objects must be strings or numbers. With plugin version 12.5, you can also use normal Xojo objects.

In OS X v10.4, NSTokenField trims whitespace around tokens but it does not trim whitespace in OS X versions 10.5.0 and 10.5.1. In OS X v10.5.2, you get whitespace-trimming behavior by either linking against the v10.4 binary or linking against the v10.5 binary and not implementing the `representedObjectForEditingString` event. If you do not want the whitespace-trimming behavior, link against the v10.5 binary and implement this method, returning the editing string if you have no represented object.

Please note: Due to the way the plugin is implemented the `Action`, `textShouldEndEditing` and `textShouldBeginEditing` events do nothing, so please use `tokenFieldAction`, `tokenFieldTextShouldEndEditing` and `tokenFieldTextShouldBeginEditing`.

Subclass of the `NSTextFieldMBS` class.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 15.1pr3](#)
- [MBS Real Studio Plugins, version 12.5pr1](#)

7.3.2 Methods

7.3.3 Constructor

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Creates a new control with size 100/100 and position 0/0

Example:

```
dim t as new NSTokenFieldMBS
```

Notes: On success the `handle` property is not zero.

See also:

- 7.3.4 Constructor(Handle as Integer) 186
- 7.3.5 Constructor(left as Double, top as Double, width as Double, height as Double) 186

7.3.4 Constructor(Handle as Integer)

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSControl handle.

Example:

```
dim t as new NSTextFieldMBS(0, 0, 100, 100)
dim v as new NSTextFieldMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSControl and the plugin retains this handle.

See also:

- 7.3.3 Constructor 185
- 7.3.5 Constructor(left as Double, top as Double, width as Double, height as Double) 186

7.3.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Creates a new control with the given size and position.

Example:

```
dim x as new NSTextFieldMBS(0, 0, 100, 20)
```

Notes: On success the handle property is not zero.

See also:

- 7.3.3 Constructor 185
- 7.3.4 Constructor(Handle as Integer) 186

7.3.6 defaultCompletionDelay as Double

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Returns the default completion delay.

Notes: The default completion delay is 0.

7.3.7 defaultTokenizingCharacterSet as NSCharacterSetMBS

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Returns the default tokenizing character set.

Notes: The default tokenizing character set is ”,”.

7.3.8 objects as Variant()

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Queries list of represented objects.

7.3.9 setObject(objects() as Variant)

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: Sets current objects.

7.3.10 Properties

7.3.11 completionDelay as Double

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The completion delay.

Notes: (Read and Write computed property)

7.3.12 tokenizingCharacterSet as NSCharacterSetMBS

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The tokenizing character set.

Notes: (Read and Write computed property)

7.3.13 tokenStyle as Integer

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The token style.

Notes: (Read and Write computed property)

7.3.14 Events

7.3.15 completionsForSubstring(substring as string, tokenIndex as Integer, byref selectedIndex as Integer) as Variant()

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide an array of appropriate completions for the contents of the receiver.

Notes: substring: The partial string that is to be completed.

tokenIndex: The index of the token being edited.

selectedIndex: Optionally, you can return by-reference an index into the returned array that specifies which of the completions should be initially selected. If none are to be selected, return by reference -1.

Returns an array of strings that are possible completions.

If the delegate does not implement this method, no completions are provided.

7.3.16 displayStringForRepresentedObject(representedObject as Variant) as string

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a string to be displayed as a proxy for the given represented object.

Notes: representedObject: A represented object of the token field.

Returns the string to be used as a proxy for representedObject. If you return nil or do not implement this method, then representedObject is displayed as the string.

7.3.17 editingStringForRepresentedObject(representedObject as Variant) as string

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a string to be edited as a proxy for a represented object.

Notes: representedObject: A represented object of the token field.

Returns a string that's an editable proxy of the represented object, or nil if the token should not be editable.

7.3.18 hasMenuForRepresentedObject(representedObject as Variant) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to specify whether the given represented object provides a menu.

Notes: representedObject: A represented object of the token field.

Returns true if the represented object has a menu, false otherwise.

By default tokens in a token field have no menus.

7.3.19 menuForRepresentedObject(representedObject as Variant) as NSMenuMBS

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a menu for the specified represented object.

Notes: representedObject: A represented object of the token field.

Returns the menu associated with the represented object.

By default tokens in a token field do not return menus.

7.3.20 readFromPasteboard(pboard as NSPasteboardMBS) as Variant()

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to return an array of objects representing the data read from the specified pasteboard.

Notes: pboard: The pasteboard from which to read the represented objects.

Returns an array of represented objects created from the pasteboard data.

7.3.21 representedObjectForEditingString(editingString as string) as Variant

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a represented object for the given editing string.

Notes: editingString: The edited string representation of a represented object.

Returns a represented object that is displayed rather than the editing string.

Note: In OS X v10.4, NSTokenField trims whitespace around tokens but it does not trim whitespace in OS X versions 10.5.0 and 10.5.1. In OS X v10.5.2, you get whitespace-trimming behavior by either linking against the v10.4 binary or linking against the v10.5 binary and not implementing the this method. If you do not want the whitespace-trimming behavior, link against the v10.5 binary and implement this method, returning the editing string if you have no represented object.

7.3.22 shouldAddObjects(tokens() as Variant, index as Integer) as Variant()

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows to validate the tokens to be added to the receiver at a particular location.

Notes: tokens: An array of tokens to be inserted in the receiver at index.

index: The index of the receiver in which the array of tokens to be validated (tokens) will be inserted.

Returns an array of validated tokens.

The event can return the array unchanged or return a modified array of tokens. To reject the add completely, return an empty array. Returning nil causes an error.

7.3.23 styleForRepresentedObject(representedObject as Variant) as Integer

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Allows the delegate to return the token style for editing the specified represented object.

Notes: representedObject: A represented object of the token field.

Returns the style that should be used to display the representedObject. Possible values are shown in NSTokenStyle Values.

If the event implements this method and returns an NSTokenStyle that differs from the style set by setTokenStyle:, the value the event returns is preferred.

If you don't implement this method, the token field's tokenStyle is used.

7.3.24 tokenFieldAction

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: The control's action was triggered.

Notes: For a button if it was pressed.

7.3.25 tokenFieldTextShouldBeginEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing.

7.3.26 tokenFieldTextShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow text editing.

7.3.27 writeRepresentedObjects(objects() as Variant, pboard as NSPasteboardMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Sent so the delegate can write represented objects to the pasteboard corresponding to a given array of display strings.

Notes: objects: An array of represented objects associated with the token field.

pboard: The pasteboard to which to write the represented objects.

Return true if you writes the represented objects to the pasteboard, false otherwise. If false, the token field writes the display strings to the NSStringPboardType pasteboard.

7.3.28 Constants

Token Styles

Constant	Value	Description
<code>NSDefaultTokenStyle</code>	0	Style best used for keyword type tokens. Available in OS X v10.4 and later.
<code>NSPlainTextTokenStyle</code>	1	Style to use for data you want represented as plain-text and without any token background. Available in OS X v10.4 and later.
<code>NSRoundedTokenStyle</code>	2	Style best used for address type tokens. Available in OS X v10.4 and later.

7.4 Globals

7.4.1 CenterResizeAddWindowMBS(win as window)

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Adds a window to the list of center resizing windows.

Notes: Please call in open event of window.

7.4.2 CenterResizeInstallMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Installs the center resize function.

Notes: The plugin can provide for Xojo and Xojo the center based window resizing.

Call this method early in app.open to prepare everything.

Then register windows with CenterResizeAddWindowMBS in window open event and unregister with CenterResizeRemoveWindowMBS in window close event. This works for all windows you register.

7.4.3 CenterResizeRemoveWindowMBS(win as window)

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Removes a window from the list of center resizing windows.

Notes: Please call in Close event of window.

Chapter 8

Cocoa Collection View

8.1 control DesktopNSCollectionViewControlMBS

8.1.1 control DesktopNSCollectionViewControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The control to host a NSCollectionView within Xojo.

Notes: Requires macOS 10.11 or newer!

See also:

https://developer.apple.com/documentation/appkit/views_and_controls/collection_view

<https://developer.apple.com/documentation/appkit/nscollectionView>

<https://developer.apple.com/documentation/appkit/nscollectionViewdatasource>

<https://developer.apple.com/documentation/appkit/nscollectionViewdelegate>

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [News from the MBS Xojo Plugins Version 21.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.3](#)
- [Smooth scrolling list control with ContainerControls](#)
- [Video about MBS Xojo Plugins 21.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [NSCollectionView for Xojo](#)
- [MBS Xojo Plugins Version 21.0 News](#)

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.0](#)

Videos

- [MBS Xojo Plugins 21.1](#)

Xojo Developer Magazine

- [19.5, page 9: News](#)
- [19.3, page 10: News](#)

8.1.2 Methods

8.1.3 `performBatchUpdates(tag as variant)`

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Encapsulates multiple insert, delete, reload, and move operations into a single animated operation.

Notes: Use this method to make multiple changes to the collection view in one single animated operation. Normally, when you insert, delete, reload, or move items, the collection view animates each change separately. Making those same changes inside the updates block causes them to be animated at the same time. This method updates the current layout information as needed before and after performing the operations in the updates block.

The order in which you make calls to insert, delete, or otherwise modify the collection view is ignored. When executing your updates block, this method gathers information about the operations you requested without performing those operations. After it gathers that information, the method reorders the operations and performs all deletion operations first, followed by all insertion operations and then all move operations. (Reloading an item is treated as a delete operation followed by an insert operation at the same location.) As a result, the indexes you specify for each set of operations must reflect the changes made by any preceding operations.

You may call this method from inside your `performBatchUpdatesWork` or `performBatchUpdatesCompleted` events.

This method calls `performBatchUpdatesWork` event, where you can perform the work and later `performBatchUpdatesCompleted` event when finished.

8.1.4 Properties

8.1.5 ScrollView as NSScrollViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The scroll view used around the collection view.

Notes: (Read only property)

8.1.6 View as NSCollectionViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The reference to the NSCollectionView object.

Notes: (Read only property)

8.1.7 Events

8.1.8 acceptDrop(draggingInfo as NSDraggingInfoMBS, indexPath as NSIndexPathMBS, dropOperation as Integer) as Integer

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Incorporates the dropped content into the collection view.

Notes: draggingInfo: The information about the drag operation.

indexPath: The index path at which the drop occurred. Use this location as the insertion point for the content.

dropOperation: The type of drop operation to perform.

Return true if the drop operation should be accepted or false if it should be rejected.

The collection view calls this event when the user releases the mouse button while it is over a valid drop target. This event is called after the validateDrop method validates that dropping the content at the specified location is possible. You must implement this event to accept the dropped content and incorporate it into the collection view.

In your implementation, use the information in the draggingInfo parameter to retrieve the data, update your data source object, and insert the appropriate items into the collection view. The dropped data is stored in the draggingPasteboard property of the dragging information object.

If the animatesToDestination property of the dragging information is true, update the image and frame for each dragged item to its new location in the collection view.

8.1.9 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

8.1.10 cancelPrefetchingForItems(indexPaths() as NSIndexPathMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies your 'prefetchDataSource' that items at the specified 'indexPaths', for which the UICollectionView previously sent prefetchItems event, are no longer expected to be displayed.

Notes: This is a good opportunity to cancel any pending resource fetches you've initiated for the items, if possible and worthwhile. This is only sent for items that don't end up being displayed; UICollectionView doesn't cancel prefetching for items that it actually instantiates and shows.

8.1.11 canDragItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean indicating whether a drag operation involving the specified items can begin.

Notes: indexPaths: The index paths of the items about to be dragged.

event: The mouse-down event that began the drag operation.

Return true if the drag operation can begin or false if it cannot.

If you do not implement this event, the collection view assumes a return value of true.

8.1.12 didChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the delegate to approve the pending highlighting of the specified items.

Notes: indexPaths: The array of NSIndexPathMBS objects corresponding to the items being highlighted.

highlightState: The new highlight state for the items.

Returns the array of NSIndexPath objects corresponding to the items that you want to receive the specified highlight. If you do not want any items to receive the specified highlight state, return an empty set.

Use this method to approve or modify the set of items targeted to receive the specified highlight state. During interactive selection or dragging, the collection view calls this method when actions occur that would affect the highlight state of items. Your implementation of the method can return the proposed set of index paths as-is or modify the set and disallow the highlighting of some or all of the items. Removing items from the set suppresses the corresponding actions, such as selecting the item or showing its eligibility as a drop target.

If you do not implement this method, the collection view updates the highlight state for the items specified by the indexPaths parameter.

8.1.13 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

8.1.14 didDeselectItems(indexPaths() as NSIndexPathMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies the delegate object that one or more items were deselected.

Notes: indexPaths: The array of NSIndexPathMBS objects corresponding to the items that were deselected.

After the user successfully deselects one or more items, the collection view calls this method to let you know that the items are no longer selected. Use this method to respond to the selection change and to make any necessary adjustments to your content or the collection view.

This method is not called when you set the selection programmatically using the methods of the NSCollectionViewMBS class.

8.1.15 didEndDisplayingItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified item was removed from the collection view.

Notes: item: The item that was removed.

`indexPath`: The index path of the item.

The collection view calls this method after removing an item from its content. You can use this method to track the removal of items and perform related tasks.

8.1.16 `didEndDisplayingSupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified supplementary view was removed from the collection view.

Notes: `view`: The supplementary view that was removed.

`elementKind`: The type of the supplementary view. Layouts are responsible for defining the types of supplementary views they support.

`indexPath`: The index path associated with the supplementary view.

The collection view calls this method after removing a supplementary view from its content. You can use this method to track the removal of views and perform related tasks.

8.1.17 `didSelectItems(indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies the delegate object that one or more items were selected.

Notes: `collectionView`: The collection view notifying you of the selection change.

`indexPaths`: The array of `NSIndexPathMBS` objects corresponding to the items that are now selected.

After the user successfully selects one or more items, the collection view calls this method to let you know that the selection has been made. Use this method to respond to the selection change and to make any necessary adjustments to your content or the collection view.

This method is not called when you set the selection programmatically using the methods of the `NSCollectionViewMBS` class.

8.1.18 `draggingImageForItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS, byref dragImageOffset as NSPointMBS) as NSImageMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Creates and returns a drag image to represent the specified items during a drag.

Notes: `indexPaths`: The index paths of the items being dragged.

NSEvent: The mouse-down event that began the drag operation. You can use the mouse location when determining what value to return in the `dragImageOffset` parameter.

dragImageOffset: The offset value to use when positioning the image. On input, the point is `NSZeroPoint`, which centers the returned image under the mouse. Your method can return a different point that repositions the drag image by the specified offset values.

Return the image to use for the dragged items.

Your implementation of this event should create an appropriate image to use during the drag operation. The collection view places the center of your image at the current mouse location. Update the value in the `dragImageOffset` parameter to shift the position of your image by the specified amount.

If you do not implement this event, the collection view uses the drag image returned by the `draggingImageForItemsAtIndexPaths` method instead.

8.1.19 `draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, dragOperation as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies your delegate that a drag session ended.

Notes: `session`: The dragging session that ended.

`screenPoint`: The end point (in screen coordinates) for the drag operation.

`operation`: The operation that was performed. Use this value to determine how the operation ended. For example, for content that was dragged to the trash, the operation type would be `NSDragOperationDelete` (32).

You can use this method to perform tasks related to the ending of a drag session.

8.1.20 `draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies your delegate that a drag session is about to begin.

Notes: `session`: The dragging session that is about to begin.

`screenPoint`: The starting point (in screen coordinates) for the drag operation. (in Cocoa coordinate system with bottom = zero)

`indexPaths`: The index paths of the items being dragged.

You can use this method to modify the dragging session or to perform other tasks related to the beginning of a drag session.

8.1.21 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

8.1.22 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

8.1.23 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

8.1.24 insetForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as NSEdgeInsetsMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks you for the margins to apply to content in the specified section.

Notes: collectionViewLayout: The layout object requesting the information.

section: The index of the section whose margins are needed.

Return the margins to apply to items in the specified section.

Implement this method when you want to provide margins for sections in the flow layout. Your implemen-

tation can return the same margins for all sections or it can return different margins for different sections. You can also adjust the margins of each section dynamically each time you update the layout. If you do not implement this method, the margins are obtained from the properties of the flow layout object.

The insets you return reflect the spacing between the items and the header and footer views of the section. They also reflect the spacing at the edges of a single row or column. For more information about how insets are applied, see the description of the `sectionInset` property.

8.1.25 `ItemApplyLayoutAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Applies the specified layout attributes to the element.

Notes: In your custom elements, you can use this method to apply the specified attributes to your content. For example, if your element object is a view controller, you would override this method and use it to apply the attributes to the root view object. When using your element with a layout object that supports custom attributes, you would also use this method to apply those custom attributes.

8.1.26 `ItemDidTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Tells the element that the layout object of the collection view changed.

Notes: `oldLayout`: The collection view's previous layout object.

`newLayout`: The current layout object associated with the collection view.

The default implementation of this method does nothing. Subclasses can override it and use it to finalize any behaviors associated with the change in layouts.

In OS X 10.11, this method is never called.

8.1.27 `itemForRepresentedObject(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks your data source object to provide the item at the specified location in the collection view.

Notes: `indexPath`: The index path that specifies the location of the item. This index path contains both

the section index and the item index within that section.

Returns a configured item object. You must not return nil from this method.

All data source objects must implement this method. Your implementation is responsible for creating, configuring, and returning the appropriate item object based on the specified index path. You do this by calling the `makeItem()` method of the collection view to retrieve an empty item object of the appropriate type. After receiving the item object, update its properties with the data from your app's data structures and return it.

You do not need to set the frame of an item's view from this method. The collection view gets the item's location and other layout-related attributes from the layout object during a separate step.

8.1.28 `ItemPreferredLayoutAttributesFittingAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS) as NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks your element if it wants to modify any layout attributes before they are applied.

Notes: `layoutAttributes`: The attributes provided by the layout object. These attributes represent the values that the layout object intends to apply to the element.

Return the final attributes to apply to the element.

The default implementation of this method returns the same attributes that are in the `layoutAttributes` parameter. You can override this method in subclasses and use it to return a different set of attributes. If you override this method, call `super` first to give the system the opportunity to make changes, then modify the returned attributes.

8.1.29 `ItemPrepareForReuse(item as NSCollectionViewItemMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Performs any necessary cleanup to prepare the element for use again.

Notes: The recycling of content is an important technique for improving performance of a collection view. Instead of creating all views from scratch, the collection view recycles views and view controllers that move offscreen. When your app subsequently calls the `makeItem` or `makeSupplementaryViewOfKind` method, the collection view retrieves a recycled object from the appropriate storage, calls this method, and then returns the object to your app.

Implement this method when you need to delete old data or when you want to restore your recycled views

to a standard initial state prior to their reuse. For example, you might use this method to restore the size of a view to some standard size or reset the alpha to 1.0 to ensure that the view is fully opaque. Do not use this method to configure the view with new data. Restoring your views to a default state in this method simplifies the configuration code you must write in your data source object later.

8.1.30 ItemWillTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Tells the element that the layout object of the collection view is about to change.

Notes: oldLayout: The current layout object used by the collection view.

newLayout: The new layout object that is about to be used by the collection view.

The default implementation of this method does nothing. Subclasses can override it and use it to prepare for the change in layouts.

Special Considerations

In OS X 10.11, this method is never called.

8.1.31 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In older Xojo versions, this event is named EnableMenuItems.

8.1.32 minimumInteritemSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks you for the spacing between successive items of a single row or column.

Notes: collectionViewLayout: The layout object requesting the information.

section: The index of the section whose inter-item spacing is needed.

Return the minimum space (in points) to apply between successive items in a single row or column.

Implement this method when you want to provide custom inter-item spacing for sections in the flow layout.

Your implementation can return the same spacing for all sections or it can return different spacing for different sections. You can also adjust the inter-item spacing of each section dynamically each time you update the layout. If you do not implement this method, the inter-item spacing is obtained from the properties of the flow layout object.

For a vertically scrolling layout, this value represents the minimum spacing between items in the same row. For a horizontally scrolling layout, this value represents the minimum spacing between items in the same column. The layout object uses this spacing only to compute how many items can fit in a single row or column. The actual spacing may be increased after the number of items has been determined. For more information about how inter-item spacing is applied, see the description of the `minimumInteritemSpacing` property.

8.1.33 `minimumLineSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks you for the spacing between successive rows or columns of a section.

Notes: `collectionViewLayout`: The layout object requesting the information.

`section`: The index of the section whose line spacing is needed.

Returns the minimum space (in points) to apply between successive lines in a section.

Implement this method when you want to provide custom line spacing for sections in the flow layout. Your implementation can return the same line spacing for all sections or it can return different line spacing for different sections. You can also adjust the line spacing of each section dynamically each time you update the layout. If you do not implement this method, the line spacing is obtained from the properties of the flow layout object.

For a vertically scrolling layout, this value represents the minimum spacing between successive rows. For a horizontally scrolling layout, this value represents the minimum spacing between successive columns. This spacing is not applied to the space between the header and the first line or between the last line and the footer. For more information about how line spacing is applied, see the description of the `minimumLineSpacing` property.

8.1.34 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to `x`, `y`.

Notes: The coordinates `x` and `y` are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

8.1.35 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

8.1.36 MouseUp(x As Integer, y As Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

8.1.37 namesOfPromisedFilesDroppedAtDestination(dropURL as String, indexPaths() as NSIndexPathMBS) as String()

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the names of the promised files that you created for a drag operation.

Notes: dropURL: The URL at which to create the promised files.

indexPaths: The index paths of the dragged items.

Return an array of strings containing the filenames you created, or intend to create, at the specified URL.

At the start of a drag operation, your app must provide the data that constitutes the items being dragged. If you specify a file promise, instead of the data itself, use this method to specify the names of the files you promised. If the files already exist, move them to the directory specified by the dropURL parameter. If you must create the files first, use this method to specify the names of the files you intend to provide and begin

creating those files asynchronously on a background thread.

The filenames you return are made available from the `namesOfPromisedFilesDroppedAtDestination` method of the `NSDraggingInfo` object passed around during the drag operation.

8.1.38 `numberOfItemsInSection(section as Integer) as Integer`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks your data source object to provide the number of items in the specified section.

Notes: All data source objects must implement this event. Your implementation should quickly return the number of items in the specified section.

Make sure the number of items you return is accurate. The `itemForRepresentedObject` event must be able to provide a visual representation for each item in the section.

8.1.39 `numberOfSections as Integer`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks your data source object to provide the total number of sections.

Notes: Implement this method when the organization of your data requires more than one section. If you do not implement this method, the collection view creates only one section.

8.1.40 `pasteboardWriterForItem(indexPath as NSIndexPathMBS) as NSPasteboardItemMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Provides the pasteboard writer for the item at the specified index path.

Notes: `collectionView:` The collection view making the request.

`indexPath:` The index path of the item requiring a pasteboard writer.

Returns the pasteboard writer object to use for managing the item data. Return `nil` to prevent the collection view from dragging the item.

You must implement this method or the `collectionView:writeItemsAtIndexPaths` method to support drag operations. The collection view calls this method in preference over the `writeItemsAtIndexPaths` method if both are implemented. If your app supports multi-image drag and drop, you must implement this method.

The collection view calls this method for each item involved in the drag operation after it has determined

that a drag should begin but before the drag operation has started. Your implementation of this method should create and return the pasteboard writer—an object conforming to the `NSPasteboardWriting` protocol—to use for providing the item’s data. Using the object you provide, the collection view creates an `NSDraggingItem` object for you and configures its `draggingFrame` and `imageComponents` properties for you using information from the item at the specified index path.

If you implement this method, the collection view does not call the `draggingImageForItemsAtIndexPaths` of your delegate or the `draggingImageForItemsAtIndexPaths` method of `NSCollectionView`.

8.1.41 `performBatchUpdatesCompleted(tag as variant, finished as boolean)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: A event to execute when the changes made in the `performBatchUpdatesWork` events have finished animating.

Notes: `finished`: A Boolean value indicating whether the animations completed successfully. The value of this parameter is true if the animations ran to completion or false if they were interrupted.

8.1.42 `performBatchUpdatesWork(tag as variant)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event that performs the needed inset, delete, reload, or move operations.

8.1.43 `prefetchItems(indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies your `'prefetchDataSource'` that items at the specified `'indexPaths'` are likely to be instantiated and displayed soon.

Notes: The `CollectionView` sends this as early as possible, to give your `'prefetchDataSource'` the opportunity to begin acquiring any resources such as thumbnail images or metadata that you might need in order to prepare the item’s content for display. The given `'indexPaths'` are sorted by the order in which they are likely to be needed.

8.1.44 `referenceSizeForFooterInSection(layout as NSCollectionViewLayoutMBS, section as Integer) as NSSizeMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks you for the size of the footer view in the specified section.

Notes: `collectionViewLayout`: The layout object requesting the information.

section: The index of the section whose footer size is requested.

The size of the footer. Return `NSZeroSize` if you do not want a footer added to the section.

If you implement this method, the flow layout object calls it to obtain the size of the footer in each section and uses that information to set the size of the corresponding views. If you do not implement this method, the footer size is obtained from the properties of the flow layout object.

The flow layout object uses only one of the returned size values. For a vertically scrolling layout, the layout object uses the height value. For a horizontally scrolling layout, the layout object uses the width value. The other value is sized appropriately to match the opposing dimension of the collection view itself. Set the size of the footer to 0 to prevent it from being displayed.

8.1.45 `referenceSizeForHeaderInSection(layout as NSCollectionViewLayoutMBS, section as Integer) as NSSizeMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks you for the size of the header view in the specified section.

Notes: `collectionViewLayout`: The layout object requesting the information.

`section`: The index of the section whose header size is requested.

Returns the size of the header. Return `NSZeroSize` if you do not want a header added to the section.

If you implement this method, the flow layout object calls it to obtain the size of the header in each section and uses that information to set the size of the corresponding views. If you do not implement this method, the header size is obtained from the properties of the flow layout object.

The flow layout object uses only one of the returned size values. For a vertically scrolling layout, the layout object uses the height value. For a horizontally scrolling layout, the layout object uses the width value. The other value is sized appropriately to match the opposing dimension of the collection view itself. Set the size of the header to 0 to prevent it from being displayed.

8.1.46 `ScaleFactorChanged(NewFactor as double)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

8.1.47 `shouldChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer) as NSIndexPathMBS()`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the delegate to approve the pending highlighting of the specified items.

Notes: `indexPaths`: The set of NSIndexPathMBS objects corresponding to the items being highlighted.
`highlightState`: The new highlight state for the items.

Return the set of NSIndexPath objects corresponding to the items that you want to receive the specified highlight. If you do not want any items to receive the specified highlight state, return an empty set.

Use this method to approve or modify the set of items targeted to receive the specified highlight state. During interactive selection or dragging, the collection view calls this method when actions occur that would affect the highlight state of items. Your implementation of the method can return the proposed set of index paths as-is or modify the set and disallow the highlighting of some or all of the items. Removing items from the set suppresses the corresponding actions, such as selecting the item or showing its eligibility as a drop target.

If you do not implement this method, the collection view updates the highlight state for the items specified by the `indexPaths` parameter.

8.1.48 `shouldDeselectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the delegate object to approve the pending deselection of items.

Notes: `indexPaths`: The array of NSIndexPathMBS objects corresponding to the items deselected by the user.

Return the array of NSIndexPathMBS objects corresponding to the items that you want to be deselected. If you do not want any items deselected, return an empty set.

Use this method to approve or modify the items that the user tries to deselect. During interactive selection, the collection view calls this method whenever the user deselects items. Your implementation of the method can return the proposed set of index paths as-is or modify the set before returning it. You might modify the set to disallow the deselection of specific items.

This method is not called when you set the selection programmatically using the methods of the NSCollectionViewMBS class. If you do not implement this method, the collection view deselects the items specified by the `indexPaths` parameter.

8.1.49 `shouldSelectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the delegate to approve the pending selection of items.

Notes: `indexPaths`: The array of `NSIndexPathMBS` objects corresponding to the items selected by the user.

Returns the array of `NSIndexPathMBS` objects corresponding to the items that you want to be selected. If you do not want any items selected, return an empty array.

Use this method to approve or modify the items that the user tries to select. During interactive selection, the collection view calls this method whenever the user selects new items. Your implementation of the method can return the proposed set of index paths as-is or modify the set before returning it. You might modify the set to disallow the selection of specific items or specific combinations of items.

This method is not called when you set the selection programmatically using the methods of the `NSCollectionViewMBS` class. If you do not implement this method, the collection view selects the items specified by the `indexPaths` parameter.

8.1.50 `sizeForItemAtIndexPath(layout as NSCollectionViewLayoutMBS, indexPath as NSIndexPathMBS) as NSSizeMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks you for the size of the specified item.

Notes: `collectionViewLayout`: The layout object requesting the information.

`indexPath`: The index path of the item.

Return the size of the item. The width and height values must both be greater than 0. Items must also not exceed the available space in the collection view.

Implement this method when you want to provide the size of items in the flow layout. Your implementation can return the same size for all items or it can return different sizes for items. You can also adjust the size of items dynamically each time you update the layout. If you do not implement this method, the size of items is obtained from the properties of the flow layout object.

The size value you return from this method must allow the item to be displayed fully by the collection view. In the scrolling direction, items can be larger than the collection view because the remaining content can always be scrolled into view, but in the nonscrolling directions, items should always be smaller than the collection view itself. For example, the width of an item in a vertically scrolling collection view must not exceed the width of the collection view minus any section insets. The flow layout does not crop an item, it bounds to make it fit into the available space.

8.1.51 transitionLayout(fromLayout as NSCollectionViewLayoutMBS, toLayout as NSCollectionViewLayoutMBS) as NSCollectionViewTransitionLayoutMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the transition layout object to use when performing an animated change between different layouts.

Notes: fromLayout: The current layout object of the collection view. This is the starting point for the transition.

toLayout: The new layout for the collection view.

Return the collection view transition layout object to use to perform the transition.

When changing layouts for a collection view, you can use this method to customize the transition layout object used to make the change. A transition layout object lets you customize the behavior of collection view elements when transitioning from one layout to the next. Normally, transitioning between layouts causes the assorted items and views to animate from their current locations directly to their new locations. By returning a custom transition object, you could have those elements follow a nonlinear path, use a different timing algorithm, or move items in response to touch events.

If you do not implement this method in your delegate object, the collection view uses a standard `UICollectionViewTransitionLayout` object for the transition.

Special Considerations

In OS X 10.11, this method is never called by the collection view.

8.1.52 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks your delegate to update the dragging items during a drag operation.

Notes: draggingInfo: The current information for the drag operation. Use this object to iterate over the dragging items.

You can use this method to update the current drag items while a drag is in progress. Updating the drag items is optional, but you might use this method to change the image for an item. For example, you might change the image when the mouse hovers over a particular part of the collection view. Use the `enumerateDraggingItems` method of the `draggingInfo` parameter to iterate over the drag items and update them as appropriate.

8.1.53 `validateDrop(draggingInfo as NSDraggingInfoMBS, byref proposedIndexPath as NSIndexPathMBS, byref dropOperation as Integer) as Integer`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Validates whether a drop operation is possible at the specified location.

Notes: `draggingInfo`: The information about the drag operation.

`proposedIndexPath`: The index path at which the drop would occur. This parameter is passed by-reference and can be modified to change the proposed index path.

`proposedDropOperation`: The type of drop operation being proposed. This parameter is passed by-reference and can be modified to change the drop operation type.

Returns a value that indicates which dragging operation to perform. Return `NSDragOperationNone` to disallow a drop at the proposed location.

Although implementation of this method is optional, you must implement it to support drops onto the associated collection view. You must also call the collection view's `registerForDraggedTypes` method to register the types of drops it supports. If you do not perform both of these actions, the collection view does not accept drops.

When an interactive drag operation occurs, the collection view calls this method to determine whether the current mouse location is a valid place to drop the content. This method may be called many times during the course of the drag operation. Your implementation should look at the proposed location and return a constant that reflects how the drop would be handled.

While validating the drop location, you can suggest a better drop location by updating the values in the `proposedIndexPath` and `proposedDropOperation` parameters. For example, you might suggest dropping the content before the specified item instead of on it. The collection view sets the `proposedDropOperation` parameter to `DropOn` when the mouse is closer to the middle of an item than to its edges; otherwise, it sets the parameter to `DropBefore`.

8.1.54 `viewForSupplementaryElement(kind as String, indexPath as NSIndexPathMBS) as NSViewMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks your data source object to provide the supplementary view at the specified location in a section of the collection view.

Notes: `kind`: The kind of supplementary view to provide. The value of this string is defined by the current layout object associated with the collection view. Layouts may define additional views to add visual content that is unrelated to specific items.

`indexPath`: The index path that identifies the section in which to place the supplementary view.

Returns a configured view object. You must not return `nil` from this method.

Implement this method if the collection view,Äôs layout object supports supplementary views. Your implementation is responsible for creating, configuring, and returning an appropriate view. You do this by calling the `makeSupplementaryViewOfKind` method of the collection view to retrieve an unconfigured view of the appropriate type. After receiving the view, update its properties and content using your app,Äôs data structures and return it.

You do not need to set the location of supplementary views inside the collection view,Äôs bounds. The collection view gets the view,Äôs location and other layout-related attributes from the layout object during a separate step.

8.1.55 `willDisplayItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified item is about to be displayed by the collection view.

Notes: `item`: The item being added.

`indexPath`: The index path of the item.

The collection view calls this method before adding new items to its content. You can use this method to track the addition of items and perform related tasks.

8.1.56 `willDisplaySupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified supplementary view is about to be displayed by the collection view.

Notes: `view`: The supplementary view being added.

`elementKind`: The type of the supplementary view. Layouts are responsible for defining the types of supplementary views they support.

`indexPath`: The index path associated with the supplementary view.

The collection view calls this method before adding new supplementary views to its content. You can use this method to track the addition of those views and perform related tasks.

8.1.57 `willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)`

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

8.1.58 writeItems(indexPaths() as NSIndexPathMBS, Pasteboard as NSPasteboardMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Places the data for the drag operation on the pasteboard.

Notes: indexPaths: The index paths of the items being dragged.

pasteboard: The pasteboard on which to place the drag data.

Return true if the drag operation can continue or no if you want to refuse the drag.

The collection view calls this method after it has determined that a drag should begin but before the drag operation has started. Your implementation of this method should do the following:

- Declare the pasteboard types you support using the declareTypes method of the provided pasteboard object.
- Write data to the pasteboard for each type you declare.
- Return true from this event.

8.1.59 Constants

Drop Operation

Constant	Value	Description
DropBefore	1	The drop occurs before the collection view item to which the item was dragged.
DropOn	0	The drop occurs at the collection view item to which the item was dragged.

8.2 control NSCollectionViewControlMBS

8.2.1 control NSCollectionViewControlMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The control to host a NSCollectionView within Xojo.

Notes: Requires macOS 10.11 or newer!

See also:

https://developer.apple.com/documentation/appkit/views_and_controls/collection_view

<https://developer.apple.com/documentation/appkit/nscollectionView>

<https://developer.apple.com/documentation/appkit/nscollectionViewdatasource>

<https://developer.apple.com/documentation/appkit/nscollectionViewdelegate>

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [News from the MBS Xojo Plugins Version 21.3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.3](#)
- [Smooth scrolling list control with ContainerControls](#)
- [Video about MBS Xojo Plugins 21.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [NSCollectionView for Xojo](#)
- [MBS Xojo Plugins Version 21.0 News](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.0](#)

Videos

- [MBS Xojo Plugins 21.1](#)

Xojo Developer Magazine

- [19.5, page 9: News](#)
- [19.3, page 10: News](#)

8.2.2 Methods

8.2.3 performBatchUpdates(tag as variant)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Encapsulates multiple insert, delete, reload, and move operations into a single animated operation.

Notes: Use this method to make multiple changes to the collection view in one single animated operation. Normally, when you insert, delete, reload, or move items, the collection view animates each change separately. Making those same changes inside the updates block causes them to be animated at the same time. This method updates the current layout information as needed before and after performing the operations in the updates block.

The order in which you make calls to insert, delete, or otherwise modify the collection view is ignored. When executing your updates block, this method gathers information about the operations you requested without performing those operations. After it gathers that information, the method reorders the operations and performs all deletion operations first, followed by all insertion operations and then all move operations. (Reloading an item is treated as a delete operation followed by an insert operation at the same location.) As a result, the indexes you specify for each set of operations must reflect the changes made by any preceding operations.

You may call this method from inside your performBatchUpdatesWork or performBatchUpdatesCompleted events.

This method calls performBatchUpdatesWork event, where you can perform the work and later performBatchUpdatesCompleted event when finished.

8.2.4 Properties

8.2.5 ScrollView as NSScrollViewMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The scroll view used around the collection view.

Notes: (Read only property)

8.2.6 View as NSCollectionViewMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The reference to the NSCollectionView object.

Notes: (Read only property)

8.2.7 Events

8.2.8 `acceptDrop(draggingInfo as NSDraggingInfoMBS, indexPath as NSIndexPathMBS, dropOperation as Integer) as Integer`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Incorporates the dropped content into the collection view.

Notes: `draggingInfo`: The information about the drag operation.

`indexPath`: The index path at which the drop occurred. Use this location as the insertion point for the content.

`dropOperation`: The type of drop operation to perform.

Return true if the drop operation should be accepted or false if it should be rejected.

The collection view calls this event when the user releases the mouse button while it is over a valid drop target. This event is called after the `validateDrop` method validates that dropping the content at the specified location is possible. You must implement this event to accept the dropped content and incorporate it into the collection view.

In your implementation, use the information in the `draggingInfo` parameter to retrieve the data, update your data source object, and insert the appropriate items into the collection view. The dropped data is stored in the `draggingPasteboard` property of the dragging information object.

If the `animatesToDestination` property of the dragging information is true, update the image and frame for each dragged item to its new location in the collection view.

8.2.9 BoundsChanged

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

8.2.10 `cancelPrefetchingForItems(indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies your 'prefetchDataSource' that items at the specified 'indexPaths', for which the Col-

collectionView previously sent prefetchItems event, are no longer expected to be displayed.

Notes: This is a good opportunity to cancel any pending resource fetches you've initiated for the items, if possible and worthwhile. This is only sent for items that don't end up being displayed; collectionView doesn't cancel prefetching for items that it actually instantiates and shows.

8.2.11 canDragItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS) as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Returns a Boolean indicating whether a drag operation involving the specified items can begin.

Notes: indexPaths: The index paths of the items about to be dragged.

event: The mouse-down event that began the drag operation.

Return true if the drag operation can begin or false if it cannot.

If you do not implement this event, the collection view assumes a return value of true.

8.2.12 Close

Plugin Version: 21.0, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

8.2.13 ConstructContextMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

8.2.14 ContextualMenuItemAction(hitItem as MenuItem) as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or

false to give others a chance.

8.2.15 didChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks the delegate to approve the pending highlighting of the specified items.

Notes: indexPaths: The array of NSIndexPathMBS objects corresponding to the items being highlighted.
highlightState: The new highlight state for the items.

Returns the array of NSIndexPath objects corresponding to the items that you want to receive the specified highlight. If you do not want any items to receive the specified highlight state, return an empty set.

Use this method to approve or modify the set of items targeted to receive the specified highlight state. During interactive selection or dragging, the collection view calls this method when actions occur that would affect the highlight state of items. Your implementation of the method can return the proposed set of index paths as-is or modify the set and disallow the highlighting of some or all of the items. Removing items from the set suppresses the corresponding actions, such as selecting the item or showing its eligibility as a drop target.

If you do not implement this method, the collection view updates the highlight state for the items specified by the indexPaths parameter.

8.2.16 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

8.2.17 didDeselectItems(indexPaths() as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies the delegate object that one or more items were deselected.

Notes: indexPaths: The array of NSIndexPathMBS objects corresponding to the items that were deselected.

After the user successfully deselects one or more items, the collection view calls this method to let you know that the items are no longer selected. Use this method to respond to the selection change and to make any

necessary adjustments to your content or the collection view.

This method is not called when you set the selection programmatically using the methods of the `NSCollectionViewMBS` class.

8.2.18 `didEndDisplayingItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified item was removed from the collection view.

Notes: `item`: The item that was removed.

`indexPath`: The index path of the item.

The collection view calls this method after removing an item from its content. You can use this method to track the removal of items and perform related tasks.

8.2.19 `didEndDisplayingSupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified supplementary view was removed from the collection view.

Notes: `view`: The supplementary view that was removed.

`elementKind`: The type of the supplementary view. Layouts are responsible for defining the types of supplementary views they support.

`indexPath`: The index path associated with the supplementary view.

The collection view calls this method after removing a supplementary view from its content. You can use this method to track the removal of views and perform related tasks.

8.2.20 `didSelectItems(indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies the delegate object that one or more items were selected.

Notes: `collectionView`: The collection view notifying you of the selection change.

`indexPaths`: The array of `NSIndexPathMBS` objects corresponding to the items that are now selected.

After the user successfully selects one or more items, the collection view calls this method to let you know that the selection has been made. Use this method to respond to the selection change and to make any

necessary adjustments to your content or the collection view.

This method is not called when you set the selection programmatically using the methods of the `NSCollectionViewMBS` class.

8.2.21 `draggingImageForItems(indexPaths() as NSIndexPathMBS, NSEvent as NSEventMBS, byref dragImageOffset as NSPointMBS) as NSImageMBS`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Creates and returns a drag image to represent the specified items during a drag.

Notes: `indexPaths`: The index paths of the items being dragged.

`NSEvent`: The mouse-down event that began the drag operation. You can use the mouse location when determining what value to return in the `dragImageOffset` parameter.

`dragImageOffset`: The offset value to use when positioning the image. On input, the point is `NSZeroPoint`, which centers the returned image under the mouse. Your method can return a different point that repositions the drag image by the specified offset values.

Return the image to use for the dragged items.

Your implementation of this event should create an appropriate image to use during the drag operation. The collection view places the center of your image at the current mouse location. Update the value in the `dragImageOffset` parameter to shift the position of your image by the specified amount.

If you do not implement this event, the collection view uses the drag image returned by the `draggingImageForItemsAtIndexPaths` method instead.

8.2.22 `draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, dragOperation as Integer)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies your delegate that a drag session ended.

Notes: `session`: The dragging session that ended.

`screenPoint`: The end point (in screen coordinates) for the drag operation.

`operation`: The operation that was performed. Use this value to determine how the operation ended. For example, for content that was dragged to the trash, the operation type would be `NSDragOperationDelete` (32).

You can use this method to perform tasks related to the ending of a drag session.

8.2.23 `draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies your delegate that a drag session is about to begin.

Notes: `session`: The dragging session that is about to begin.

`screenPoint`: The starting point (in screen coordinates) for the drag operation. (in Cocoa coordinate system with bottom = zero)

`indexPaths`: The index paths of the items being dragged.

You can use this method to modify the dragging session or to perform other tasks related to the beginning of a drag session.

8.2.24 `EnableMenuItems`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named `MenuBarSelected`.

8.2.25 `FrameChanged`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

8.2.26 `GotFocus`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named `FocusReceived`.

Notes:

This only fires if the control itself got focus and not a sub control.

8.2.27 insetForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as NSEdgeInsetsMBS

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Asks you for the margins to apply to content in the specified section.

Notes: collectionViewLayout: The layout object requesting the information.

section: The index of the section whose margins are needed.

Return the margins to apply to items in the specified section.

Implement this method when you want to provide margins for sections in the flow layout. Your implementation can return the same margins for all sections or it can return different margins for different sections. You can also adjust the margins of each section dynamically each time you update the layout. If you do not implement this method, the margins are obtained from the properties of the flow layout object.

The insets you return reflect the spacing between the items and the header and footer views of the section. They also reflect the spacing at the edges of a single row or column. For more information about how insets are applied, see the description of the sectionInset property.

8.2.28 ItemApplyLayoutAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Applies the specified layout attributes to the element.

Notes: In your custom elements, you can use this method to apply the specified attributes to your content. For example, if your element object is a view controller, you would override this method and use it to apply the attributes to the root view object. When using your element with a layout object that supports custom attributes, you would also use this method to apply those custom attributes.

8.2.29 ItemDidTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Tells the element that the layout object of the collection view changed.

Notes: oldLayout: The collection view,Â’s previous layout object.

newLayout: The current layout object associated with the collection view.

The default implementation of this method does nothing. Subclasses can override it and use it to finalize any behaviors associated with the change in layouts.

In OS X 10.11, this method is never called.

8.2.30 `itemForRepresentedObject(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks your data source object to provide the item at the specified location in the collection view.

Notes: `indexPath`: The index path that specifies the location of the item. This index path contains both the section index and the item index within that section.

Returns a configured item object. You must not return nil from this method.

All data source objects must implement this method. Your implementation is responsible for creating, configuring, and returning the appropriate item object based on the specified index path. You do this by calling the `makeItem()` method of the collection view to retrieve an empty item object of the appropriate type. After receiving the item object, update its properties with the data from your app's data structures and return it.

You do not need to set the frame of an item's view from this method. The collection view gets the item's location and other layout-related attributes from the layout object during a separate step.

8.2.31 `ItemPreferredLayoutAttributesFittingAttributes(item as NSCollectionViewItemMBS, layoutAttributes as NSCollectionViewLayoutAttributesMBS) as NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks your element if it wants to modify any layout attributes before they are applied.

Notes: `layoutAttributes`: The attributes provided by the layout object. These attributes represent the values that the layout object intends to apply to the element.

Return the final attributes to apply to the element.

The default implementation of this method returns the same attributes that are in the `layoutAttributes` parameter. You can override this method in subclasses and use it to return a different set of attributes. If you override this method, call `super` first to give the system the opportunity to make changes, then modify the returned attributes.

8.2.32 ItemPrepareForReuse(item as NSCollectionViewItemMBS)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Performs any necessary cleanup to prepare the element for use again.

Notes: The recycling of content is an important technique for improving performance of a collection view. Instead of creating all views from scratch, the collection view recycles views and view controllers that move offscreen. When your app subsequently calls the `makeItem` or `makeSupplementaryViewOfKind` method, the collection view retrieves a recycled object from the appropriate storage, calls this method, and then returns the object to your app.

Implement this method when you need to delete old data or when you want to restore your recycled views to a standard initial state prior to their reuse. For example, you might use this method to restore the size of a view to some standard size or reset the alpha to 1.0 to ensure that the view is fully opaque. Do not use this method to configure the view with new data. Restoring your views to a default state in this method simplifies the configuration code you must write in your data source object later.

8.2.33 ItemWillTransition(item as NSCollectionViewItemMBS, oldLayout as NSCollectionViewLayoutMBS, newLayout as NSCollectionViewLayoutMBS)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Tells the element that the layout object of the collection view is about to change.

Notes: `oldLayout`: The current layout object used by the collection view.

`newLayout`: The new layout object that is about to be used by the collection view.

The default implementation of this method does nothing. Subclasses can override it and use it to prepare for the change in layouts.

Special Considerations

In OS X 10.11, this method is never called.

8.2.34 LostFocus

Plugin Version: 21.0, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named `FocusLost`.

Notes:

This only fires if the control itself lost focus and not a sub control.

8.2.35 `minimumInteritemSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double`

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Asks you for the spacing between successive items of a single row or column.

Notes: `collectionViewLayout`: The layout object requesting the information.

`section`: The index of the section whose inter-item spacing is needed.

Return the minimum space (in points) to apply between successive items in a single row or column.

Implement this method when you want to provide custom inter-item spacing for sections in the flow layout. Your implementation can return the same spacing for all sections or it can return different spacing for different sections. You can also adjust the inter-item spacing of each section dynamically each time you update the layout. If you do not implement this method, the inter-item spacing is obtained from the properties of the flow layout object.

For a vertically scrolling layout, this value represents the minimum spacing between items in the same row. For a horizontally scrolling layout, this value represents the minimum spacing between items in the same column. The layout object uses this spacing only to compute how many items can fit in a single row or column. The actual spacing may be increased after the number of items has been determined. For more information about how inter-item spacing is applied, see the description of the `minimumInteritemSpacing` property.

8.2.36 `minimumLineSpacingForSectionAtIndex(layout as NSCollectionViewLayoutMBS, section as Integer) as double`

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Asks you for the spacing between successive rows or columns of a section.

Notes: `collectionViewLayout`: The layout object requesting the information.

`section`: The index of the section whose line spacing is needed.

Returns the minimum space (in points) to apply between successive lines in a section.

Implement this method when you want to provide custom line spacing for sections in the flow layout. Your implementation can return the same line spacing for all sections or it can return different line spacing for different sections. You can also adjust the line spacing of each section dynamically each time you update the layout. If you do not implement this method, the line spacing is obtained from the properties of the flow layout object.

For a vertically scrolling layout, this value represents the minimum spacing between successive rows. For a horizontally scrolling layout, this value represents the minimum spacing between successive columns. This

spacing is not applied to the space between the header and the first line or between the last line and the footer. For more information about how line spacing is applied, see the description of the `minimumLineSpacing` property.

8.2.37 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to `x`, `y`.

Notes: The coordinates `x` and `y` are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return `True` if you are going to handle the `MouseDown`. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the `MouseDown` and `MouseUp` events.

If you return `False`, the system handles the `MouseDown` so the above event handlers do not get called.

8.2.38 `MouseDown(x as Integer, y as Integer)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to `x`, `y`.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

8.2.39 `MouseUp(x As Integer, y As Integer)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the `x` and `y` parameters to determine if the mouse button was released within the control's boundaries.

8.2.40 `namesOfPromisedFilesDroppedAtDestination(dropURL as String, indexPaths() as NSIndexPathMBS) as String()`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Returns the names of the promised files that you created for a drag operation.

Notes: dropURL: The URL at which to create the promised files.

indexPaths: The index paths of the dragged items.

Return an array of strings containing the filenames you created, or intend to create, at the specified URL.

At the start of a drag operation, your app must provide the data that constitutes the items being dragged. If you specify a file promise, instead of the data itself, use this method to specify the names of the files you promised. If the files already exist, move them to the directory specified by the dropURL parameter. If you must create the files first, use this method to specify the names of the files you intend to provide and begin creating those files asynchronously on a background thread.

The filenames you return are made available from the `namesOfPromisedFilesDroppedAtDestination` method of the `NSDraggingInfo` object passed around during the drag operation.

8.2.41 `numberOfItemsInSection(section as Integer)` as Integer

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks your data source object to provide the number of items in the specified section.

Notes: All data source objects must implement this event. Your implementation should quickly return the number of items in the specified section.

Make sure the number of items you return is accurate. The `itemForRepresentedObject` event must be able to provide a visual representation for each item in the section.

8.2.42 `numberOfSections` as Integer

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks your data source object to provide the total number of sections.

Notes: Implement this method when the organization of your data requires more than one section. If you do not implement this method, the collection view creates only one section.

8.2.43 `Open`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.

In Xojo version 2021r3 and newer this event is named `Opening`.

8.2.44 `pasteboardWriterForItem(indexPath as NSIndexPathMBS) as NSPasteboardItemMBS`

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Provides the pasteboard writer for the item at the specified index path.

Notes: `collectionView`: The collection view making the request.

`indexPath`: The index path of the item requiring a pasteboard writer.

Returns the pasteboard writer object to use for managing the item data. Return nil to prevent the collection view from dragging the item.

You must implement this method or the `collectionView:writeItemsAtIndexPaths` method to support drag operations. The collection view calls this method in preference over the `writeItemsAtIndexPaths` method if both are implemented. If your app supports multi-image drag and drop, you must implement this method.

The collection view calls this method for each item involved in the drag operation after it has determined that a drag should begin but before the drag operation has started. Your implementation of this method should create and return the pasteboard writer—an object conforming to the `NSPasteboardWriting` protocol—to use for providing the item’s data. Using the object you provide, the collection view creates an `NSDraggingItem` object for you and configures its `draggingFrame` and `imageComponents` properties for you using information from the item at the specified index path.

If you implement this method, the collection view does not call the `draggingImageForItemsAtIndexPaths` of your delegate or the `draggingImageForItemsAtIndexPaths` method of `NSCollectionView`.

8.2.45 `performBatchUpdatesCompleted(tag as variant, finished as boolean)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: A event to execute when the changes made in the `performBatchUpdatesWork` events have finished animating.

Notes: `finished`: A Boolean value indicating whether the animations completed successfully. The value of this parameter is true if the animations ran to completion or false if they were interrupted.

8.2.46 `performBatchUpdatesWork(tag as variant)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: The event that performs the needed inset, delete, reload, or move operations.

8.2.47 `prefetchItems(indexPaths())` as `NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies your 'prefetchDataSource' that items at the specified 'indexPaths' are likely to be instantiated and displayed soon.

Notes: The `CollectionView` sends this as early as possible, to give your 'prefetchDataSource' the opportunity to begin acquiring any resources such as thumbnail images or metadata that you might need in order to prepare the item's content for display. The given 'indexPaths' are sorted by the order in which they are likely to be needed.

8.2.48 `referenceSizeForFooterInSection(layout as NSIndexPathMBS, section as Integer)` as `NSSizeMBS`

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Asks you for the size of the footer view in the specified section.

Notes: `collectionViewLayout`: The layout object requesting the information.

`section`: The index of the section whose footer size is requested.

The size of the footer. Return `NSZeroSize` if you do not want a footer added to the section.

If you implement this method, the flow layout object calls it to obtain the size of the footer in each section and uses that information to set the size of the corresponding views. If you do not implement this method, the footer size is obtained from the properties of the flow layout object.

The flow layout object uses only one of the returned size values. For a vertically scrolling layout, the layout object uses the height value. For a horizontally scrolling layout, the layout object uses the width value. The other value is sized appropriately to match the opposing dimension of the collection view itself. Set the size of the footer to 0 to prevent it from being displayed.

8.2.49 `referenceSizeForHeaderInSection(layout as NSIndexPathMBS, section as Integer)` as `NSSizeMBS`

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Asks you for the size of the header view in the specified section.

Notes: `collectionViewLayout`: The layout object requesting the information.

`section`: The index of the section whose header size is requested.

Returns the size of the header. Return `NSZeroSize` if you do not want a header added to the section.

If you implement this method, the flow layout object calls it to obtain the size of the header in each section and uses that information to set the size of the corresponding views. If you do not implement this method, the header size is obtained from the properties of the flow layout object.

The flow layout object uses only one of the returned size values. For a vertically scrolling layout, the layout object uses the height value. For a horizontally scrolling layout, the layout object uses the width value. The other value is sized appropriately to match the opposing dimension of the collection view itself. Set the size of the header to 0 to prevent it from being displayed.

8.2.50 ScaleFactorChanged(NewFactor as double)

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

8.2.51 shouldChangeItems(indexPaths() as NSIndexPathMBS, highlightState as Integer) as NSIndexPathMBS()

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks the delegate to approve the pending highlighting of the specified items.

Notes: indexPaths: The set of NSIndexPathMBS objects corresponding to the items being highlighted.

highlightState: The new highlight state for the items.

Return the set of NSIndexPath objects corresponding to the items that you want to receive the specified highlight. If you do not want any items to receive the specified highlight state, return an empty set.

Use this method to approve or modify the set of items targeted to receive the specified highlight state. During interactive selection or dragging, the collection view calls this method when actions occur that would affect the highlight state of items. Your implementation of the method can return the proposed set of index paths as-is or modify the set and disallow the highlighting of some or all of the items. Removing items from the set suppresses the corresponding actions, such as selecting the item or showing its eligibility as a drop target.

If you do not implement this method, the collection view updates the highlight state for the items specified by the indexPaths parameter.

8.2.52 `shouldDeselectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks the delegate object to approve the pending deselection of items.

Notes: `indexPaths`: The array of `NSIndexPathMBS` objects corresponding to the items deselected by the user.

Return the array of `NSIndexPathMBS` objects corresponding to the items that you want to be deselected. If you do not want any items deselected, return an empty set.

Use this method to approve or modify the items that the user tries to deselect. During interactive selection, the collection view calls this method whenever the user deselects items. Your implementation of the method can return the proposed set of index paths as-is or modify the set before returning it. You might modify the set to disallow the deselection of specific items.

This method is not called when you set the selection programmatically using the methods of the `NSCollectionViewMBS` class. If you do not implement this method, the collection view deselects the items specified by the `indexPaths` parameter.

8.2.53 `shouldSelectItems(indexPaths() as NSIndexPathMBS) as NSIndexPathMBS()`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks the delegate to approve the pending selection of items.

Notes: `indexPaths`: The array of `NSIndexPathMBS` objects corresponding to the items selected by the user.

Returns the array of `NSIndexPathMBS` objects corresponding to the items that you want to be selected. If you do not want any items selected, return an empty array.

Use this method to approve or modify the items that the user tries to select. During interactive selection, the collection view calls this method whenever the user selects new items. Your implementation of the method can return the proposed set of index paths as-is or modify the set before returning it. You might modify the set to disallow the selection of specific items or specific combinations of items.

This method is not called when you set the selection programmatically using the methods of the `NSCollectionViewMBS` class. If you do not implement this method, the collection view selects the items specified by the `indexPaths` parameter.

8.2.54 sizeForItemAtIndexPath(layout as NSCollectionViewLayoutMBS, indexPath as NSIndexPathMBS) as NSSizeMBS

Plugin Version: 21.3, Platform: macOS, Targets: .

Function: Asks you for the size of the specified item.

Notes: collectionViewLayout: The layout object requesting the information.

indexPath: The index path of the item.

Return the size of the item. The width and height values must both be greater than 0. Items must also not exceed the available space in the collection view.

Implement this method when you want to provide the size of items in the flow layout. Your implementation can return the same size for all items or it can return different sizes for items. You can also adjust the size of items dynamically each time you update the layout. If you do not implement this method, the size of items is obtained from the properties of the flow layout object.

The size value you return from this method must allow the item to be displayed fully by the collection view. In the scrolling direction, items can be larger than the collection view because the remaining content can always be scrolled into view, but in the nonscrolling directions, items should always be smaller than the collection view itself. For example, the width of an item in a vertically scrolling collection view must not exceed the width of the collection view minus any section insets. The flow layout does not crop an item, it bounds to make it fit into the available space.

8.2.55 transitionLayout(fromLayout as NSCollectionViewLayoutMBS, toLayout as NSCollectionViewLayoutMBS) as NSCollectionViewTransitionLayoutMBS

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Returns the transition layout object to use when performing an animated change between different layouts.

Notes: fromLayout: The current layout object of the collection view. This is the starting point for the transition.

toLayout: The new layout for the collection view.

Return the collection view transition layout object to use to perform the transition.

When changing layouts for a collection view, you can use this method to customize the transition layout object used to make the change. A transition layout object lets you customize the behavior of collection view elements when transitioning from one layout to the next. Normally, transitioning between layouts causes the assorted items and views to animate from their current locations directly to their new locations. By returning a custom transition object, you could have those elements follow a nonlinear path, use a different

timing algorithm, or move items in response to touch events.

If you do not implement this method in your delegate object, the collection view uses a standard `UICollectionViewTransitionLayout` object for the transition.

Special Considerations

In OS X 10.11, this method is never called by the collection view.

8.2.56 `updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks your delegate to update the dragging items during a drag operation.

Notes: `draggingInfo`: The current information for the drag operation. Use this object to iterate over the dragging items.

You can use this method to update the current drag items while a drag is in progress. Updating the drag items is optional, but you might use this method to change the image for an item. For example, you might change the image when the mouse hovers over a particular part of the collection view. Use the `enumerateDraggingItems` method of the `draggingInfo` parameter to iterate over the drag items and update them as appropriate.

8.2.57 `validateDrop(draggingInfo as NSDraggingInfoMBS, byref proposedIndexPath as NSIndexPathMBS, byref dropOperation as Integer) as Integer`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Validates whether a drop operation is possible at the specified location.

Notes: `draggingInfo`: The information about the drag operation.

`proposedIndexPath`: The index path at which the drop would occur. This parameter is passed by-reference and can be modified to change the proposed index path.

`proposedDropOperation`: The type of drop operation being proposed. This parameter is passed by-reference and can be modified to change the drop operation type.

Returns a value that indicates which dragging operation to perform. Return `NSDragOperationNone` to disallow a drop at the proposed location.

Although implementation of this method is optional, you must implement it to support drops onto the associated collection view. You must also call the collection view's `registerForDraggedTypes` method to register the types of drops it supports. If you do not perform both of these actions, the collection view does not accept drops.

When an interactive drag operation occurs, the collection view calls this method to determine whether the current mouse location is a valid place to drop the content. This method may be called many times during the course of the drag operation. Your implementation should look at the proposed location and return a constant that reflects how the drop would be handled.

While validating the drop location, you can suggest a better drop location by updating the values in the `proposedDropIndexPath` and `proposedDropOperation` parameters. For example, you might suggest dropping the content before the specified item instead of on it. The collection view sets the `proposedDropOperation` parameter to `DropOn` when the mouse is closer to the middle of an item than to its edges; otherwise, it sets the parameter to `DropBefore`.

8.2.58 `viewForSupplementaryElement(kind as String, indexPath as NSIndexPathMBS) as NSViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Asks your data source object to provide the supplementary view at the specified location in a section of the collection view.

Notes: `kind`: The kind of supplementary view to provide. The value of this string is defined by the current layout object associated with the collection view. Layouts may define additional views to add visual content that is unrelated to specific items.

`indexPath`: The index path that identifies the section in which to place the supplementary view.

Returns a configured view object. You must not return `nil` from this method.

Implement this method if the collection view's layout object supports supplementary views. Your implementation is responsible for creating, configuring, and returning an appropriate view. You do this by calling the `makeSupplementaryViewOfKind` method of the collection view to retrieve an unconfigured view of the appropriate type. After receiving the view, update its properties and content using your app's data structures and return it.

You do not need to set the location of supplementary views inside the collection view's bounds. The collection view gets the view's location and other layout-related attributes from the layout object during a separate step.

8.2.59 `willDisplayItem(item as NSCollectionViewItemMBS, indexPath as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified item is about to be displayed by the collection view.

Notes: `item`: The item being added.

`indexPath`: The index path of the item.

The collection view calls this method before adding new items to its content. You can use this method to track the addition of items and perform related tasks.

8.2.60 `willDisplaySupplementaryView(view as NSViewMBS, elementKind as String, indexPath as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Notifies the delegate that the specified supplementary view is about to be displayed by the collection view.

Notes: `view`: The supplementary view being added.

`elementKind`: The type of the supplementary view. Layouts are responsible for defining the types of supplementary views they support.

`indexPath`: The index path associated with the supplementary view.

The collection view calls this method before adding new supplementary views to its content. You can use this method to track the addition of those views and perform related tasks.

8.2.61 `willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)`

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

8.2.62 `writeItems(indexPaths() as NSIndexPathMBS, Pasteboard as NSPasteboardMBS) as Boolean`

Plugin Version: 21.0, Platform: macOS, Targets: .

Function: Places the data for the drag operation on the pasteboard.

Notes: `indexPaths`: The index paths of the items being dragged.

`pasteboard`: The pasteboard on which to place the drag data.

Return true if the drag operation can continue or no if you want to refuse the drag.

The collection view calls this method after it has determined that a drag should begin but before the drag operation has started. Your implementation of this method should do the following:

- Declare the pasteboard types you support using the `declareTypes` method of the provided pasteboard object.
- Write data to the pasteboard for each type you declare.
- Return true from this event.

8.2.63 Constants

Drop Operation

Constant	Value	Description
<code>DropBefore</code>	1	The drop occurs before the collection view item to which the item was dragged.
<code>DropOn</code>	0	The drop occurs at the collection view item to which the item was dragged.

8.3 class `NSCollectionViewFlowLayoutInvalidationContextMBS`

8.3.1 class `NSCollectionViewFlowLayoutInvalidationContextMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An object that identifies the portions of a flow layout object that need to be updated.

Notes: Layout objects use invalidation contexts to optimize the layout process and avoid unnecessary work. You use this class to specify whether the `NSCollectionViewFlowLayout` object should fetch new size information from its delegate. You can also prevent the flow layout object from updating its layout information altogether.

When you want to invalidate your flow layout object, call the `NewInvalidationContext` method of your layout object and instantiate the resulting class. (The implementation of that method in `NSCollectionViewFlowLayout` returns this class.) After instantiating this class, set the properties to appropriate values and pass the object to the `invalidateLayoutWithContext` method of the layout object. Subclass of the `NSCollectionViewLayoutInvalidationContextMBS` class.

8.3.2 Methods

8.3.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.3.4 Properties

8.3.5 `invalidateFlowLayoutAttributes` as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the flow layout object should invalidate its current attributes.

Notes: Setting this property to `NO` tells the flow layout object to keep its existing layout information, effectively stopping the invalidation process. Typically, you set this property to `false` only if you subclass `NSCollectionViewFlowLayoutMBS` and update changed layout information directly.

The default value of this property is `true`, which causes the flow layout object to throw out its existing layout information and recompute it.

(Read and Write property)

8.3.6 invalidateFlowLayoutDelegateMetrics as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the flow layout object should fetch new size information from its delegate.

Notes: As part of the invalidation process, the flow layout object normally asks its delegate to provide size information for the items in the flow layout. This behavior is necessary when the size of the items can change because it ensures that the corresponding layout attributes are always updated. However, if you know that the size of items has not changed, you can set this property to false. Doing so causes the flow layout to use its existing size information rather than querying the delegate, which saves time.

The default value of this property is YES, which causes the flow layout object to query the delegate for new size information.

(Read and Write property)

8.4 class `NSCollectionViewFlowLayoutMBS`

8.4.1 class `NSCollectionViewFlowLayoutMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A layout that organizes items into a flexible and configurable arrangement.

Notes: In a flow layout, the first item is positioned in the top-left corner and other items are laid out either horizontally or vertically based on the scroll direction, which is configurable. Items may be the same size or different sizes, and you may use the flow layout object or the collection view's delegate object to specify the size of items and the spacing around them. The flow layout also lets you specify custom header and footer views for each section.

You can use an `NSCollectionViewFlowLayoutMBS` object as-is or subclass it to modify more aspects of the layout behavior. There are several ways to customize the basic layout behavior that do not require subclassing. For example, you can use a delegate object to change the size and spacing of items dynamically. Subclassing is appropriate for more advanced layout changes, such as adding supplementary views or decoration views, supporting custom layout attributes, or customizing the layout animations when inserting or deleting items.

see also

<https://developer.apple.com/documentation/appkit/nscollectionviewflowlayout>

Subclass of the `NSCollectionViewLayoutMBS` class.

Blog Entries

- [NSCollectionView for Xojo](#)

8.4.2 Methods

8.4.3 `collapseSectionAtIndex(sectionIndex as Integer)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Collapses the section with the given index.

8.4.4 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.4.5 ElementKindSectionFooter as String

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A supplementary view that acts as a footer for a given section.

8.4.6 ElementKindSectionHeader as String

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A supplementary view that acts as a header for a given section.

8.4.7 expandSectionAtIndex(sectionIndex as Integer)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Expands the section with the given index.

8.4.8 Properties

8.4.9 estimatedItemSize as NSSizeMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The estimated size of items in the collection view.

Notes: Providing an estimated item size lets the collection view defer some of the calculations needed to determine the size of its content, which can improve performance. Instead of explicitly computing the size of each item, the flow layout assumes that offscreen items have the estimated size. The estimated size is used only until an actual value is calculated. The default value of this property is zero size.

If the value of this property is not NSZeroSize, the flow layout uses the estimated size you specified. If all of your items actually have the same size, use the `itemSize` property to set their size and set this property to NSZeroSize.

(Read and Write property)

8.4.10 footerReferenceSize as NSSizeMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The default size to use for section footers.

Notes: The layout object uses only the value that is appropriate for the current scrolling direction. In other

words, the layout object uses only the height value when the content scrolls vertically, setting the width of the footer to the width of the collection view. Similarly, the layout object uses only the width value when the content scrolls horizontally, setting the footer,Äôs height to the height of the collection view. If the size value for the appropriate dimension is 0, the layout object omits the footer entirely.

The default value of this property is zero size.
(Read and Write property)

8.4.11 headerReferenceSize as NSSizeMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The default size to use for section headers.

Notes: The layout object uses only the value that is appropriate for the current scrolling direction. In other words, the layout object uses only the height value when the content scrolls vertically, setting the width of the header to the width of the collection view. Similarly, the layout object uses only the width value when the content scrolls horizontally, setting the header,Äôs height to the height of the collection view. If the size value for the appropriate dimension is 0, the layout object omits the header entirely.

The default value of this property is zero size.
(Read and Write property)

8.4.12 itemSize as NSSizeMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The default size to use for items.

Notes: This property contains the default size of items.

All items are set to the same size. This value applies only to items and not to supplementary views.
The default value of this property is (50.0, 50.0).
(Read and Write property)

8.4.13 minimumInteritemSpacing as Double

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function:

The minimum spacing (in points) to use between items in the same row or column.

Notes:

For a vertically scrolling layout, the value represents the minimum spacing between items in the same row. For a horizontally scrolling layout, the value represents the minimum spacing between items in the same column. The layout object uses this spacing only to compute how many items can fit in a single row or column. The actual spacing may be increased after the number of items has been determined.

The default value of this property is 10.0.
(Read and Write property)

8.4.14 minimumLineSpacing as Double

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The minimum spacing (in points) to use between rows or columns.

Notes: For a vertically scrolling layout, the value represents the minimum spacing between successive rows. For a horizontally scrolling layout, the value represents the minimum spacing between successive columns. This spacing is not applied to the space between the header view and the first line or between the last line and the footer view. Figure 1 shows how the line spacing is applied to rows of unevenly sized items, illustrating how the actual spacing between individual items may be greater than the minimum value.

The default value of this property is 10.0.
(Read and Write property)

8.4.15 scrollDirection as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The scroll direction of the layout.

Notes: The flow layout scrolls along one axis only, either horizontally or vertically. When the scroll direction is `ScrollDirectionVertical`, the width of the content never exceeds the width of the collection view itself but the height grows as needed to accommodate the current items. When the scroll direction is `ScrollDirectionHorizontal`, the height never exceeds the height of the collection view but the width grows as needed.

The default value of this property is `ScrollDirectionVertical`.
(Read and Write property)

8.4.16 sectionFootersPinToVisibleBounds as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Set to true to get footers that pin to the bottom while scrolling.

Notes: Enabling this feature may affect the parenting of header and footer views.
(Read and Write property)

8.4.17 `sectionHeadersPinToVisibleBounds` as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Set to true to get headers that pin to the top of the visible area.

Notes: Enabling this feature may affect the parenting of header and footer views.
(Read and Write property)

8.4.18 `sectionInset` as `NSEdgeInsetsMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The margins used to lay out content in a section.

Notes: Section insets reflect the spacing at the outer edges of the section. The margins affect the positioning of the header view, the minimum space on either side of each line of items, and the distance from the last line to the footer view. The margin insets do not affect the size of the header and footer views in the nonscrolling direction.

The default insets are all set to 0.

(Read and Write property)

8.4.19 `sectionAtIndexIsCollapsed(sectionIndex as Integer)` as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Whether the section is collapsed.

Notes: True if the specified section is currently collapsed; false if not, or if there is no such section. Defaults to false.

(Read and Write computed property)

8.4.20 Constants

Scroll Directions

Constant	Value	Description
ScrollDirectionHorizontal	1	The layout scrolls content horizontally.
ScrollDirectionVertical	0	The layout scrolls content vertically.

8.5 class `NSCollectionViewGridLayoutMBS`

8.5.1 class `NSCollectionViewGridLayoutMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A layout that displays a single section of items in a row and column grid..

Notes: The `NSCollectionViewGridLayoutMBS` object provides the same layout behavior offered by the `NSCollectionViewMBS` class prior to macOS 10.11, and you can use it in cases where you want to maintain the old appearance while still taking advantage of newer collection view features.

Configuring a Collection View to Use a Grid Layout

You can configure a collection view to use a grid layout object programmatically or at design time:

- At design time, set the `Layout` attribute of your collection view to `Grid`.

Create an `NSCollectionViewGridLayoutMBS` object programmatically and assign it to the collection view,Äôs `collectionViewLayout` property.

- A grid layout displays only items and does not display supplementary views or decoration views. Use the properties of this class to configure the number of rows and columns in the grid. You can also use these properties to configure the spacing between items and the minimum sizes.

Subclass of the `NSCollectionViewLayoutMBS` class.

8.5.2 Methods

8.5.3 `backgroundColors` as `NSColorMBS()`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The array of background colors to use when drawing the grid.

Notes: The `NSColorMBS` objects in this property are used to draw the grid,Äôs background. The appearance of the background depends on the value you specify:

- Specifying `nil` fills the background with the collection view,Äôs default background color.
- Specifying an empty array causes the collection view to draw no background color.
- Specifying an array with one color object fills the background with the specified color.

- Specifying an array with more than one color object causes the collection view to use the specified colors to create a checkerboard pattern. Each successive grid item is displayed with the next color in the array, cycling back to the beginning of the array when the last color is reached.

The default value of this property is nil.

8.5.4 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.5.5 setBackgroundColors(Colors() as NSColorMBS = nil)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Sets the array of background colors to use when drawing the grid.

Notes: The NSColorMBS objects in this property are used to draw the grid's background. The appearance of the background depends on the value you specify:

- Specifying nil fills the background with the collection view's default background color.
- Specifying an empty array causes the collection view to draw no background color.
- Specifying an array with one color object fills the background with the specified color.
- Specifying an array with more than one color object causes the collection view to use the specified colors to create a checkerboard pattern. Each successive grid item is displayed with the next color in the array, cycling back to the beginning of the array when the last color is reached.

The default value of this property is nil.

8.5.6 Properties

8.5.7 margins as NSEdgeInsetsMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The amount of empty space (in points) around the grid's content.

Notes: The default value of this property is zeros. Changing this property to a new value invalidates the layout.

(Read and Write property)

8.5.8 `maximumItemSize` as `NSSizeMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The largest allowable size for an item,Ãs view.

Notes: Use this property to limit the maximum size of items displayed in the grid. The default value of this property is (0.0, 0.0), which imposes no maximum size for items.

(Read and Write property)

8.5.9 `maximumNumberOfColumns` as `Integer`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The maximum number of columns to display in the collection view,Ãs visible area.

Notes: Use this value to specify the maximum number of columns that should be visible in the collection view at any given time. The grid layout object uses this value during layout to configure the position and spacing of items. The default value of this property is 0, which means that there is no maximum number of columns.

(Read and Write property)

8.5.10 `maximumNumberOfRows` as `Integer`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The maximum number of rows to display in the collection view,Ãs visible area.

Notes: Use this value to specify the maximum number of rows to display in the collection view at any given time. The grid layout object uses this value during layout to configure the position and spacing of items. The default value of this property is 0, which means that there is no maximum number of rows.

(Read and Write property)

8.5.11 `minimumInteritemSpacing` as `Double`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The minimum spacing (in points) to use between items in the same row or column.

Notes: For a vertically scrolling layout, the value represents the minimum spacing between items in the same row. For a horizontally scrolling layout, the value represents the minimum spacing between items in the same column. The layout object uses this spacing only to compute how many items can fit in a single row or column. The actual spacing may be increased after the number of items has been determined.

The default value of this property is 0.0.

(Read and Write property)

8.5.12 `minimumItemSize` as `NSSizeMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The smallest allowable size for an item,Ã’s view.

Notes: Use this property to ensure that items have a minimum size when displayed in the grid. The default value of this property is (0.0, 0.0), which imposes no minimum size for items.

(Read and Write property)

8.5.13 `minimumLineSpacing` as `Double`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The minimum spacing (in points) to use between rows or columns.

Notes: For a vertically scrolling layout, the value represents the minimum spacing between successive rows. For a horizontally scrolling layout, the value represents the minimum spacing between successive columns. This spacing is not applied to the space between the header view and the first line or between the last line and the footer view.

The default value of this property is 0.0.

(Read and Write property)

8.5.14 Constants

Highlight States

Constant	Value	Description
<code>HighlightAsDropTarget</code>	3	The drop target highlight state. This type of highlight is applied when the item is the target of a drop operation on the collection view. After the drop operation completes, the highlight state returns to <code>HighlightNone</code> .
<code>HighlightForDeselection</code>	2	The deselection highlight state. During interactive selection, this state is used to indicate that the item will become deselected when interactions end. After interactions end, the highlight state returns to <code>HighlightNone</code> .
<code>HighlightForSelection</code>	1	The selected highlight state. This type of highlight is applied when an item is selected. During interactive highlighting, this state is also applied to indicate that the item will become highlighted.
<code>HighlightNone</code>	0	No highlight state.

8.6 class `NSCollectionViewItemMBS`

8.6.1 class `NSCollectionViewItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The visual representation for a single data element in a collection view.

Notes: Item objects are view controllers, and you use their view hierarchies to display your content. The default implementation of this class supports the creation of a simple item that displays a single image or string. If the appearance or layout of your items is more sophisticated, you can subclass and configure the view hierarchy based on your needs.

see also

<https://developer.apple.com/documentation/appkit/nscollectionViewitem>
Subclass of the `NSViewControllerMBS` class.

8.6.2 Methods

8.6.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.6.4 copy as `NSCollectionViewItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates a copy of the item.

8.6.5 Properties

8.6.6 `collectionView` as `NSCollectionViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The collection view that owns the item.

Notes: Use this property as a convenient way to access the collection view that owns the item.
(Read only property)

8.6.7 highlightState as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The highlight state currently applied to the item.

Notes: The highlight state provides a visual indication of operations happening to items in the collection view. The highlight state normally toggles between the HighlightNone and HighlightForSelection states, but other states may be applied to indicate transient conditions. For example, the HighlightForDeselection state is applied during interactive selections when a currently selected item is about to be deselected.

(Read and Write property)

8.6.8 identifier as String

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The identifier.

Notes: (Read only property)

8.6.9 imageView as NSImageViewMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An image view outlet that you can use to display images.

Notes: This is a convenience property for accessing an image view in your item,Äôs view hierarchy.

(Read and Write property)

8.6.10 selected as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean indicating whether the item is currently selected.

Notes: The value of this property is true when the item is selected or false when it is not.

(Read and Write property)

8.6.11 textField as NSTextFieldMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A text field outlet that you can use to display a string.

Notes: This is a convenience property for accessing a text field in your item,Äôs view hierarchy.

(Read and Write property)

8.6.12 Constants

Highlight States

Constant	Value	Description
HighlightAsDropTarget	3	The drop target highlight state. This type of highlight is applied when the item is the target of a drop operation on the collection view. After the drop operation completes, the highlight state returns to HighlightNone.
HighlightForDeselection	2	The deselection highlight state. During interactive selection, this state is used to indicate that the item will become deselected when interactions end. After interactions end, the highlight state returns to HighlightNone.
HighlightForSelection	1	The selected highlight state. This type of highlight is applied when an item is selected. During interactive highlighting, this state is also applied to indicate that the item will become highlighted.
HighlightNone	0	No highlight state.

8.7 class NSCollectionViewLayoutAttributesMBS

8.7.1 class NSCollectionViewLayoutAttributesMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An object that contains layout-related attributes for an element in a collection view.

Notes: During the layout, the layout object creates instances of NSCollectionViewLayoutAttributesMBS for each element displayed in the collection view. The layout attributes describe the position of an element and other information such as its alpha and position on the z axis. The collection view later applies the layout attributes to the onscreen elements.

The only time you interact with layout attribute objects is when you implement a custom layout, and the interactions are straightforward. When asked for layout attributes for a specific element, your layout object uses the methods of this class to create an appropriate instance of the class based on the type of the requested element. It then configures the properties of the object and returns it to the requester.

8.7.2 Methods

8.7.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.7.4 copy as NSCollectionViewLayoutAttributesMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates a copy of the object.

8.7.5 ElementKindInterItemGapIndicator as String

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The element kind string assigned to the attributes object when it represents an inter-item gap.

8.7.6 `layoutAttributesForDecorationView(decorationViewKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a layout attributes object for a decoration view based on the specified information.

Notes: `decorationViewKind`: A string that identifies the type of the decoration view. Use this string to differentiate from among the decoration views in a given section. This parameter must contain a valid value. `indexPath`: The index path of the item. You can use this information to identify the item in your app's data structures.

Returns a new layout attributes object configured with the initial attributes for the decoration view.

Call this method when you need to create a layout attributes object for a decoration view in a collection view. Decoration views are a tertiary type of content that display visual adornments in your collection view interface. For example, decoration views might display custom backgrounds. This method uses the parameters to set the initial values of the `indexPath` and `representedElementKind` properties the returned object.

8.7.7 `layoutAttributesForInterItemGapBefore(indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a layout attributes object for an inter-item gap view at the specified index path.

Notes: `indexPath`: The index path at which to insert the gap view. The gap is placed after the item specified by the index path. This parameter must contain a valid value.

Returns a new layout attributes object configured with the initial attributes for the inter-item gap view.

Call this method when you need to create a layout attributes object for an inter-item gap view in a collection view. Gap views are used during drag and drop to indicate the area where content will drop. This method uses the parameters to set the initial values of the `indexPath` property of the returned object. The `representedElementKind` property is set to `ElementKindInterItemGapIndicator`.

8.7.8 `layoutAttributesForItem(indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a layout attributes object for the item at the specified index path.

Notes: `indexPath`: The index path of the item. You can use this information to identify the item in your app's data structures. This parameter must contain a valid value.

Returns a new layout attributes object containing the initial attributes for the item.

Call this method when you need to create a layout attributes object for an item in a collection view. Items are the main type of content presented by a collection view. Items are grouped into sections, although a collection view may have only one section. This method assigns the provided index path to the `indexPath` property of the returned object.

8.7.9 `layoutAttributesForSupplementaryView(elementKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a layout attributes object for a supplementary view based on the specified information.

Notes: `elementKind`: A string that identifies the type of the supplementary view. Use this string to differentiate from among the supplementary views in a given section. This parameter must contain a valid value.
`indexPath`: The index path of the item. You can use this information to identify the item in your app's data structures. This parameter must contain a valid value.

Returns a new layout attributes object configured with the initial attributes for the supplementary view.

Call this method when you need to create a layout attributes object for a supplementary view in a collection view. Supplementary views are a secondary type of content that display data related to a specific section. For example, header and footer views in a grid layout implemented using supplementary views. This method uses the parameters to set the initial values of the `indexPath` and `representedElementKind` properties the returned object.

8.7.10 Properties

8.7.11 `alpha` as Double

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The transparency of the element.

Notes: Possible values are between 0.0 (fully transparent) and 1.0 (fully opaque). The default value is 1.0. Transparent items continue to participate in hit testing for the collection view.
(Read and Write property)

8.7.12 frame as NSSizeMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The frame rectangle of the element.

Notes: The frame rectangle is measured in points and specified in the collection view,Ãs coordinate system. Setting the value of this property also updates the value in the size property.
(Read and Write property)

8.7.13 Hidden as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the element is hidden.

Notes: The default value of this property is false. As an optimization, the collection view might not create the corresponding view when the value of this property is true. Because there might not be a view, hidden elements do not participate in hit testing for the collection view.
(Read and Write property)

8.7.14 indexPath as NSIndexPathMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The index path of the element.

Notes: Use the index path to locate information about the item in your app,Ãs data structures. For supplementary and decoration views, you must also use the `representedElementKind` property to identify the element.
(Read and Write property)

8.7.15 representedElementCategory as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The type of the element.

Notes: Use this property to distinguish whether the layout attributes apply to an item, a supplementary view, a decoration view, or another type of element presented by the collection view.
(Read only property)

8.7.16 representedElementKind as String

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The identifier for specific elements of your collection view interface.

Notes: For supplementary and decoration views, you use this string to distinguish between views in a given section. You also use this string to identify the intended purpose of the view in your collection view interface. When the value of the `representedElementCategory` property is `ElementCategoryItem`, this property is nil. (Read only property)

8.7.17 size as NSSizeMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The size of the element.

Notes: Setting the value of this property also updates the value in the frame property. (Read and Write property)

8.7.18 zIndex as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The element's position on the z axis.

Notes: Use this property to specify the front-to-back ordering of items during layout. Items with higher index values appear on top of those with lower values. Items with the same value have an undetermined order.

The default value of this property is 0. (Read and Write property)

8.7.19 Constants

Element Categories

Constant	Value	Description
<code>ElementCategoryDecorationView</code>	2	The element is a decoration view. Decoration views represent visual elements that do not contain any data of their own.
<code>ElementCategoryInterItemGap</code>	3	The element is an inter-item gap. An inter-item gap element is a custom visual indicator that is displayed between items when dropping items into a collection view.
<code>ElementCategoryItem</code>	0	The element is an item. Items represent the main content of your collection view.
<code>ElementCategorySupplementaryView</code>	1	The element is a supplementary view. Use supplementary views for single views that contain some data but are associated with an entire section. For example, use them to specify header or footer views for a section.

8.8 class `NSCollectionViewLayoutInvalidationContextMBS`

8.8.1 class `NSCollectionViewLayoutInvalidationContextMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An object that identifies the portions of your layout that need to be updated.

Notes: Invalidation contexts are a way to improve the efficiency of layout operations and must be supported explicitly by the layout object. Instead of invalidating the entire layout, you can create an invalidation layout object that specifies only the portions of the layout that changed. You then pass that invalidation context to the `invalidateLayoutWithContext:` method of the layout object.

Typically, you ask the layout object to create an invalidation context for you. The `NSCollectionViewLayoutMBS` class defines methods for creating a supported invalidation context. If you define a custom layout, you can define additional methods for creating invalidation contexts with custom information. Layout objects may also create invalidation contexts in response to specific changes. For example, layout objects automatically create invalidation contexts when you change the collection view, the data source, when you insert or delete items, and when you reload the collection view, the data.

8.8.2 Methods

8.8.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.8.4 `invalidateDecorationElementsOfKind(elementKind as String, indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Marks the specified decoration views as invalid so that their layout information can be updated.

Notes: `elementKind`: A string that identifies the type of the decoration views. This parameter must not be nil or an empty string.

`indexPaths`: An array of `NSIndexPathMBS` objects. Each index path contains the section in which the decoration view appears.

Call this method when you want the layout object to recompute attributes for one or more decoration views. All of the views must be of the type specified by the `elementKind` parameter. The method adds the views you specify to the `invalidatedDecorationIndexPaths` property. You can call this method more than once for the specified `elementKind` value.

8.8.5 invalidatedItemIndexPaths as NSIndexPathMBS()

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The set of items whose layout attributes are invalid.

Notes: The set contains zero or more NSIndexPathMBS objects, each of which identifies an invalid item.

8.8.6 invalidateItemsAtIndexPaths(indexPaths()) as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Marks the specified items as invalid so that their layout information can be updated.

Notes: indexPaths: An array of NSIndexPathMBS objects. Each index path represents an item whose layout needs to be recomputed.

Call this method when you want the layout object to recompute attributes for a specific set of items. The items you provide are added to the invalidatedItemIndexPaths property. You can call this method more than once to create a merged set of items.

8.8.7 invalidateSupplementaryElementsOfKind(elementKind as String, indexPaths()) as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Marks the specified supplementary views as invalid so that their layout information can be updated.

Notes: elementKind: A string that identifies the type of the supplementary views. This parameter must not be nil or an empty string.

indexPaths: An array of NSIndexPathMBS objects. Each index path contains the section in which the supplementary view appears.

Call this method when you want the layout object to recompute attributes for one or more supplementary views. All of the views must be of the type specified by the elementKind parameter. The method adds the views you specify to the invalidatedSupplementaryIndexPaths property. You can call this method more than once for the specified elementKind value.

8.8.8 Properties

8.8.9 contentOffsetAdjustment as NSPointMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The delta value to add to the collection view,Äôs content offset.

Notes: The content offset adjustment shifts the position of content inside the collection view by the specified amount. You use this value to make tweaks based on how you want to present your content. For example, you might use it to ensure that the first line of items is always lined up at the same position in the collection view,Äôs visible rectangle. When making adjustments, you can specify both positive and negative values. The default value of this property is zero/zero.

(Read and Write property)

8.8.10 `contentSizeAdjustment` as `NSSizeMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The delta value to add to the collection view,Äôs content size.

Notes: Use this property to update the size of the collection view,Äôs content area, as computed by the associated layout object. The default value of this property is zero/zero. Changing the value causes the collection view to add the specified height and width values to its content size. Thus, positive values grow the content area and negative values shrink it. You might add space around the content area to provide a visual buffer for your collection view content.

(Read and Write property)

8.8.11 `invalidateDataSourceCounts` as `Boolean`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the layout object should ask for new section and item counts.

Notes: The collection view sets this property in response to specific layout invalidation scenarios. For example, the collection view sets the property to true when you insert or delete items or call the collection view,Äôs `reloadData` method.

When this property is set to true, the layout object must query the data source for the new number of sections and items. IT should also update its layout based on the updated number of sections and items.

(Read only property)

8.8.12 `invalidatedDecorationIndexPaths` as `Dictionary`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A dictionary containing the decoration views whose layout attributes are invalid.

Notes: The keys in this dictionary are the element kind strings of the decoration views. The value for each key is an array object containing one or more `NSIndexPath` objects, each of which identifies the section containing the decoration view.

(Read only property)

8.8.13 invalidatedSupplementaryIndexPaths as Dictionary

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A dictionary containing the supplementary views whose layout attributes are invalid.

Notes: The keys in this dictionary are the element kind strings of the supplementary views. The value for each key is an array object containing one or more NSIndexPath objects, each of which identifies the section containing the supplementary view.

(Read only property)

8.8.14 invalidateEverything as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether all layout data should be marked as invalid.

Notes: The collection view sets this property in response to specific layout invalidation scenarios. For example, the collection view sets the property to true when you change the current layout object, change the data source of the collection view, or call the reloadData method and subsequently request a layout invalidation context.

When this property is set to true, the layout object must throw away all previous layout information and recompute it.

(Read only property)

8.9 class `NSCollectionViewLayoutMBS`

8.9.1 class `NSCollectionViewLayoutMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An abstract base class that you subclass and use to generate layout information for a collection view.

Notes: The job of a layout object is to perform the calculations needed to determine the placement and appearance of items, supplementary views, and other content in the collection view. The layout object does not apply the layout attributes it generates to the views in your interface. Instead, it passes those layout attributes to the collection view, which then creates the needed views and applies the layout attributes to them.

see also

<https://developer.apple.com/documentation/appkit/nscollectionviewlayout>

Blog Entries

- [NSCollectionView for Xojo](#)

8.9.2 Methods

8.9.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

8.9.4 `invalidateLayout`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Invalidates all layout information and triggers a layout update.

Notes: Call this method when you make changes that require updating all of the current layout information. This method marks the layout as invalid and returns right away, so you can call this method multiple times from the same block of code without triggering multiple layout updates. During the next update cycle, the collection view requests new layout information and updates its contents accordingly.

8.9.5 `invalidateLayoutWithContext(context as NSCollectionViewLayoutInvalidationContextMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Invalidates specific parts of the layout using the specified context object.

Notes: context: The context object indicating which parts of the layout need to be updated.

Call this method when you make changes that need to be reflected by the collection view, but which do not require the replacement of all of the layout information. You use this method to minimize the work performed by the layout object. Instead of optimizing everything, the specified context object indicates which parts of the layout need to be recomputed. All other layout information is left alone.

When implementing a custom layout, you can override this method and use it to process information provided by a custom invalidation context. You are not required to provide a custom invalidation context but might do so if you are able to provide additional properties that can help optimize layout updates. If you override this method, you must call `super` at some point in your implementation.

8.9.6 `NewInvalidationContext` as `NSCollectionViewLayoutInvalidationContextMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new invalidation context object.

8.9.7 `NewLayoutAttributes` as `NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new layout attributes object.

8.9.8 Properties

8.9.9 `collectionView` as `NSCollectionViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The collection view object currently using this layout.

Notes: When you assign a layout object to a collection view, the collection view automatically updates this property.

(Read only property)

8.10 class `NSCollectionViewMBS`

8.10.1 class `NSCollectionViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An ordered collection of data items displayed in a customizable layout.

Notes: The simplest type of collection view displays its items in a grid, but you can define layouts to arrange items however you like. For example, you might create a layout where items are arranged in a circle. You can also change layouts dynamically at runtime whenever you need to present items differently.

See also:

https://developer.apple.com/documentation/appkit/views_and_controls/collection_view

<https://developer.apple.com/documentation/appkit/nscollectionView>

Subclass of the `NSViewMBS` class.

Blog Entries

- [MBS Xojo Plugins, version 21.3pr2](#)
- [MBS Xojo Plugins, version 21.3pr1](#)
- [MBS Xojo Plugins, version 21.1pr3](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.0](#)
- [MBS Xojo Plugins, version 20.6pr2](#)

8.10.2 Methods

8.10.3 `backgroundColors` as `NSColorMBS()`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An array containing the collection view's background colors.

Notes: This property contains an array of `NSColorMBS` objects, representing the colors to use when drawing the background grid. Specifying an empty array or `nil` causes the collection view to use the default colors returned by the `controlAlternatingRowBackgroundColors` method.

When a background view is specified for the collection view, the colors in this property are ignored.

8.10.4 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new `NSCollectionView` object.

See also:

8.10. CLASS NSCOLLECTIONVIEWMBS	267
• 8.10.5 Constructor(Handle as Integer)	267
• 8.10.6 Constructor(left as double, top as double, width as double, height as double)	267

8.10.5 Constructor(Handle as Integer)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

Notes: Pass a valid handle to a NSTextField to initialize.

See also:

- 8.10.4 Constructor 266
- 8.10.6 Constructor(left as double, top as double, width as double, height as double) 267

8.10.6 Constructor(left as double, top as double, width as double, height as double)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSTextField object.

See also:

- 8.10.4 Constructor 266
- 8.10.5 Constructor(Handle as Integer) 267

8.10.7 deleteItems(indexPaths() as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Deletes the items at the specified index paths.

Example:

```
// remove item at index
```

```
Dim index As Integer = 1
```

```
// remove from data source
MainWindow.items.RemoveAt(index)
```

```
// remove also from view
Dim items() As NSIndexPathMBS
items.Append NSIndexPathMBS.indexPathForItem(index)
```

```
MainWindow.collectionView.deleteItems items
```

Notes: `indexPaths`: A set of `NSIndexPathMBS` objects, each of which includes a section and item index corresponding to the insertion point of a single item. Specifying `nil` for this parameter raises an exception.

After removing items from your data source object, use this method to synchronize those changes with the collection view. Calling this method lets the collection view know that it must update its internal data structures and possibly update its visual appearance. In response, the collection view asks the layout object to update the positions of the remaining objects. If the layout object indicates that there are changes to the visible items, the collection view animates the affected items into place.

When inserting or deleting multiple sections and items, you can animate all of your changes at once using the `performBatchUpdates` method.

8.10.8 `deleteSections(sections as NSIndexPathSetMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Deletes the specified sections and their contained items.

Notes: `sections`: An index set containing the indexes of the sections that you want to delete. This parameter must not be `nil`.

Use this method to delete entire sections and their contained items. Always update your data source object before calling this method. Calling this method kicks off an update (and possible animations) to delete the specified sections. Specifically, the collection view asks the layout object for the final layout attributes for any deleted sections and may also ask for updated layout attributes for any remaining sections. If the layout attributes of any visible items changed, those changes are animated into place.

When inserting or deleting multiple sections and items, you can animate all of your changes at once using the `performBatchUpdates` method.

8.10.9 `deselectAll`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Deselects all items in the collection view.

Notes: This method works only when the `selectable` and `allowsEmptySelection` properties are both `true YES`. If either property is set to `false`, this method quietly does nothing and any connected menu item is disabled.

This method consults the delegate object regarding the selection. Specifically, it calls the delegate's

shouldDeselectItems method to see if the items should be selected. For any items that are selected, it calls the didDeselectItems method.

8.10.10 deselectItems(indexPaths() as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Removes the specified items from the current selection.

Notes: indexPaths: The index paths of the items you want to deselect.

Use this method to reduce the current selection. If you want to animate the deselection of the new items, call this method on the collection view,Äôs animator proxy object instead. This method does not call any methods of the delegate object when making the selection.

8.10.11 frameForItem(Index as Integer) as NSRectMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the frame of the collection view item at the specified index.

Notes: index: The index of the collection view item.

Return the frame calculated by the receiver where it intends to place the subview for the NSCollectionViewItem at the given index. The rectangle is returned in the collection view,Äôs coordinate system.

You can use this method in the draggingImageForItemsAtIndexes method to determine which views are in the visible portion of the enclosing scroll view.

Overriding this method will have no effect on the collection view,Äôs subview layout.

See also:

- 8.10.12 frameForItem(Index as Integer, numberOfItems as Integer) as NSRectMBS

8.10.12 frameForItem(Index as Integer, numberOfItems as Integer) as NSRectMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the frame of the collection view item at the specified index.

Notes: index: The index of the collection view item.

Returns the frame calculated by the receiver where it intends to place the subview for the NSCollectionViewItem at the given index. The rectangle is returned in the collection view,Äôs coordinate system.

You can use this method in the `draggingImageForItemsAtIndexes` method to determine which views are in the visible portion of the enclosing scroll view.

Overriding this method will have no effect on the collection view,Äôs subview layout.

See also:

- 8.10.11 `frameForItem(Index as Integer) as NSRectMBS` 269

8.10.13 `indexPathForItem(indexPath as NSCollectionViewItemMBS) as NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index path of the specified item.

Notes: Returns the item,Äôs index path or nil if the item is not in the collection view.

See also:

- 8.10.14 `indexPathForItem(x as double, y as double) as NSIndexPathMBS` 270

8.10.14 `indexPathForItem(x as double, y as double) as NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index path of the specified item.

Example:

```
// remove item at index
```

```
Dim index As Integer = 1
```

```
// remove from data source
```

```
MainWindow.items.RemoveAt(index)
```

```
// remove also from view
```

```
Dim items() As NSIndexPathMBS
```

```
items.Append NSIndexPathMBS(indexPathForItem(index))
```

```
MainWindow.collectionView.deleteItems items
```

Notes: Returns the item,Äôs index path or nil if the item is not in the collection view.

See also:

- 8.10.13 `indexPathForItem(indexPath as NSCollectionViewItemMBS) as NSIndexPathMBS` 270

8.10.15 indexPathsForVisibleItems as NSIndexPathMBS()

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index paths of the currently active items.

Notes: The set of NSIndexPathMBS objects corresponding to the currently visible items.

The index paths returned by this method belong to items that are active and currently being managed by the collection view. As a result, the returned set may include index paths for items that are outside of the collection view,Ãs actual visible rectangle. For example, it may contain index paths for items that were recently visible but have since been scrolled out of view. To test whether an item is visible, check to see if its frame rectangle intersects the visibleRect of the collection view.

8.10.16 indexPathsForVisibleSupplementaryElementsOfKind(elementKind as String) as NSIndexPathMBS()

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index paths of the currently active supplementary views.

Notes: elementKind: The kind of the supplementary views you want returned. The layout object defines the kinds of supplementary views it supports. This parameter must not be nil.

Return the set of NSIndexPathMBS objects. The returned array may be empty.

The index paths returned by this method correspond to supplementary views that are active and currently being managed by the collection view. The set may include index paths for views that are outside of the collection view,Ãs actual visible rectangle. For example, it might contain index paths for views that were recently visible but have since been scrolled out of the visible rectangle. To test whether a view is actually visible, check to see if its frame rectangle intersects the visibleRect of the collection view.

8.10.17 insertItems(indexPaths()) as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Inserts new items into the collection view at the specified locations.

Notes: indexPaths: A set of NSIndexPathMBS objects, each of which includes a section and item index corresponding to the insertion point of a single item. Specifying nil for this parameter raises an exception.

After adding new items to your data source object, use this method to synchronize those changes with the collection view. Calling this method lets the collection view know that it must update its internal data structures and possibly update its visual appearance. In response, the collection view asks the layout object for information about the new objects. If the layout object indicates that the new items should appear on-

screen, the collection view asks the data source to provide the appropriate content, animating that content into position as needed.

When inserting or deleting multiple sections and items, you can animate all of your changes at once using the `performBatchUpdates` method.

8.10.18 `insertSections(sections as NSIndexSetMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Inserts new sections at the specified indexes.

Notes: `sections`: An index set containing the indexes at which you want to insert new sections. This parameter must not be nil.

This method tells the collection view to insert the specified sections and update itself. Always update your data source object before calling this method. Calling this method kicks off an update (and possible animations) to add the new sections. Specifically, the collection view asks the layout object for any updated layout attributes related to the new sections or any existing sections. If the layout attributes of any visible items changed, those changes are animated into place.

When inserting or deleting multiple sections and items, you can animate all of your changes at once using the `performBatchUpdates` method.

8.10.19 `item(index as Integer) as NSCollectionViewItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the item associated with the specified index path.

Notes: The item for the specified index path or nil if no item is available.

For efficiency, the collection view does not create items until they are needed, and usually it creates them only when they need to be displayed onscreen. If the collection view does not currently have an item for the specified index path, because that item would be positioned offscreen, this method returns nil.

See also:

- 8.10.20 `item(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS`

272

8.10.20 `item(indexPath as NSIndexPathMBS) as NSCollectionViewItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the item associated with the specified index path.

Notes: The item for the specified index path or nil if no item is available.

For efficiency, the collection view does not create items until they are needed, and usually it creates them only when they need to be displayed onscreen. If the collection view does not currently have an item for the specified index path, because that item would be positioned offscreen, this method returns nil.

See also:

- 8.10.19 `item(index as Integer)` as `NSCollectionViewItemMBS`

272

8.10.21 `layoutAttributesForItem(indexPath as NSIndexPathMBS)` as `NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the layout information for the item at the specified index path.

Notes: The layout attributes of the item or nil if no item exists at the specified path.

This method updates the layout information as needed before returning the specified attributes. Always use this method to retrieve the layout attributes for items in the collection view. Do not query the layout object directly.

8.10.22 `layoutAttributesForSupplementaryElementOfKind(kind as String, indexPath as NSIndexPathMBS)` as `NSCollectionViewLayoutAttributesMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the layout information for the supplementary view at the specified index path.

Notes: `kind`: The kind of the supplementary view whose attributes you want. The layout object defines the kinds of supplementary views it supports. This parameter must not be nil.

`indexPath`: The index path of the supplementary view. Normally, this path

Returns the layout attributes of the supplementary view or nil if no item exists at the specified path.

This method updates the layout information as needed before returning the specified attributes. Always use this method to retrieve the layout attributes for supplementary views in the collection view. Do not query the layout object directly.

8.10.23 `makeItem(indexPath as NSIndexPathMBS)` as `NSCollectionViewItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates or returns a reusable item object of the specified type.

Notes: `indexPath`: The index path specifying the location of the item. The data source object receives this information in its `itemForRepresentedObjectAtIndex` method and you should just pass it along.

8.10.24 `makeSupplementaryViewOfKind(elementKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewSectionHeaderViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Creates or returns a reusable supplementary view of the specified type.

Notes: `elementKind`: The kind of supplementary view to create. This value is defined by the layout object. This parameter must not be an empty string or nil.

`identifier`: The reuse identifier for the specified item. This is the identifier you specified when registering the supplementary view. This parameter must not be nil.

`indexPath`: The index path specifying the location of the supplementary view. The data source object receives this information in its `viewForSupplementaryElementOfKind` method and you should just pass it along.

Returns a view that adopts the `NSCollectionViewElement` protocol, e.g. a `NSCollectionViewSectionHeaderViewMBS` object.

This method looks for a recycled supplementary view of the specified type and returns it if one exists. If one does not exist, it creates it one.

8.10.25 `moveItem(indexPath as NSIndexPathMBS, toIndexPath as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Moves an item from one location to another in the collection view.

Notes: `indexPath`: The index path of the item that you want to move. This parameter must not be nil.

`newIndexPath`: The index path of the item,Â’s new location. This parameter must not be nil.

After rearranging items in your data source object, use this method to synchronize those changes with the collection view. Calling this method lets the collection view know that it must update its internal data structures and possibly update its visual appearance. You can move the item to a different section or to a new location in the same section. The collection view updates the layout as needed to account for the move, animating cells into position in response.

When inserting or deleting multiple sections and items, you can animate all of your changes at once using the `performBatchUpdates` method.

8.10.26 `moveSection(section as Integer, toSection as Integer)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Moves a section from its current location to a new location.

Notes: `section`: The index of the section that you want to move.

`newSection`: The new index at which to insert the section.

Use this method to reorganize sections and their contained items. Always update your data source object before calling this method. Calling this method kicks off an update (and possible animations) to move the specified section to its new location. Specifically, the collection view asks the layout object for any updated layout attributes related to the new sections or any existing sections. If the layout attributes of any visible items changed, those changes are animated into place.

When inserting or deleting multiple sections and items, you can animate all of your changes at once using the `performBatchUpdates` method.

8.10.27 `numberOfItemsInSection(section as Integer) as Integer`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the number of items in the specified section.

Notes: `section`: The index of the section whose item count you want. This index is 0-based.

Returns the number of items in the section.

Use this method to get the number of items currently displayed by the collection view for the specified section. Do not call the methods of the data source to get this information.

8.10.28 `reloadData`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Reloads all of the data for the collection view.

Notes: Call this method when the data in your data source object changes or when you want to force the collection view to update its contents. When you call this method, the collection view discards any currently visible items and views and redisplay them. For efficiency, the collection view displays only the items and supplementary views that are visible after reloading the data. If the collection view's size changes as a result of reloading the data, the collection view adjusts its scrolling offsets accordingly.

Do not call this method in the middle of animation blocks where items are being inserted or deleted. The methods used to insert and delete items automatically update the collection view's contents.

8.10.29 reloadItems(indexPaths() as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Reloads only the specified items.

Notes: indexPaths: The index paths of the specific items that you want to reload. Specifying nil for this parameter raises an exception.

Call this method to update specific items in your collection view. You call this method when the underlying data for those items changes and you want to update the visual appearance of those items. When you call this method, the collection view discards the specified items and asks your data source to provide new ones. For efficiency, the collection view requests only the items that are visible.

8.10.30 reloadSections(sections as NSIndexPathMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Reloads the data in the specified sections of the collection view.

Notes: sections: The indexes of the sections that you want to reload. Specifying nil for this parameter raises an exception.

Call this method when the data for the specified sections changes or when you want to force the appearance of those sections to be updated. When you call this method, the collection view discards visible elements in the section along with any cached attributes for those elements. For efficiency, it then asks the layout object to provide fresh attributes for only the visible items and views and requests new views for those elements. Do not call this method in the middle of animation blocks where items are being inserted or deleted. The methods used to insert and delete items automatically update the collection view's contents.

8.10.31 scrollToItems(indexPaths() as NSIndexPathMBS, scrollPosition as Integer)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Scrolls the collection view contents until the specified items are visible.

Example:

```
dim collectionView as NSCollectionViewMBS
```

```
Dim indexPaths() As NSIndexPathMBS  
indexPaths.Append NSIndexPathMBS.indexPathForItem(0)
```

```
collectionView.scrollToItems(indexPaths, NSCollectionViewMBS.NSCollectionViewScrollPositionTop)
```

Notes: `indexPaths`: The index paths of the items. The layout attributes of these items define the bounding box that needs to be scrolled onscreen.

`scrollPosition`: The options for scrolling the bounding box of the specified items into view. You may combine one vertical and one horizontal scrolling option when calling this method. Specifying more than one option for either the vertical or horizontal directions raises an exception.

To animate the scrolling operation, call this method on the collection view,Äôs animator proxy object instead.

8.10.32 `selectAll`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Selects all items in the collection view, if doing so is possible.

Notes: This method works only when the `selectable` and `allowsMultipleSelection` properties are both true. If either property is set to false, this method quietly does nothing and any connected menu item is disabled.

This method consults the delegate object regarding the selection. Specifically, it calls the delegate,Äôs `shouldSelectItems` method to see if the items should be selected. For any items that are selected, it calls the `didSelectItems` method.

8.10.33 `selection as NSIndexPathMBS()`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The array of index paths representing the currently selected items.

Notes: This property reflects the index paths of the currently selected items, where each index path contains a section number and an index number for the item in that section. This property is updated automatically when the user selects items interactively. You can also change the selection programmatically by assigning a new value to this property. To animate changes to the selection, call this method on the collection view,Äôs animator proxy object instead.

It is a programmer error to specify an index path that does not refer to a valid item in the data source. If you specify an invalid index path, this method raises an exception.

This property is key-value observable. Other methods that modify the selection automatically update this property.

8.10.34 `selectItems(indexPaths() as NSIndexPathMBS, scrollPosition as Integer)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Adds the specified items to the current selection and optionally scrolls the items into position.

Notes: `indexPaths`: The index paths of the items you want to select.

`scrollPosition`: The options for scrolling the newly selected items into view. You may combine one vertical and one horizontal scrolling option when calling this method. Specifying more than one option for either the vertical or horizontal directions raises an exception.

Use this method to extend the current selection. If you want to animate the selection of the new items, call this method on the collection view,Ãs animator proxy object instead. This method does not call any methods of the delegate object when making the selection.

8.10.35 `setBackgroundColors(Colors() as NSColorMBS = nil)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Sets array containing the collection view,Ãs background colors.

Notes: This property contains an array of `NSColorMBS` objects, representing the colors to use when drawing the background grid. Specifying an empty array or `nil` causes the collection view to use the default colors returned by the `controlAlternatingRowBackgroundColors` method.

When a background view is specified for the collection view, the colors in this property are ignored.

8.10.36 `setDraggingSourceOperationMask(mask as integer, local as boolean)`

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: Configures the default value returned from `draggingSourceOperationMaskForLocal`.

Notes: `mask`: The types of drag operations allowed.

`isLocal`: If true, `mask` applies when the drag destination object is in the same application as the receiver; if false, `mask` applies when the destination object is outside the receiver's application.

By default, `draggingSourceOperationMaskForLocal` returns `NSDragOperationEvery` when `isLocal` is true and `NSDragOperationNone` when `isLocal` is false.

Available in Mac OS X v10.5 and later.

8.10.37 `setSelection(indexPaths() as NSIndexPathMBS)`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Sets the selection.

8.10.38 supplementaryViewForElementKind(elementKind as String, indexPath as NSIndexPathMBS) as NSCollectionViewSectionHeaderViewMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns the supplementary view associated with the specified index path.

Notes: elementKind: The kind of the supplementary views you want returned. The layout object defines the kinds of supplementary views it supports. This parameter must not be nil.

indexPath: The index path whose supplementary view you want.

Returns the view for the specified index path or nil if no view is available.

For efficiency, the collection view does not create supplementary views until they are needed. Typically, views are created only when they need to be displayed onscreen. If the collection view does not currently have a supplementary view for the specified index path, because that view would be positioned offscreen, this method returns nil.

8.10.39 toggleSectionCollapse(sender as NSViewMBS)

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: Collapses the section in which the sender resides into a single horizontally scrollable row.

Notes: Please pass the button clicked as sender.

8.10.40 visibleItems as NSCollectionViewItemMBS()

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns an array of the actively managed items in the collection view.

Notes: An array of NSCollectionViewItemMBS objects. The returned array may be empty.

The items returned by this method represent the ones that are active and currently being managed by the collection view. This array may contain items that are outside of the collection view,Â actual visible rectangle. For example, it may contain items that were recently visible but have since been scrolled out of view. To test whether an item is actually visible, check to see if its frame rectangle intersects the visibleRect of the collection view.

8.10.41 visibleSupplementaryViewsOfKind(elementKind as String) as NSViewMBS()

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Returns an array of the actively managed supplementary views in the collection view.

Notes: `elementKind`: The kind of the supplementary views you want returned. The layout object defines the kinds of supplementary views it supports. This parameter must not be nil.

Returns an array of view objects. The returned array may be empty.

The views returned by this method represent the ones that are active and are currently being managed by the collection view. The array may contain supplementary views that are outside of the collection view,Ãs actual visible rectangle. For example, it might contain views that were recently visible but have since been scrolled out of the visible rectangle. To test whether a view is actually visible, check to see if its frame rectangle intersects the `visibleRect` of the collection view.

8.10.42 Properties

8.10.43 `allowsEmptySelection` as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the collection view may have no selected items.

Notes: The default value of this property is true, which allows the collection view to have no selected items. Setting this property to false causes the collection view to always leave at least one item selected.
(Read and Write property)

8.10.44 `allowsMultipleSelection` as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the user may select more than one item in the collection view.

Notes: The value of this property is true if the collection view supports the selection of more than one item at a time. The default value of this property is false. Changing the value from true to false reduces the current selection to the first item in the selected group.
(Read and Write property)

8.10.45 `animator` as `NSCollectionViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The animator proxy object.

Example:

```
dim collectionView as NSCollectionViewMBS // your control
```

```

Dim context As NSAnimationContextMBS = NSAnimationContextMBS.currentContext
context.beginGrouping
context.allowsImplicitAnimation = True
context.duration = 0.5

Dim view As Variant = collectionView animator
Dim animator As NSCollectionViewMBS = view

// scroll to top
Dim indexPaths() As NSIndexPathMBS
indexPaths.Append NSIndexPathMBS.indexPathForItem(0)

animator.scrollToItems(indexPaths, NSCollectionViewMBS.NSCollectionViewScrollPositionTop)

context.endGrouping

```

Notes: You can call methods on the animator instead of the direct object. Methods with animation support will then be animated.
(Read only property)

8.10.46 `backgroundView` as `NSViewMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The background view placed behind all items and supplementary views.

Notes: The view you assign to this property is positioned underneath all other content and sized automatically to match the enclosing clip view,Äôs frame. The view itself does not scroll with the rest of the collection view content. The view,Äôs layer redraw policy is also changed to `NSViewLayerContentsRedrawNever`.

In macOS 10.12 and later, a collection view that sets both `backgroundView` and `backgroundColors` shows `backgroundColors [0]` through all areas that are not opaquely covered by the `backgroundView`.
(Read and Write property)

8.10.47 `backgroundViewScrollsWithContent` as `Boolean`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the collection view,Äôs background view scrolls with the items and other content.

Notes: The default value of this property is false, which means that `backgroundView` (if it exists) fills the collection view's visible area and remains stationary when the collection view's content is scrolled. When

the value of this property is YES, `backgroundView` matches the collection view's frame and scrolls with the collection view's items and other content.

Changing the value of this property also changes the background view,Äôs parent. When `backgroundView` floats behind the scrolling content, it is a sibling of the collection view,Äôs clip view. When it scrolls with the collection view,Äôs content, it is a subview of the collection view.

(Read and Write property)

8.10.48 `collectionViewLayout` as `NSCollectionViewLayoutMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The layout object used to organize the collection view,Äôs content.

Notes: Typically, you specify the layout object at design time in Interface Builder. The layout object works with your data source object to generate the needed items and views to display. In macOS 10.11 and later, using a data source object is recommended, but you may specify nil for this property if your collection view does not use a data source object. The collection view uses the `NSCollectionViewGridLayout` object by default.

Assigning a new value to this property changes the layout object and causes the collection view to update its contents immediately and without animations.

(Read and Write property)

8.10.49 `firstResponder` as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the collection view is the first responder.

Notes: The value of this property is true when the collection view is the first responder. This property is fully key-value observing compliant.

(Read only property)

8.10.50 `numberOfSections` as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The number of sections in the collection view.

Notes: This property contains the number of sections reported by the data source object. If the collection view does not use a data source object, the value in this property is 1.

(Read only property)

8.10.51 selectable as Boolean

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the user may select items in the collection view.

Notes: The value of this property is true when the collection view allows the user to select items, or false when it does not. You can set selections programmatically regardless of this setting.

The default value of this property is false. Changing the value from YES to NO removes the current selection if there is one.

(Read and Write property)

8.10.52 Constants

Scroll Position

Constant	Value	Description
<code>NSCollectionViewScrollPositionBottom</code>	4	Scroll so that the bottom edge of the bounding box is adjacent to the bottom edge of the collection view,Ãs bounds. This option must not be combined with the <code>NSCollectionViewScrollPositionTop</code> , <code>NSCollectionViewScrollPositionCenteredVertically</code> , <code>NSCollectionViewScrollPositionNearestHorizontalEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionCenteredHorizontally</code>	16	Scroll so that the selected items,Ã bounding box is centered horizontally within the collection view,Ãs bounds. This option must not be combined with the <code>NSCollectionViewScrollPositionLeft</code> , <code>NSCollectionViewScrollPositionRight</code> , <code>NSCollectionViewScrollPositionLeadingEdge</code> , <code>NSCollectionViewScrollPositionTrailingEdge</code> , <code>NSCollectionViewScrollPositionNearestVerticalEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionCenteredVertically</code>	2	Scroll so that the bounding box of the selected items is centered vertically within the collection view,Ãs bounds. This option must not be combined with the <code>NSCollectionViewScrollPositionTop</code> , <code>NSCollectionViewScrollPositionBottom</code> , or <code>NSCollectionViewScrollPositionNearestHorizontalEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionLeadingEdge</code>	64	Scroll so that the leading edge of the selected items,Ã bounding box is adjacent to the leading edge of the collection view,Ãs bounds. Use this option to support both left-to-right and right-to-left scrolling appropriately. This option must not be combined with the <code>NSCollectionViewScrollPositionLeft</code> , <code>NSCollectionViewScrollPositionCenteredHorizontally</code> , <code>NSCollectionViewScrollPositionRight</code> , <code>NSCollectionViewScrollPositionTrailingEdge</code> , or <code>NSCollectionViewScrollPositionNearestVerticalEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionLeft</code>	8	Scroll so that the left edge of the selected items,Ã bounding box is adjacent to the left edge of the collection view,Ãs bounds. This option must not be combined with the <code>NSCollectionViewScrollPositionCenteredHorizontally</code> , <code>NSCollectionViewScrollPositionRight</code> , <code>NSCollectionViewScrollPositionLeadingEdge</code> , <code>NSCollectionViewScrollPositionTrailingEdge</code> , or <code>NSCollectionViewScrollPositionNearestVerticalEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionNearestHorizontalEdge</code>	512	Scroll so that the bounding box is adjacent to the nearest edge (leading or trailing) of the collection view. This option must not be combined with the <code>NSCollectionViewScrollPositionTop</code> , <code>NSCollectionViewScrollPositionCenteredVertically</code> , <code>NSCollectionViewScrollPositionBottom</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionNearestVerticalEdge</code>	256	Scroll so that the bounding box is adjacent to the nearest edge (leading or trailing) of the collection view. Use this option to support both left-to-right and right-to-left scrolling appropriately. This option must not be combined with the <code>NSCollectionViewScrollPositionLeft</code> , <code>NSCollectionViewScrollPositionCenteredHorizontally</code> , <code>NSCollectionViewScrollPositionRight</code> , <code>NSCollectionViewScrollPositionLeadingEdge</code> , <code>NSCollectionViewScrollPositionTrailingEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionNone</code>	0	Do not scroll.
<code>NSCollectionViewScrollPositionRight</code>	32	Scroll so that the right edge of the selected items,Ã bounding box is adjacent to the right edge of the collection view,Ãs bounds. This option must not be combined with the <code>NSCollectionViewScrollPositionLeft</code> , <code>NSCollectionViewScrollPositionCenteredHorizontally</code> , <code>NSCollectionViewScrollPositionLeadingEdge</code> , <code>NSCollectionViewScrollPositionTrailingEdge</code> , or <code>NSCollectionViewScrollPositionNearestVerticalEdge</code> options, but may be combined with other options.
<code>NSCollectionViewScrollPositionTop</code>	1	Scroll so that the top edge of the selected items,Ã bounding box is adjacent to the top edge of the collection view,Ãs bounds. This option must not be combined with the <code>NSCollectionViewScrollPositionCenteredVertically</code> , <code>NSCollectionViewScrollPositionBottom</code> , or <code>NSCollectionViewScrollPositionNearestHorizontalEdge</code> options, but may be combined with other options.

8.11 class NSCollectionViewSectionHeaderViewMBS

8.11.1 class NSCollectionViewSectionHeaderViewMBS

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: A protocol that defines a button to control the collapse of a collection view,Äôs section.

Notes: A collection view can support a section that can collapse into a single horizontally scrollable row, similar to the groupings in the icon view in Finder. To ensure that the collection view can communicate with the button that controls the collapsing of a section, the section header view object connects to the button via sectionCollapseButton property.

Subclass of the NSViewMBS class.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.3](#)
- [MBS Xojo Plugins, version 21.3pr4](#)
- [MBS Xojo Plugins, version 21.3pr2](#)

8.11.2 Methods

8.11.3 Constructor

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSCollectionViewSectionHeaderView object.

See also:

- 8.11.4 Constructor(Handle as Integer) 285
- 8.11.5 Constructor(left as double, top as double, width as double, height as double) 286

8.11.4 Constructor(Handle as Integer)

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSCollectionViewSectionHeaderView object.

Notes: Pass a valid handle to a NSCollectionViewSectionHeaderView to initialize.

See also:

- 8.11.3 Constructor 285
- 8.11.5 Constructor(left as double, top as double, width as double, height as double) 286

8.11.5 Constructor(left as double, top as double, width as double, height as double)

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSCollectionViewSectionHeaderView object.
See also:

- 8.11.3 Constructor 285
- 8.11.4 Constructor(Handle as Integer) 285

8.11.6 Properties

8.11.7 sectionCollapseButton as NSButtonMBS

Plugin Version: 21.3, Platform: macOS, Targets: Desktop only.

Function: A control that lets users collapse and open a collection view section.

Notes: For the best user experience, set this property to the button that lets users control the collapsing of a section so that the collection view can show and hide the button appropriately, based on whether the section,Ãs items can be displayed in the available space. The collection view uses its toggleSectionCollapse property to access this button.
(Read and Write property)

8.12 class NSCollectionViewTransitionLayoutMBS

8.12.1 class NSCollectionViewTransitionLayoutMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: An object that implements custom behaviors when changing from one layout to another in a collection view.

Notes: Transition layout objects are commonly used to implement interactive transitions between layouts, where the transition itself is driven by a gesture recognizer.

Subclass of the NSCollectionViewLayoutMBS class.

8.12.2 Methods

8.12.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

See also:

- 8.12.4 Constructor(currentLayout as NSCollectionViewLayoutMBS, nextLayout as NSCollectionViewLayoutMBS) 287

8.12.4 Constructor(currentLayout as NSCollectionViewLayoutMBS, nextLayout as NSCollectionViewLayoutMBS)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Initializes and returns the transition layout object.

Notes: currentLayout: The layout object currently in use by the collection view.

newLayout: The new layout object that is about to be installed into the collection view.

This method initializes the transition layout object and saves references to the current and new layout objects. If you subclass and implement your own initialization method, you must call this method to initialize the superclass.

See also:

- 8.12.3 Constructor

8.12.5 Properties

8.12.6 `currentLayout` as `NSCollectionViewLayoutMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The collection view,Äôs current layout object.

Notes: Use this object to retrieve the initial layout attributes for elements of the collection view. If the transition is ultimately cancelled, the collection view animates its items back to the attributes provided by this object.

(Read only property)

8.12.7 `nextLayout` as `NSCollectionViewLayoutMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The collection view,Äôs new layout object.

Notes: Use this object to retrieve the final layout attributes for elements of the collection view. If the transition completes as expected, the collection view animates its items to the attributes provided by this object.

(Read only property)

8.12.8 `transitionProgress` as `Double`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The completion percentage of the transition.

Notes: During the transition, set the value of this property periodically and call the `invalidateLayout` method to force the collection view to update item positions. For example, when driving a transition using a gesture recognizer, you can set this property from the handler method of your gesture recognizer.

(Read and Write property)

8.12.9 `valueForKey(AnimatedKey as String)` as `Double`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: Queries or sets the value of a key whose value you use during the animation.

Notes: `value:` The value of the key.

`key:` The key that you define for your custom transition layout.

Use this method to query or update the value of a specific key that you use in your custom transition layout.

Use this method to retrieve floating-point values that relate to laying out the contents of your collection view. The key you specify is a string that you define and that has some meaning to your layout,Ãs implementation. At points during an interactive transition, you can assign new values to that key.
(Read and Write computed property)

8.13 class `NSCollectionViewUpdateItemMBS`

8.13.1 class `NSCollectionViewUpdateItemMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: A description of a single change to make to an item in a collection view.

Notes: You do not create instances of this class directly. When updating its content, the collection view object creates them and passes them to the layout object, `prepareForCollectionViewUpdates` method, which can use them to prepare for the upcoming changes.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

8.13.2 Methods

8.13.3 Constructor

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The private constructor.

8.13.4 Properties

8.13.5 `indexPathAfterUpdate` as `NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The index path of the item after the update.

Notes: The value of this property is nil for an action of type `UpdateActionDelete`.
(Read only property)

8.13.6 `indexPathBeforeUpdate` as `NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The index path of the item before the update.

Notes: The value of this property is nil for an action of type `UpdateActionInsert`.
(Read only property)

8.13.7 updateAction as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The action being performed on the item.

Notes: For a list of relevant actions, see constants.

(Read only property)

8.13.8 Constants

Update Actions

Constant	Value	Description
UpdateActionDelete	1	Remove the action from the collection view.
UpdateActionInsert	0	Insert the item into the collection view.
UpdateActionMove	3	Move the item from its current location to a new location.
UpdateActionNone	4	Take no action on the item.
UpdateActionReload	2	Reload the item, which consists of deleting and then inserting the item.

8.14 class NSIndexPathMBS

8.14.1 class NSIndexPathMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: A list of indexes that together represent the path to a specific location in a tree of nested arrays.

Notes: Each index in an index path represents the index into an array of children from one node in the tree to another, deeper, node.

Blog Entries

- [MBS Xojo Plugins, version 23.6pr1](#)

8.14.2 Methods

8.14.3 compare(other as NSIndexPathMBS) as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Indicates the depth-first traversal order of the receiving index path and another index path.

Notes: Returns the depth-first traversal ordering of the receiving index path and indexPath.

NSOrderedAscending = -1: The receiving index path comes before indexPath.

NSOrderedDescending = 1: The receiving index path comes after indexPath.

NSOrderedSame = 0: The receiving index path and indexPath are the same index path.

8.14.4 Constructor(Index as Integer)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Initializes an index path with a single node.

Notes: Returns an initialized NSIndexPath object representing a one-node index path with index.

See also:

- 8.14.5 Constructor(Indexes() as Integer)

292

8.14.5 Constructor(Indexes() as Integer)

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Initializes an index path with the given nodes and length.

Notes: indexes: Array of indexes to make up the index path.

See also:

- 8.14.4 Constructor(Index as Integer)

8.14.6 copy as NSIndexPathMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Creates a copy of the object.

8.14.7 indexAtPosition(position as Integer) as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Provides the value at a particular node in the index path.

Notes: position: Index value of the desired node. Node numbering starts at zero.

Returns the index value at node or NSNotFound (-1) if the node is outside the range of the index path.

8.14.8 indexes as Integer()

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Queries indexes.

8.14.9 indexPathByAddingIndex(index as Integer) as NSIndexPathMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an index path containing the nodes in the receiving index path plus another given index.

Notes: index: Index to append to the index path,Ãs indexes.

Returns a new index path containing the receiving index path,Ãs indexes and index.

8.14.10 indexPathByRemovingLastIndex as NSIndexPathMBS

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an index path with the nodes in the receiving index path, excluding the last one.

Notes: Returns an empty NSIndexPath instance if the receiving index path,Ãs length is 1 or less.

In OS X v10.4 this method returns nil when the length of the receiving index path is 1 or less. On iOS and macOS 10.5 and later this method never returns nil.

8.14.11 `indexPathForItem(item as Integer, section as Integer = 0)` as `NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Initializes an index path with the indexes of a specific item and section in a collection view.

Notes: item: An index number identifying an item in a `NSCollectionViewMBS` object in a section identified by the section parameter.

section: An index number identifying a section in a `NSCollectionViewMBS` object.

8.14.12 `indexPathWithIndex(index as Integer)` as `NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Initializes an index path with a single node.

Notes: Returns an initialized `NSIndexPath` object representing a one-node index path with index.

8.14.13 `indexPathWithIndexes(indexes() as Integer)` as `NSIndexPathMBS`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: Initializes an index path with the given nodes and length.

Notes: indexes: Array of indexes to make up the index path.

8.14.14 Properties

8.14.15 `item as Integer`

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: An index number identifying an item in a section of a collection view.

Notes: The section the item is in is identified by the value of section.

(Read only property)

8.14.16 length as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: The number of nodes in the index path.

Notes: (Read only property)

8.14.17 section as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop, Console & Web.

Function: An index number identifying a section in a table view or collection view.

Notes: (Read only property)

Chapter 9

Cocoa Controls

9.1 class BevelButton

9.1.1 class BevelButton

Plugin Version: 3.0, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.1.2 Methods

9.1.3 NSButtonMBS as NSButtonMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox BevelButton1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.2 class CanvasGesturesMBS

9.2.1 class CanvasGesturesMBS

Plugin Version: 18.2, Platform: macOS, Targets: Desktop only.

Function: The class to catch canvas gesture events.

Notes: Please use one global instance of this class to catch the events for all canvases you track.

Updated class for v23.2 to work with DesktopCanvas control.

Blog Entries

- [MBS Xojo Plugins, version 24.1pr3](#)
- [News from the MBS Xojo Plugins Version 23.2](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 23.2](#)
- [MBS Xojo Plugins, version 23.2pr1](#)
- [MBS Xojo Plugins, version 22.5pr7](#)
- [MBS Xojo Plugins, version 19.2pr7](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 18.2](#)
- [MBS Xojo Plugins, version 18.2pr5](#)

Videos

- [Presentation from Munich conference about MBS Plugins.](#)
- [Presentation from Xojo Developer Conference 2019 in Miami.](#)

Xojo Developer Magazine

- [21.4, page 10: News](#)
- [17.5, page 40: What's New in the MBS Plugins, With the Plugins growing every year, here are new capabilities you may have missed by Stefanie Juchmes](#)
- [16.4, page 11: News](#)

9.2.2 Methods

9.2.3 AddCanvas(c as Canvas)

Plugin Version: 18.2, Platform: macOS, Targets: Desktop only.

9.2. CLASS CANVASGESTURESMB 299

Function: Adds a canvas to the list of canvases to track.

Notes: Please add canvas controls in Window.Open event.

See also:

- 9.2.4 AddCanvas(c as DesktopCanvas) 299

9.2.4 AddCanvas(c as DesktopCanvas)

Plugin Version: 23.2, Platform: macOS, Targets: Desktop only.

Function: Adds a canvas to the list of canvases to track.

Notes: Please add canvas controls in Window.Open event.

See also:

- 9.2.3 AddCanvas(c as Canvas) 298

9.2.5 Constructor

Plugin Version: 18.2, Platform: macOS, Targets: Desktop only.

Function: The constructor.

9.2.6 Destructor

Plugin Version: 18.2, Platform: macOS, Targets: Desktop only.

Function: The destructor.

9.2.7 RemoveCanvas(c as Canvas)

Plugin Version: 18.2, Platform: macOS, Targets: Desktop only.

Function: Removes a canvas from the list.

Notes: Please remove canvas controls in Window.Close event.

The plugin tries to remove all when a window closes automatically.

See also:

- 9.2.8 RemoveCanvas(c as DesktopCanvas) 299

9.2.8 RemoveCanvas(c as DesktopCanvas)

Plugin Version: 23.2, Platform: macOS, Targets: Desktop only.

Function: Removes a canvas from the list.

Notes: Please remove canvas controls in Window.Close event.

The plugin tries to remove all when a window closes automatically.

See also:

- 9.2.7 RemoveCanvas(c as Canvas)

299

9.2.9 Properties

9.2.10 CanvasCount as Integer

Plugin Version: 18.2, Platform: macOS, Targets: Desktop only.

Function: Number of Canvas controls registered.

Notes: (Read only property)

9.2.11 Events

9.2.12 beginGestureWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 18.2, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a touch gesture.

Notes: e: An event object representing the gesture beginning.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.2.13 endGestureWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 18.2, Platform: macOS, Targets: .

Function: Informs the receiver that the user has ended a touch gesture.

Notes: e: An event object representing the gesture end.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.2.14 magnifyWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 18.2, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a pinch gesture.

Notes: e: An event object representing the magnify gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.2.15 rotateWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 18.2, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a rotation gesture.

Notes: e: An event object representing the rotate gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.2.16 scrollWheel(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 19.2, Platform: macOS, Targets: .

Function: Informs the receiver that the mouse,Äôs scroll wheel has moved.

Notes: The default implementation simply passes this message to the next responder.

e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.

Return true if you handled this event.

9.2.17 smartMagnifyWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 19.2, Platform: macOS, Targets: .

Function: Informs the receiver that the user performed a smart zoom gesture.

Notes: The smart zoom gesture is a two-finger double tap on trackpads. In response to this event, you

should intelligently magnify the content.

e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.
Return true if you handled this event.

9.2.18 `swipeWithEvent(can as Variant, e as NSEventMBS)` as boolean

Plugin Version: 18.2, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a swipe gesture.

Notes: e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.2.19 `touchesBeganWithEvent(can as Variant, e as NSEventMBS)` as boolean

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Informs the receiver that new set of touches has been recognized.

Notes: can: The Canvas or DesktopCanvas control.

e: An event object representing the beginning of a touch.

The system sends the event to the view under the touch in the key window.

This isn't always the point of contact with the touch device. A touch that transitions from resting to active may be part of a `touchesBeganWithEvent` set.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.2.20 `touchesCancelledWithEvent(can as Variant, e as NSEventMBS)` as boolean

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Informs the receiver that tracking of touches has been cancelled for any reason.

Notes: can: The Canvas or DesktopCanvas control.
e: An event object representing the cancellation of a touch event.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.
Return true if you handled this event.

9.2.21 touchesEndedWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Returns that a set of touches have been removed.
Notes: can: The Canvas or DesktopCanvas control.
e: An event object representing the ending of a touch event.

The system sends the event to the view under the touch in the key window.

This isn't always the point of removal with the touch device. A touch that transitions from active to resting may be part of an touchesEndedWithEvent set.

Available in Mac OS X v10.6 and later.
Return true if you handled this event.

9.2.22 touchesMovedWithEvent(can as Variant, e as NSEventMBS) as boolean

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Informs the receiver that one or more touches has moved.
Notes: can: The Canvas or DesktopCanvas control.
e: An event object representing a touch movement.

The system sends the to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.
Return true if you handled this event.

9.3 class Checkbox

9.3.1 class Checkbox

Plugin Version: 2.9, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.3.2 Methods

9.3.3 NSButtonMBS as NSButtonMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox CheckBox1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.4 class ComboBox

9.4.1 class ComboBox

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The built in ComboBox class in Xojo.

9.4.2 Methods

9.4.3 NSComboBoxMBS as NSComboBoxMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSComboBoxMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.5 class CustomNSScrollerMBS

9.5.1 class CustomNSScrollerMBS

Plugin Version: 12.1, Platform: macOS, Targets: Desktop only.

Function: The class for a custom scroller.

Notes: Subclass of the NSScrollerMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 12.4pr4](#)
- [Custom Scrollbars for Real Studio Cocoa Apps](#)
- [MBS Real Studio Plugins, version 12.1pr6](#)

9.5.2 Methods

9.5.3 Constructor

Plugin Version: 12.1, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSScroller object.

See also:

- 9.5.4 Constructor(Handle as Integer) 306
- 9.5.5 Constructor(left as Double, top as Double, width as Double, height as Double) 306

9.5.4 Constructor(Handle as Integer)

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: The constructor.

See also:

- 9.5.3 Constructor 306
- 9.5.5 Constructor(left as Double, top as Double, width as Double, height as Double) 306

9.5.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 12.1, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSScroller object.

See also:

9.5. CLASS CUSTOMNSSCROLLERMBS	307
• 9.5.3 Constructor	306
• 9.5.4 Constructor(Handle as Integer)	306

9.5.6 Destructor

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The destructor.

9.5.7 Events

9.5.8 acceptsFirstMouse(e as NSEventMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Overridden by subclasses to return true if the receiver should be sent a mouseDown event for an initial mouse-down event, false if not.

Notes: The receiver can either return a value unconditionally or use the location of event e to determine whether or not it wants the event. The default implementation ignores the event and returns false.

Implement this event in a subclass to allow instances to respond to click-through. This allows the user to click on a view in an inactive window, activating the view with one click, instead of clicking first to make the window active and then clicking the view. Most view objects refuse a click-through attempt, so the event simply activates the window. Many control objects, however, such as instances of NSButton and NSSlider, do accept them, so the user can immediately manipulate the control without having to release the mouse button.

9.5.9 acceptsFirstResponder as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Whether to accept first responder.

Notes: Return true if your control can have the focus and false if not.

9.5.10 becomeFirstResponder as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Called when the object gets focus.

Notes: Return true to accept.

9.5.11 beginGestureWithEvent(e as NSEventMBS) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a touch gesture.

Notes: e: An event object representing the gesture beginning.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.5.12 canBecomeKeyView as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Returns whether the receiver can become key view.

Notes: Returns true if the receiver can become key view, false otherwise.

9.5.13 Close

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: The event called when the custom view is destroyed.

9.5.14 concludeDragOperation(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging operation is complete, signaling the receiver to perform any necessary clean-up.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

For this method to be invoked, the previous performDragOperation must have returned true.

The destination implements this method to perform any tidying up that it needs to do, such as updating its visual representation now that it has incorporated the dragged data. This message is the last message sent from sender to the destination during a dragging session.

If the sender object's animatesToDestination property was set to true in prepareForDragOperation, then the drag image is still visible. At this point you should draw the final visual representation in the view. When this method returns, the drag image is removed from the screen. If your final visual representation matches the visual representation in the drag, this is a seamless transition.

9.5.15 draggingEnded(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Implement this event to be notified when a drag operation ends in some other destination.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

This method might be used by a destination doing auto-expansion in order to collapse any auto-expands.

9.5.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image enters destination bounds or frame; delegate returns dragging operation to perform.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in NSDragOperation in the NSDraggingInfo reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous draggingEntered message.

Invoked when a dragged image enters the destination but only if the destination has registered for the pasteboard data type involved in the drag operation. Specifically, this method is invoked when the mouse pointer enters the destination's bounds rectangle (if it is a view object) or its frame rectangle (if it is a window object).

This method must return a value that indicates which dragging operation the destination will perform when the image is released. In deciding which dragging operation to return, the method should evaluate the overlap between both the dragging operations allowed by the source (obtained from sender with the draggingSourceOperationMask method) and the dragging operations and pasteboard data types the destination itself supports.

If none of the operations is appropriate, this method should return NSDragOperationNone (this is the default response if the method is not implemented by the destination). A destination will still receive draggingUpdated and draggingExited even if NSDragOperationNone is returned by this method.

9.5.17 draggingExited(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image exits the destination’s bounds rectangle (in the case of a view object) or its frame rectangle (in the case of a window object).

Notes: sender: The object sending the message; use it to get details about the dragging operation.

9.5.18 `draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging session has completed.

Notes: session: The dragging session.

screenPoint: The point where the drag ended, in screen coordinates.

operation: The drag operation. See constants for drag operation types.

Available in OS X v10.7 and later.

9.5.19 `draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the drag moves on the screen.

Notes: session: The dragging session.

screenPoint: The point where the drag moved to, in screen coordinates.

Available in OS X v10.7 and later.

9.5.20 `draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Declares the types of operations the source allows to be performed. (required)

Notes: session: The dragging session.

context: The dragging context. See `NSDraggingContext` constants for the supported values.

Return the appropriate dragging operation as defined in constants.

In the future Apple may provide more specific "within" values in the future. To account for this, for unrecognized localities, return the operation mask for the most specific context that you are concerned with.

9.5.21 draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the drag will begin.

Notes: session: The dragging session.

screenPoint: The point where the drag will begin, in screen coordinates.

Available in OS X v10.7 and later.

9.5.22 draggingUpdated(sender as NSDraggingInfoMBS) as Integer

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked periodically as the image is held within the destination area, allowing modification of the dragging operation or mouse-pointer position.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in NSDragOperation in the NSDraggingInfo reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous draggingEntered message.

For this to be invoked, the destination must have registered for the pasteboard data type involved in the drag operation. The messages continue until the image is either released or dragged out of the window or view.

This method provides the destination with an opportunity to modify the dragging operation depending on the position of the mouse pointer inside of the destination view or window object. For example, you may have several graphics or areas of text contained within the same view and wish to tailor the dragging operation, or to ignore the drag event completely, depending upon which object is underneath the mouse pointer at the time when the user releases the dragged image and the performDragOperation method is invoked.

You typically examine the contents of the pasteboard in the draggingEntered method, where this examination is performed only once, rather than in the draggingUpdated method, which is invoked multiple times.

Only one destination at a time receives a sequence of draggingUpdated messages. If the mouse pointer is within the bounds of two overlapping views that are both valid destinations, the uppermost view receives these messages until the image is either released or dragged out.

9.5.23 drawArrow(g as NSGraphicsMBS, Arrow as Integer, highlight as boolean)

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Called when arrow need to be drawn.

Notes: Draw the scroll button indicated by arrow, which is either NSScrollerIncrementArrow (the down or right scroll button) or NSScrollerDecrementArrow (up or left).

If flag is true, the button is drawn highlighted; otherwise it's drawn normally. You should never need to invoke this method directly, but may wish to override it to customize the appearance of scroll buttons.

9.5.24 drawKnob(g as NSGraphicsMBS)

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Draw the knob.

9.5.25 drawKnobSlotInRect(g as NSGraphicsMBS, slotRect as NSRectMBS, highlight as boolean)

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Draw the knob slot in the rectangle.

9.5.26 drawParts(g as NSGraphicsMBS)

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Called when parts need to be drawn.

9.5.27 endGestureWithEvent(e as NSEventMBS) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has ended a touch gesture.

Notes: e: An event object representing the gesture end.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.5.28 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Returns whether the modifier keys will be ignored for this dragging session.

Notes: session: The dragging session.

Return true if the modifier keys will be ignored, false otherwise.
Available in OS X v10.7 and later.

9.5.29 isOpaque as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Whether this view is opaque.

9.5.30 keyDown(e as NSEventMBS) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

9.5.31 keyUp(e as NSEventMBS) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

9.5.32 magnifyWithEvent(e as NSEventMBS) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a pinch gesture.

Notes: e: An event object representing the magnify gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.5.33 `menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS)` as `NSMenuMBS`

Plugin Version: 12.3, Platform: macOS, Targets: .

Function: Overridden by subclasses to return a context-sensitive pop-up menu for a given mouse-down event.

Notes: `theEvent`: An object representing a mouse-down event.

`defaultMenu`: The menu as constructed by super class.

The receiver can use information in the mouse event, such as its location over a particular element of the receiver, to determine what kind of menu to return. For example, a text object might display a text-editing menu when the cursor lies over text and a menu for changing graphics attributes when the cursor lies over an embedded image.

The default implementation returns the default menu.

9.5.34 `mouseDown(e as NSEventMBS, x as Double, y as Double)` as `boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.35 `mouseDownCanMoveWindow` as `boolean`

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: This event is called so you can decide what happens with mouse down.

Notes: Return true if you do not need to handle a mouse down and it can pass through to superviews; False if you need to handle the mouse down.

This allows iApp-type applications to determine the region by which a window can be moved. By default, this method returns false if the view is opaque; otherwise, it returns true. Subclasses can override this method to return a different value.

9.5.36 `mouseDragged(e as NSEventMBS, x as Double, y as Double)` as `boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.37 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.38 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.39 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.40 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.41 Open

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: The event called when the custom NSView is created.

9.5.42 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.5.43 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.5.44 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.5.45 performDragOperation(sender as NSDraggingInfoMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked after the released image has been removed from the screen, signaling the receiver to import the pasteboard data.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return if the destination accepts the data, it returns true; otherwise it returns false. The default is to return false.

For this method to be invoked, the previous `prepareForDragOperation` message must have returned true. The destination should implement this method to do the real work of importing the pasteboard data represented by the image.

If the sender object's `animatesToDestination` was set to true in `prepareForDragOperation`, then setup any

animation to arrange space for the drag items to animate to. Also at this time, enumerate through the dragging items to set their destination frames and destination images.

9.5.46 prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the image is released, allowing the receiver to agree to or refuse drag operation.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Returns true if the receiver agrees to perform the drag operation and false if not.

This method is invoked only if the most recent draggingEntered or draggingUpdated message returned an acceptable drag-operation value.

If you want the drag items to animate from their current location on screen to their final location in your view, set the sender object's animatesToDestination property to true in your implementation of this method.

9.5.47 pressureChange(e as NSEventMBS) as boolean

Plugin Version: 15.1, Platform: macOS, Targets: .

Function: Informs the current object that a pressure change occurred on a system that supports pressure sensitivity.

Notes: This method is invoked automatically in response to user actions. event is the event that initiated the change in pressure.

Available in OS X v10.10.3 and later.

9.5.48 resignFirstResponder as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Focus is going away.

Notes: Return true to accept.

9.5.49 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.50 `rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.51 `rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.5.52 `rotateWithEvent(e as NSEventMBS) as boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a rotation gesture.

Notes: e: An event object representing the rotate gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.5.53 `scrollWheel(e as NSEventMBS) as boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Informs the subclass that the mouse's scroll wheel has moved.

Notes: e: An object encapsulating information about the wheel-scrolling event.

The default implementation simply passes this message to the next responder.

Return true to not pass the event.

9.5.54 `swipeWithEvent(e as NSEventMBS) as boolean`

Plugin Version: 12.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a swipe gesture.

Notes: e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.
Available in Mac OS X v10.6 and later.
Return true if you handled this event.

9.5.55 updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging images should be changed.

Notes: sender: The object sending the message; use this object to get details about the dragging operation.

While a destination may change the dragging images at any time, it is recommended to wait until this method is called before updating the dragging images.

This allows the system to delay changing the dragging images until it is likely that the user will drop on this destination. Otherwise, the dragging images will change too often during the drag which would be distracting to the user.

9.5.56 viewDidMoveToWindow

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Informs the receiver that it has been added to a new view hierarchy.

Notes: The default implementation does nothing; subclasses can implement this event to perform whatever actions are necessary.

window may return nil when this method is invoked, indicating that the receiver does not currently reside in any window. This occurs when the receiver has just been removed from its superview or when the receiver has just been added to a superview that does not itself have a window. Overrides of this method may choose to ignore such cases if they are not of interest.

9.5.57 wantsPeriodicDraggingUpdates as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Asks the destination object whether it wants to receive periodic draggingUpdated messages.

Notes: Return true if the destination wants to receive periodic draggingUpdated messages, false otherwise.

If the destination returns false, these messages are sent only when the mouse moves or a modifier flag changes. Otherwise the destination gets the default behavior, where it receives periodic dragging-updated messages

even if nothing changes.

9.6 class CustomNSTokenFieldMBS

9.6.1 class CustomNSTokenFieldMBS

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The class for a custom NSTextField.

Notes: Subclass of the NSTextFieldMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 12.5pr2](#)

9.6.2 Methods

9.6.3 Constructor

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The dummy constructor doing nothing.

See also:

- 9.6.4 Constructor(Handle as Integer) 321
- 9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double) 321

9.6.4 Constructor(Handle as Integer)

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: The constructor.

See also:

- 9.6.3 Constructor 321
- 9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double) 321

9.6.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSTextField object.

See also:

- 9.6.3 Constructor 321
- 9.6.4 Constructor(Handle as Integer) 321

9.6.6 Destructor

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The destructor.

9.6.7 Events

9.6.8 `acceptsFirstMouse(e as NSEventMBS)` as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Overridden by subclasses to return true if the receiver should be sent a `mouseDown` event for an initial mouse-down event, false if not.

Notes: The receiver can either return a value unconditionally or use the location of event `e` to determine whether or not it wants the event. The default implementation ignores the event and returns false.

Implement this event in a subclass to allow instances to respond to click-through. This allows the user to click on a view in an inactive window, activating the view with one click, instead of clicking first to make the window active and then clicking the view. Most view objects refuse a click-through attempt, so the event simply activates the window. Many control objects, however, such as instances of `NSButton` and `NSSlider`, do accept them, so the user can immediately manipulate the control without having to release the mouse button.

9.6.9 `acceptsFirstResponder` as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Whether to accept first responder.

Notes: Return true if your control can have the focus and false if not.

9.6.10 `becomeFirstResponder` as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Called when the object gets focus.

Notes: Return true to accept.

9.6.11 `beginGestureWithEvent(e as NSEventMBS)` as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a touch gesture.

Notes: e: An event object representing the gesture beginning.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.6.12 canBecomeKeyView as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Returns whether the receiver can become key view.

Notes: Returns true if the receiver can become key view, false otherwise.

9.6.13 Close

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: The event called when the custom view is destroyed.

9.6.14 concludeDragOperation(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging operation is complete, signaling the receiver to perform any necessary clean-up.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

For this method to be invoked, the previous performDragOperation must have returned true.

The destination implements this method to perform any tidying up that it needs to do, such as updating its visual representation now that it has incorporated the dragged data. This message is the last message sent from sender to the destination during a dragging session.

If the sender object's animatesToDestination property was set to true in prepareForDragOperation, then the drag image is still visible. At this point you should draw the final visual representation in the view. When this method returns, the drag image is removed from the screen. If your final visual representation matches the visual representation in the drag, this is a seamless transition.

9.6.15 draggingEnded(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Implement this event to be notified when a drag operation ends in some other destination.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

This method might be used by a destination doing auto-expansion in order to collapse any auto-expands.

9.6.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image enters destination bounds or frame; delegate returns dragging operation to perform.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in NSDragOperation in the NSDraggingInfo reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous draggingEntered message.

Invoked when a dragged image enters the destination but only if the destination has registered for the pasteboard data type involved in the drag operation. Specifically, this method is invoked when the mouse pointer enters the destination's bounds rectangle (if it is a view object) or its frame rectangle (if it is a window object).

This method must return a value that indicates which dragging operation the destination will perform when the image is released. In deciding which dragging operation to return, the method should evaluate the overlap between both the dragging operations allowed by the source (obtained from sender with the draggingSourceOperationMask method) and the dragging operations and pasteboard data types the destination itself supports.

If none of the operations is appropriate, this method should return NSDragOperationNone (this is the default response if the method is not implemented by the destination). A destination will still receive draggingUpdated and draggingExited even if NSDragOperationNone is returned by this method.

9.6.17 draggingExited(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image exits the destination's bounds rectangle (in the case of a view object) or its frame rectangle (in the case of a window object).

Notes: sender: The object sending the message; use it to get details about the dragging operation.

9.6.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging session has completed.

Notes: session: The dragging session.

screenPoint: The point where the drag ended, in screen coordinates.

operation: The drag operation. See constants for drag operation types.

Available in OS X v10.7 and later.

9.6.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the drag moves on the screen.

Notes: session: The dragging session.

screenPoint: The point where the drag moved to, in screen coordinates.

Available in OS X v10.7 and later.

9.6.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Declares the types of operations the source allows to be performed. (required)

Notes: session: The dragging session.

context: The dragging context. See NSDraggingContext constants for the supported values.

Return the appropriate dragging operation as defined in constants.

In the future Apple may provide more specific "within" values in the future. To account for this, for unrecognized localities, return the operation mask for the most specific context that you are concerned with.

9.6.21 `draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the drag will begin.

Notes: `session`: The dragging session.

`screenPoint`: The point where the drag will begin, in screen coordinates.

Available in OS X v10.7 and later.

9.6.22 `draggingUpdated(sender as NSDraggingInfoMBS) as Integer`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked periodically as the image is held within the destination area, allowing modification of the dragging operation or mouse-pointer position.

Notes: `sender`: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in `NSDragOperation` in the `NSDraggingInfo` reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous `draggingEntered` message.

For this to be invoked, the destination must have registered for the pasteboard data type involved in the drag operation. The messages continue until the image is either released or dragged out of the window or view.

This method provides the destination with an opportunity to modify the dragging operation depending on the position of the mouse pointer inside of the destination view or window object. For example, you may have several graphics or areas of text contained within the same view and wish to tailor the dragging operation, or to ignore the drag event completely, depending upon which object is underneath the mouse pointer at the time when the user releases the dragged image and the `performDragOperation` method is invoked.

You typically examine the contents of the pasteboard in the `draggingEntered` method, where this examination is performed only once, rather than in the `draggingUpdated` method, which is invoked multiple times.

Only one destination at a time receives a sequence of `draggingUpdated` messages. If the mouse pointer is within the bounds of two overlapping views that are both valid destinations, the uppermost view receives these messages until the image is either released or dragged out.

9.6.23 endGestureWithEvent(e as NSEventMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has ended a touch gesture.

Notes: e: An event object representing the gesture end.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.6.24 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Returns whether the modifier keys will be ignored for this dragging session.

Notes: session: The dragging session.

Return true if the modifier keys will be ignored, false otherwise.

Available in OS X v10.7 and later.

9.6.25 isOpaque as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Whether this view is opaque.

9.6.26 keyDown(e as NSEventMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

Please return true in `becomeFirstResponder` and `acceptsFirstResponder`, so your `nview` can become first responder and receive key events.

9.6.27 keyUp(e as NSEventMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

Please return true in `becomeFirstResponder` and `acceptsFirstResponder`, so your `nview` can become first responder and receive key events.

9.6.28 `magnifyWithEvent(e as NSEventMBS)` as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a pinch gesture.

Notes: `e`: An event object representing the magnify gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.6.29 `menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS)` as NSMenuMBS

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Overridden by subclasses to return a context-sensitive pop-up menu for a given mouse-down event.

Notes: `theEvent`: An object representing a mouse-down event.

`defaultMenu`: The menu as constructed by super class.

The receiver can use information in the mouse event, such as its location over a particular element of the receiver, to determine what kind of menu to return. For example, a text object might display a text-editing menu when the cursor lies over text and a menu for changing graphics attributes when the cursor lies over an embedded image.

The default implementation returns the default menu.

9.6.30 `mouseDown(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.31 `mouseDownCanMoveWindow` as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: This event is called so you can decide what happens with mouse down.

Notes: Return true if you do not need to handle a mouse down and it can pass through to superviews; False if you need to handle the mouse down.

This allows iApp-type applications to determine the region by which a window can be moved. By default, this method returns false if the view is opaque; otherwise, it returns true. Subclasses can override this method to return a different value.

9.6.32 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.33 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.34 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.35 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.36 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.37 Open

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: The event called when the custom NSView is created.

9.6.38 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.6.39 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.6.40 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.6.41 performDragOperation(sender as NSDraggingInfoMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked after the released image has been removed from the screen, signaling the receiver to import the pasteboard data.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return if the destination accepts the data, it returns true; otherwise it returns false. The default is to return false.

For this method to be invoked, the previous `prepareForDragOperation` message must have returned true. The destination should implement this method to do the real work of importing the pasteboard data represented by the image.

If the sender object's `animatesToDestination` was set to true in `prepareForDragOperation`, then setup any animation to arrange space for the drag items to animate to. Also at this time, enumerate through the dragging items to set their destination frames and destination images.

9.6.42 `prepareForDragOperation(sender as NSDraggingInfoMBS)` as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the image is released, allowing the receiver to agree to or refuse drag operation.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Returns true if the receiver agrees to perform the drag operation and false if not.

This method is invoked only if the most recent `draggingEntered` or `draggingUpdated` message returned an acceptable drag-operation value.

If you want the drag items to animate from their current location on screen to their final location in your view, set the sender object's `animatesToDestination` property to true in your implementation of this method.

9.6.43 `pressureChange(e as NSEventMBS)` as boolean

Plugin Version: 15.1, Platform: macOS, Targets: .

Function: Informs the current object that a pressure change occurred on a system that supports pressure sensitivity.

Notes: This method is invoked automatically in response to user actions. `event` is the event that initiated the change in pressure.

Available in OS X v10.10.3 and later.

9.6.44 `resignFirstResponder` as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Focus is going away.

Notes: Return true to accept.

9.6.45 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.46 rightMouseDownDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.47 rightMouseDownUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.6.48 rotateWithEvent(e as NSEventMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a rotation gesture.

Notes: e: An event object representing the rotate gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.6.49 scrollWheel(e as NSEventMBS) as boolean

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Informs the subclass that the mouse's scroll wheel has moved.

Notes: e: An object encapsulating information about the wheel-scrolling event.

The default implementation simply passes this message to the next responder. Return true to not pass the event.

9.6.50 swipeWithEvent(e as NSEventMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a swipe gesture.

Notes: e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.6.51 updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging images should be changed.

Notes: sender: The object sending the message; use this object to get details about the dragging operation.

While a destination may change the dragging images at any time, it is recommended to wait until this method is called before updating the dragging images.

This allows the system to delay changing the dragging images until it is likely that the user will drop on this destination. Otherwise, the dragging images will change too often during the drag which would be distracting to the user.

9.6.52 viewDidMoveToWindow

Plugin Version: 12.5, Platform: macOS, Targets: .

Function: Informs the receiver that it has been added to a new view hierarchy.

Notes: The default implementation does nothing; subclasses can implement this event to perform whatever actions are necessary.

window may return nil when this method is invoked, indicating that the receiver does not currently reside in any window. This occurs when the receiver has just been removed from its superview or when the receiver has just been added to a superview that does not itself have a window. Overrides of this method may choose to ignore such cases if they are not of interest.

9.6.53 `wantsPeriodicDraggingUpdates` as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Asks the destination object whether it wants to receive periodic draggingUpdated messages.

Notes: Return true if the destination wants to receive periodic draggingUpdated messages, false otherwise.

If the destination returns false, these messages are sent only when the mouse moves or a modifier flag changes. Otherwise the destination gets the default behavior, where it receives periodic dragging-updated messages even if nothing changes.

9.7 class CustomNSViewMBS

9.7.1 class CustomNSViewMBS

Plugin Version: 7.7, Platform: macOS, Targets: Desktop only.

Function: The class for a custom NSView.

Notes: Subclass of the NSViewMBS class.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 16.5pr2](#)
- [MBS Real Studio Plugins, version 12.3pr3](#)
- [MBS Real Studio Plugins, version 12.3pr2](#)
- [MBS Real Studio Plugins, version 12.0pr7](#)
- [MBS Real Studio Plugins, version 11.3pr14](#)
- [MBS Real Studio Plugins, version 11.3pr9](#)
- [Gestures on Mac OS X](#)
- [MBS Real Studio Plugins, version 11.2pr6](#)
- [MBS Plugins 11.1 Release notes](#)
- [MBS Plugins 10.3 Release Notes](#)

9.7.2 Methods

9.7.3 Constructor

Plugin Version: 7.7, Platform: macOS, Targets: Desktop only.

Function: The dummy constructor doing nothing.

See also:

- [9.7.4 Constructor\(Handle as Integer\)](#) 335
- [9.7.5 Constructor\(left as Double, top as Double, width as Double, height as Double\)](#) 336

9.7.4 Constructor(Handle as Integer)

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given CustomNSView handle.

Notes: The handle is casted to a CustomNSView and the plugin retains this handle.

See also:

- 9.7.3 Constructor 335
- 9.7.5 Constructor(left as Double, top as Double, width as Double, height as Double) 336

9.7.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 7.7, Platform: macOS, Targets: Desktop only.

Function: The constructor for a new custom NSView object.
See also:

- 9.7.3 Constructor 335
- 9.7.4 Constructor(Handle as Integer) 335

9.7.6 Destructor

Plugin Version: 12.5, Platform: macOS, Targets: Desktop only.

Function: The destructor.

9.7.7 Events

9.7.8 acceptsFirstResponder(e as NSEventMBS) as boolean

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Overridden by subclasses to return true if the receiver should be sent a mouseDown event for an initial mouse-down event, false if not.

Notes: The receiver can either return a value unconditionally or use the location of event e to determine whether or not it wants the event. The default implementation ignores the event and returns false.

Implement this event in a subclass to allow instances to respond to click-through. This allows the user to click on a view in an inactive window, activating the view with one click, instead of clicking first to make the window active and then clicking the view. Most view objects refuse a click-through attempt, so the event simply activates the window. Many control objects, however, such as instances of NSButton and NSSlider, do accept them, so the user can immediately manipulate the control without having to release the mouse button.

9.7.9 acceptsFirstResponder as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: Whether to accept first responder.

Notes: Return true if your control can have the focus and false if not.

9.7.10 becomeFirstResponder as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: Called when the object gets focus.

Notes: Return true to accept.

9.7.11 beginGestureWithEvent(e as NSEventMBS) as boolean

Plugin Version: 10.3, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a touch gesture.

Notes: e: An event object representing the gesture beginning.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.7.12 canBecomeKeyView as boolean

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Returns whether the receiver can become key view.

Notes: Returns true if the receiver can become key view, false otherwise.

9.7.13 Close

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: The event called when the custom view is destroyed.

9.7.14 concludeDragOperation(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging operation is complete, signaling the receiver to perform any necessary clean-up.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

For this method to be invoked, the previous `performDragOperation` must have returned true.

The destination implements this method to perform any tidying up that it needs to do, such as updating its visual representation now that it has incorporated the dragged data. This message is the last message sent from sender to the destination during a dragging session.

If the sender object's `animatesToDestination` property was set to true in `prepareForDragOperation`, then the drag image is still visible. At this point you should draw the final visual representation in the view. When this method returns, the drag image is removed from the screen. If your final visual representation matches the visual representation in the drag, this is a seamless transition.

9.7.15 `draggingEnded(sender as NSDraggingInfoMBS)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Implement this event to be notified when a drag operation ends in some other destination.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

This method might be used by a destination doing auto-expansion in order to collapse any auto-expands.

9.7.16 `draggingEntered(sender as NSDraggingInfoMBS) as Integer`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image enters destination bounds or frame; delegate returns dragging operation to perform.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in `NSDragOperation` in the `NSDraggingInfo` reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous `draggingEntered` message.

Invoked when a dragged image enters the destination but only if the destination has registered for the paste-board data type involved in the drag operation. Specifically, this method is invoked when the mouse pointer enters the destination's bounds rectangle (if it is a view object) or its frame rectangle (if it is a window object).

This method must return a value that indicates which dragging operation the destination will perform when the image is released. In deciding which dragging operation to return, the method should evaluate the

overlap between both the dragging operations allowed by the source (obtained from sender with the `draggingSourceOperationMask` method) and the dragging operations and pasteboard data types the destination itself supports.

If none of the operations is appropriate, this method should return `NSDragOperationNone` (this is the default response if the method is not implemented by the destination). A destination will still receive `draggingUpdated` and `draggingExited` even if `NSDragOperationNone` is returned by this method.

9.7.17 `draggingExited(sender as NSDraggingInfoMBS)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image exits the destination's bounds rectangle (in the case of a view object) or its frame rectangle (in the case of a window object).

Notes: sender: The object sending the message; use it to get details about the dragging operation.

9.7.18 `draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging session has completed.

Notes: session: The dragging session.

screenPoint: The point where the drag ended, in screen coordinates.

operation: The drag operation. See constants for drag operation types.

Available in OS X v10.7 and later.

9.7.19 `draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the drag moves on the screen.

Notes: session: The dragging session.

screenPoint: The point where the drag moved to, in screen coordinates.

Available in OS X v10.7 and later.

9.7.20 `draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Declares the types of operations the source allows to be performed. (required)

Notes: session: The dragging session.

context: The dragging context. See `NSDraggingContext` constants for the supported values.

Return the appropriate dragging operation as defined in constants.

In the future Apple may provide more specific "within" values in the future. To account for this, for unrecognized localities, return the operation mask for the most specific context that you are concerned with.

9.7.21 `draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the drag will begin.

Notes: session: The dragging session.

screenPoint: The point where the drag will begin, in screen coordinates.

Available in OS X v10.7 and later.

9.7.22 `draggingUpdated(sender as NSDraggingInfoMBS) as Integer`

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked periodically as the image is held within the destination area, allowing modification of the dragging operation or mouse-pointer position.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in `NSDragOperation` in the `NSDraggingInfo` reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous `draggingEntered` message.

For this to be invoked, the destination must have registered for the pasteboard data type involved in the drag operation. The messages continue until the image is either released or dragged out of the window or view.

This method provides the destination with an opportunity to modify the dragging operation depending on

the position of the mouse pointer inside of the destination view or window object. For example, you may have several graphics or areas of text contained within the same view and wish to tailor the dragging operation, or to ignore the drag event completely, depending upon which object is underneath the mouse pointer at the time when the user releases the dragged image and the `performDragOperation` method is invoked.

You typically examine the contents of the pasteboard in the `draggingEntered` method, where this examination is performed only once, rather than in the `draggingUpdated` method, which is invoked multiple times.

Only one destination at a time receives a sequence of `draggingUpdated` messages. If the mouse pointer is within the bounds of two overlapping views that are both valid destinations, the uppermost view receives these messages until the image is either released or dragged out.

9.7.23 `drawFocusRingMask(g as NSGraphicsMBS)` as boolean

Plugin Version: 12.3, Platform: macOS, Targets: .

Function: Draw the focus ring mask for the view.

Notes: If false is returned, the default method from `NSView` class runs.

This method provides the shape of the focus ring mask by drawing the focus ring mask. An implementation of this method should draw in the view's interior (bounds) coordinate space, that the focus ring style has been set (it will be set it to `NSFocusRingOnly` to capture the focus ring itself), and that the fill and stroke colors have been set to an arbitrary fully opaque color.

Subclasses that find the default behavior insufficient should only draw the focus ring shape.

The `NSView` default implementation of this method simply fills `self.bounds`. Available in Mac OS X v10.7 and later.

Please use `NSGraphicsMBS` class for drawing.

9.7.24 `DrawRect(g as NSGraphicsMBS, left as Double, top as Double, width as Double, height as Double)`

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: The paint event with the rectangle which needs to be redrawn.

9.7.25 endGestureWithEvent(e as NSEventMBS) as boolean

Plugin Version: 10.3, Platform: macOS, Targets: .

Function: Informs the receiver that the user has ended a touch gesture.

Notes: e: An event object representing the gesture end.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.7.26 focusRingMaskBounds as NSRectMBS

Plugin Version: 12.3, Platform: macOS, Targets: .

Function: Returns the focus ring mask bounds.

Notes: Return nil to run default NSView method.

Return a rectangle containing the mask in the view's interior (bounds) coordinate space.

The mask bounds allows the focus ring's overall size and position to be determined before it is drawn.

Subclasses must override this method if they require the display of a focus ring.

The NSView default implementation of this method simply returns NSRectMBS.Zero.

Note: The information provided by focusRingMaskBounds will enable Accessibility to identify selected subelements for zoom tracking, so it is important that this method provide a reasonably tight bounding box and that noteFocusRingMaskChanged is invoked as described.

9.7.27 ignoreModifierKeysForDraggingSession(session as NSDraggingSessionMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Returns whether the modifier keys will be ignored for this dragging session.

Notes: session: The dragging session.

Return true if the modifier keys will be ignored, false otherwise.

Available in OS X v10.7 and later.

9.7.28 isFlipped as Boolean

Plugin Version: 16.5, Platform: macOS, Targets: .

Function: The event to return a boolean value indicating whether the view uses a flipped coordinate system.

Notes: The default value of this property is false, which results in a non-flipped coordinate system. In a non-flipped coordinate system, the origin is in the lower-left corner of the view and positive y-values extend upward. In a flipped coordinate system, the origin is in the upper-left corner of the view and y-values extend downward. X-values always extend to the right.

If you want your view to use a flipped coordinate system, override this property and return true.

9.7.29 isOpaque as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: Whether this view is opaque.

9.7.30 keyDown(e as NSEventMBS) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

Please return true in `becomeFirstResponder` and `acceptsFirstResponder`, so your `nview` can become first responder and receive key events.

9.7.31 keyUp(e as NSEventMBS) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

Please return true in `becomeFirstResponder` and `acceptsFirstResponder`, so your `nview` can become first responder and receive key events.

9.7.32 magnifyWithEvent(e as NSEventMBS) as boolean

Plugin Version: 10.3, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a pinch gesture.

Notes: `e`: An event object representing the magnify gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.7.33 `menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS`

Plugin Version: 12.3, Platform: macOS, Targets: .

Function: Overridden by subclasses to return a context-sensitive pop-up menu for a given mouse-down event.

Notes: `theEvent`: An object representing a mouse-down event.

`defaultMenu`: The menu as constructed by super class.

The receiver can use information in the mouse event, such as its location over a particular element of the receiver, to determine what kind of menu to return. For example, a text object might display a text-editing menu when the cursor lies over text and a menu for changing graphics attributes when the cursor lies over an embedded image.

The default implementation returns the default menu.

9.7.34 `mouseDown(e as NSEventMBS, x as Double, y as Double) as boolean`

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.35 `mouseDownCanMoveWindow` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: This event is called so you can decide what happens with mouse down.

Notes: Return true if you do not need to handle a mouse down and it can pass through to superviews; False if you need to handle the mouse down.

This allows iApp-type applications to determine the region by which a window can be moved. By default, this method returns false if the view is opaque; otherwise, it returns true. Subclasses can override this method to return a different value.

9.7.36 mouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.37 mouseEntered(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.38 mouseExited(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.39 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.40 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.41 Open

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: The event called when the custom `NSView` is created.

9.7.42 `otherMouseDown(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.7.43 `otherMouseDragged(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.7.44 `otherMouseUp(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.7.45 `performDragOperation(sender as NSDraggingInfoMBS)` as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked after the released image has been removed from the screen, signaling the receiver to import the pasteboard data.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return if the destination accepts the data, it returns true; otherwise it returns false. The default is to return false.

For this method to be invoked, the previous `prepareForDragOperation` message must have returned true. The destination should implement this method to do the real work of importing the pasteboard data repre-

sent by the image.

If the sender object's `animatesToDestination` was set to true in `prepareForDragOperation`, then setup any animation to arrange space for the drag items to animate to. Also at this time, enumerate through the dragging items to set their destination frames and destination images.

9.7.46 `prepareForDragOperation(sender as NSDraggingInfoMBS)` as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the image is released, allowing the receiver to agree to or refuse drag operation.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Returns true if the receiver agrees to perform the drag operation and false if not.

This method is invoked only if the most recent `draggingEntered` or `draggingUpdated` message returned an acceptable drag-operation value.

If you want the drag items to animate from their current location on screen to their final location in your view, set the sender object's `animatesToDestination` property to true in your implementation of this method.

9.7.47 `pressureChange(e as NSEventMBS)` as boolean

Plugin Version: 15.1, Platform: macOS, Targets: .

Function: Informs the current object that a pressure change occurred on a system that supports pressure sensitivity.

Notes: This method is invoked automatically in response to user actions. `event` is the event that initiated the change in pressure.

Available in OS X v10.10.3 and later.

9.7.48 `resignFirstResponder` as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: Focus is going away.

Notes: Return true to accept.

9.7.49 `rightMouseDown(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.50 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.51 rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 7.7, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.7.52 rotateWithEvent(e as NSEventMBS) as boolean

Plugin Version: 10.3, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a rotation gesture.

Notes: e: An event object representing the rotate gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.7.53 scrollWheel(e as NSEventMBS) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Informs the subclass that the mouse's scroll wheel has moved.

Notes: e: An object encapsulating information about the wheel-scrolling event.

The default implementation simply passes this message to the next responder.

Return true to not pass the event.

9.7.54 swipeWithEvent(e as NSEventMBS) as boolean

Plugin Version: 10.3, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a swipe gesture.

Notes: e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.7.55 updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Invoked when the dragging images should be changed.

Notes: sender: The object sending the message; use this object to get details about the dragging operation.

While a destination may change the dragging images at any time, it is recommended to wait until this method is called before updating the dragging images.

This allows the system to delay changing the dragging images until it is likely that the user will drop on this destination. Otherwise, the dragging images will change too often during the drag which would be distracting to the user.

9.7.56 viewDidMoveToWindow

Plugin Version: 11.2, Platform: macOS, Targets: .

Function: Informs the receiver that it has been added to a new view hierarchy.

Notes: The default implementation does nothing; subclasses can implement this event to perform whatever actions are necessary.

window may return nil when this method is invoked, indicating that the receiver does not currently reside in any window. This occurs when the receiver has just been removed from its superview or when the receiver has just been added to a superview that does not itself have a window. Overrides of this method may choose to ignore such cases if they are not of interest.

9.7.57 wantsPeriodicDraggingUpdates as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Asks the destination object whether it wants to receive periodic draggingUpdated messages.

Notes: Return true if the destination wants to receive periodic draggingUpdated messages, false otherwise.

If the destination returns false, these messages are sent only when the mouse moves or a modifier flag changes. Otherwise the destination gets the default behavior, where it receives periodic dragging-updated messages even if nothing changes.

9.8 class DateTimePicker

9.8.1 class DateTimePicker

Plugin Version: 20.5, Platform: macOS, Targets: Desktop only.

Function: The built-in date time picker control in Xojo.

Notes: Only for Xojo 2020r2 or newer.

9.8.2 Methods

9.8.3 NSDatePickerMBS as NSDatePickerMBS

Plugin Version: 20.5, Platform: macOS, Targets: Desktop only.

Function: Queries date picker object for the control.

Notes: Only for Xojo 2020r2 or newer.

9.9 class DesktopBevelButton

9.9.1 class DesktopBevelButton

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.9.2 Methods

9.9.3 NSButtonMBS as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox BevelButton1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.10 class DesktopButton

9.10.1 class DesktopButton

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in Button class in Xojo.

9.10.2 Methods

9.10.3 NSButtonMBS as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox PushButton1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.11 class DesktopCheckBox

9.11.1 class DesktopCheckBox

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.11.2 Methods

9.11.3 NSButtonMBS as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox CheckBox1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.12 class DesktopComboBox

9.12.1 class DesktopComboBox

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in ComboBox class in Xojo.

9.12.2 Methods

9.12.3 NSComboBoxMBS as NSComboBoxMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSComboBoxMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.13 class DesktopDateTimePicker

9.13.1 class DesktopDateTimePicker

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built-in date time picker control in Xojo.

Notes: Only for Xojo 2021r3 or newer.

9.13.2 Methods

9.13.3 NSDatePickerMBS as NSDatePickerMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Queries date picker object for the control.

Notes: Only for Xojo 2020r2 or newer.

9.14 class DesktopDisclosureTriangle

9.14.1 class DesktopDisclosureTriangle

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in DisclosureTriangle class in Xojo.

9.14.2 Methods

9.14.3 NSButtonMBS as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox DisclosureTriangle1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.15 class DesktopGroupBox

9.15.1 class DesktopGroupBox

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in Groupbox class in Xojo.

9.15.2 Methods

9.15.3 NSBoxMBS as NSBoxMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSBoxMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.16 class DesktopImageViewer

9.16.1 class DesktopImageViewer

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in ImageWell class in Xojo.

9.16.2 Methods

9.16.3 NSImageViewMBS as NSImageViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSImageViewMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.17 control DesktopNSButtonControlMBS

9.17.1 control DesktopNSButtonControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The Xojo control for a NSButton.

Notes: This control embeds a special NSButton subclass.

Designed for Xojo 2021r3 and newer.

Please use view property to access the underlying object and set properties.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo / Real Studio Plugins, version 13.2pr6](#)

9.17.2 Properties

9.17.3 AlternateTitle as String

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The title that the button displays when it's in its alternate state.

Notes: (Read and Write property)

9.17.4 BezelStyle as Integer

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The appearance of the border, if the view has one.

Notes: Use this constants:

(Read and Write property)

9.17.5 ButtonType as Integer

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The button type.

Notes: See also NSButtonMBS.ButtonType and the constants there like NSPushOnPushOffButton.

(Read and Write property)

NSRoundedBezelStyle	= 1
NSRegularSquareBezelStyle	= 2
NSThickSquareBezelStyle	= 3
NSThickerSquareBezelStyle	= 4
NSDisclosureBezelStyle	= 5
NSShadowlessSquareBezelStyle	= 6
NSCircularBezelStyle	= 7
NSTexturedSquareBezelStyle	= 8
NSHelpButtonBezelStyle	= 9
NSSmallSquareBezelStyle	= 10
NSTexturedRoundedBezelStyle	= 11
NSRoundRectBezelStyle	= 12
NSRecessedBezelStyle	= 13
NSRoundedDisclosureBezelStyle	= 14

9.17.6 Title as String

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The title displayed on the button when it's in its normal state.

Notes: (Read and Write property)

9.17.7 View as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.
(Read only property)

9.17.8 Events

9.17.9 Action

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The action event.

9.17.10 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.17.11 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.17.12 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.17.13 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.17.14 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.17.15 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.
In older Xojo versions, this event is named EnableMenuItems.

9.17.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ã region at the location passed in to x, y.
Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.
Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.17.17 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.
Notes: Mouse location is local to the control passed in to x, y.
As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.17.18 MouseUp(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.
Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.17.19 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.17.20 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.18 control DesktopNSComboBoxControlMBS

9.18.1 control DesktopNSComboBoxControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The Xojo control for a NSComboBox.

Notes: This control embeds a special NSComboBox subclass.

Designed for Xojo 2021r3 and newer.

Please use view property to access the underlying object and set properties.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.4](#)
- [MBS Xojo Plugins, version 21.4pr2](#)

Xojo Developer Magazine

- [19.6, page 10: News](#)

9.18.2 Properties

9.18.3 View as NSComboBoxMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.

(Read only property)

9.18.4 Events

9.18.5 Action

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The action event run, when e.g. return key is pressed.

9.18.6 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.18.7 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.18.8 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.18.9 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.18.10 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.18.11 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.
In older Xojo versions, this event is named EnableMenuItems.

9.18.12MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.
Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.
Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.18.13 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.
Notes: Mouse location is local to the control passed in to x, y.
As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.18.14 MouseUp(x As Integer, y As Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.
Notes: Use the x and y parameters to determine if the mouse button was released within the control's

boundaries.

9.18.15 ScaleFactorChanged(NewFactor as double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.18.16 TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent when a control with editable text begins an editing session.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidBeginEditingNotification`.

This event is invoked when the user begins editing text in a control such as a text field or a form field. The control posts a `NSControlTextDidBeginEditingNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is also delivered for inspection.

9.18.17 TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent when the text in the receiving control changes.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidChangeNotification`.

This event is invoked when text in a control such as a text field or form changes. The control posts a `NSControlTextDidChangeNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is provided as parameter for inspection.

9.18.18 TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent when a control with editable text ends an editing session.

Notes: Notification: The notification object. The name of the notification is always NSControlTextDidEndEditingNotification.

This event is invoked when the user stops editing text in a control such as a text field or form. The control posts a NSControlTextDidEndEditingNotification notification, and if the control' subclass implements this event, it is automatically registered to receive the notification. The field editor is also provided for inspection.

9.18.19 textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.18.20 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow end of text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.18.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.19 control DesktopNSDatePickerControlMBS

9.19.1 control DesktopNSDatePickerControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The control to put a Mac OS X date picker on a Xojo window.

Notes: This control embeds a special NSDatePicker subclass.

Designed for Xojo 2021r3 and newer.

Please use view property to access the underlying object and set properties.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo Plugins, version 19.2pr1](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr7](#)

9.19.2 Properties

9.19.3 AcceptTabs as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether the control should accept tab keys.

Notes: If true, the plugin will not forward the tab keydown/keyup events to Xojo, because Xojo would do switch to next control.

(Read and Write property)

9.19.4 View as NSDatePickerMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The reference to the underlying NSDatePicker.

Notes: (Read only property)

9.19.5 Events

9.19.6 Action

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event is called when user clicks on a date/time and changes something.

9.19.7 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.19.8 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.19.9 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.19.10 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.19.11 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.19.12 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In older Xojo versions, this event is named EnableMenuItems.

9.19.13MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.19.14 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.19.15 MouseUp(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.19.16 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.19.17 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.20 control DesktopNSOutlineControlMBS

9.20.1 control DesktopNSOutlineControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The control for a NSOutlineView.

Notes: Please use NSOutlineControlMBS for hierarchical lists and NSTableControlMBS for normal lists.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.1](#)
- [MBS Xojo Plugins, version 17.1pr4](#)

Videos

- [Presentation from London conference about MBS Plugins.](#)

Xojo Developer Magazine

- [15.3, page 10: News](#)

9.20.2 Properties

9.20.3 AcceptTabs as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether the control should accept tab keys.

Notes: If true, the plugin will not forward the tab keydown/keyup events to Xojo, because Xojo would do switch to next control.

(Read and Write property)

9.20.4 allowsColumnReordering as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to rearrange columns by dragging their headers.

Notes: The default value of this property is true, which allows the user to rearrange the table view,Äôs columns. You can rearrange columns programmatically regardless of this setting.

(Read and Write property)

9.20.5 allowsColumnResizing as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to resize columns by dragging between their headers.

Notes: The default of this property is true, which allows the user to resize the table view's columns. You can resize columns programmatically regardless of this setting.

(Read and Write property)

9.20.6 allowsColumnSelection as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select columns by clicking their headers.

Notes: The default is false, which prevents the user from selecting columns (if you create the table view in Interface Builder, the default value is true). You can select columns programmatically regardless of this setting.

(Read and Write property)

9.20.7 allowsEmptySelection as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select zero columns or rows.

Notes: The default is true, which allows the user to select zero columns or rows.

(Read and Write property)

9.20.8 allowsMultipleSelection as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select more than one column or row at a time.

Notes: The default is false, which allows the user to select only one column or row at a time. You can select multiple columns or rows programmatically regardless of this setting.

(Read and Write property)

9.20.9 autohidesScrollers as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view automatically hides its scroll bars when they are not needed.

Notes: The horizontal and vertical scroll bars are hidden independently of each other. When the value of this property is YES and the content of the scroll view doesn't extend beyond the size of the clip view on a given axis, the scroller on that axis is removed to leave more room for the content.

(Read and Write property)

9.20.10 hasHorizontalScroller as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a horizontal scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a horizontal scroller as needed. The default value of this property is false.

(Read and Write property)

9.20.11 hasVerticalScroller as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a vertical scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a vertical scroller as needed. The default value of this property is false.

(Read and Write property)

9.20.12 ScrollView as NSScrollViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The scroll view used in this control.

Notes: (Read only property)

9.20.13 View as NSOutlineViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The outline view used in this control.

Notes: (Read only property)

See also:

- 9.20.90 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS
399

9.20.14 Events

9.20.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a drop operation was successful.

Notes: info: An object that contains more information about this dragging operation.

item: The parent of the item over which the cursor was placed when the mouse button was released.

index: The index of the child of item over which the cursor was placed when the mouse button was released.

Return true if the drop operation was successful, otherwise false.

The data source should incorporate the data from the dragging pasteboard in the implementation of this method. You can get the data for the drop operation from info using the draggingPasteboard method. The return value indicates success or failure of the drag operation to the system.

9.20.16 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.20.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the child item at the specified index of a given item.

Notes: index: The index of the child item from item to return.

item: An item in the data source.

Return the child item at index of item. If item is nil, returns the appropriate child item of the root object.

Children of a given parent item are accessed sequentially. In order for the collapsed state of the outline view to remain consistent when it is reloaded you must always return the same object for a specified child and item.

Do not call `reloadData` from this method.

This event is called very frequently, so it must be efficient.

9.20.18 `ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked whenever the user moves a column in the outline view.

9.20.19 `ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked whenever the user resizes a column in the outline view.

9.20.20 `concludeDragOperation(info as NSDraggingInfoMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when the dragging operation is complete, signaling the receiver to perform any necessary clean-up.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

For this method to be invoked, the previous `performDragOperation` must have returned true.

The destination implements this method to perform any tidying up that it needs to do, such as updating its visual representation now that it has incorporated the dragged data. This message is the last message sent from sender to the destination during a dragging session.

If the sender object's `animatesToDestination` property was set to true in `prepareForDragOperation`, then the drag image is still visible. At this point you should draw the final visual representation in the view. When this method returns, the drag image is removed from the screen. If your final visual representation matches the visual representation in the drag, this is a seamless transition.

9.20.21 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the cell to use in a given column for a given item.

Notes: tableColumn: The table column for which the cell is required. This value may be nil.

item: The item for which the cell is required.

Return the cell to use in column tableColumn for item item, or nil. The cell must properly implement copyWithZone (since it may be copied by by the outline view).

You can return a different data cell for any particular combination of table column and item, or a cell that will be used for the entire row (a full-width cell). If tableColumn is non-nil, you should return a cell. Typically, you should default to returning the result from [tableColumn dataCellForRow:row] .

When each row (identified by the item) is being drawn, this method is first called with a nil value for tableColumn. At this time, you can return a cell that is used to draw the entire row, acting like a group. If you do return a cell for the nil table column, your implementations of the other corresponding data source and delegate methods must be prepared to be invoked with a nil value for tableColumn. If do not return a cell for the nil table column, the method is called once for each column in the outline view, as usual.

9.20.22 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implemented to know when a new row view is added to the table.

Notes: rowView: The new row view.

row: The row to which the view was added.

This event is for NSView-based outline views. At this point, you can choose to add in extra views or modify any properties on rowView.

9.20.23 didClickTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent at the time the mouse button subsequently goes up in outlineView and tableColumn has been ,Äúclicked,Äù without having been dragged anywhere.

9.20.24 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.20.25 didDragTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent at the time the mouse button goes up in outlineView and tableColumn has been dragged during the time the mouse button was down.

9.20.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implemented to know when a row view is removed from the table

Notes: rowView: The row view that was removed.

row: The number of the row that was removed due to being moved offscreen, or -1 if the row was removed from the table so it is no longer valid.

The removed rowView may be reused by the table, so any additionally inserted views should be removed at this point.

9.20.27 didTile

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The tableview did tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.20.28 DoubleClick

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse made a double click.

9.20.29 draggingEnded(info as NSDraggingInfoMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this event to be notified when a drag operation ends in some other destination.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

This method might be used by a destination doing auto-expansion in order to collapse any auto-expands.

9.20.30 draggingExited(info as NSDraggingInfoMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when the dragged image enters destination bounds or frame; delegate returns dragging operation to perform.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in NSDragOperation in the NSDraggingInfo reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous draggingEntered message.

Invoked when a dragged image enters the destination but only if the destination has registered for the pasteboard data type involved in the drag operation. Specifically, this method is invoked when the mouse pointer enters the destination's bounds rectangle (if it is a view object) or its frame rectangle (if it is a window object).

This method must return a value that indicates which dragging operation the destination will perform when the image is released. In deciding which dragging operation to return, the method should evaluate the overlap between both the dragging operations allowed by the source (obtained from sender with the draggingSourceOperationMask method) and the dragging operations and pasteboard data types the destination itself supports.

If none of the operations is appropriate, this method should return NSDragOperationNone (this is the default response if the method is not implemented by the destination). A destination will still receive draggingUpdated and draggingExited even if NSDragOperationNone is returned by this method.

9.20.31 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to know when the given dragging session has ended.

Notes: session: The dragging session that ended.

screenPoint: The point onscreen at which the drag ended.

operation: A mask specifying the types of drag operations permitted by the dragging source.

You can implement this optional delegate method to know when the dragging source operation ended at a specific location, such as the trash (by checking for an operation of `NSDragOperationDelete`).

9.20.32 `draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method know when the given dragging session is about to begin and potentially modify the dragging session.

Notes: session: The dragging session that is about to begin.

screenPoint: The point onscreen at which the drag is to begin.

draggedItems: A array of items to be dragged, excluding items for which `pasteboardWriterForItem` returns nil.

The `draggedItems` array directly matches the pasteboard writer array used to begin the dragging session with the `NSView` method `beginDraggingSessionWithItems`. Hence, the order is deterministic, and can be used in `acceptDrop` when enumerating the `NSDraggingInfo` protocol's pasteboard classes.

9.20.33 `FocusLost`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named `LostFocus`.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.20.34 `FocusReceived`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.20.35 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.20.36 heightForRowByItem(item as NSOutlineViewItemMBS) as Double

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the height in points of the row containing item.

Notes: item: The row item.

Return the height of the row.

Values returned by this method should not include intercell spacing and must be greater than 0. Implement this event to support an outline view with varying row heights.

For large tables in particular, you should make sure that this method is efficient. NSOutlineView may cache the values this method returns, so if you would like to change a row's height make sure to invalidate the row height by calling noteHeightOfRowsWithIndexesChanged. NSOutlineView automatically invalidates its entire row height cache in reloadData and noteNumberOfRowsChanged.

If you call viewAtColumn or rowViewAtRow within your implementation of this method, an exception is thrown.

To avoid the possibility of a hang due to unexpected recursion, don't call geometry-calculating methods such as bounds, rectOfColumn, or any NSTableView method that calls tile within your implementation of this method.

9.20.37 isGroupItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean that indicates whether a given row should be drawn in the „group row,“ style.

Notes: item: An item in the outline view.

Return true to indicate a particular row should have the „group row“ style drawn for that row, otherwise false.

If the cell in that row is an instance of `NSTextFieldCell` and contains only a string value, the „group row,“ style attributes are automatically applied for that cell.

9.20.38 `isItemExpandable(item as NSOutlineViewItemMBS)` as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the a given item is expandable.

Notes: item: An item in the data source.

Returns true if item can be expanded to display its children, otherwise NO.

This method may be called quite often, so it must be efficient.

Do not call `reloadData` from this method.

9.20.39 `ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when the did collapse notification is posted—that is, whenever the user collapses an item in the outline view.

9.20.40 `ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the user expands an item in the outline view.

9.20.41 itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked by outlineView to return the item for the archived object.

Notes: object: An archived representation of an item in outlineView's data source.

Return the unarchived item corresponding to object. If the item is an archived object, this method may return the object.

When the outline view is restoring the saved expanded items, this method is called for each expanded item, to translate the archived object to an outline view item.

You must implement this method if you are automatically saving expanded items (that is, if autosaveExpandedItems returns true).

9.20.42 ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the user is about to collapse an item in the outline view.

9.20.43 ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the user is about to expand an item in the outline view.

9.20.44 LeftMouseDown(e as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.45 LeftMouseDown(e as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the left button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.46 LeftMouseUp(e as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.47 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In older Xojo versions, this event is named EnableMenuItems.

9.20.48 MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.20.49 `mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Event sent whenever the mouse button is clicked in outlineView while the cursor is in a column header tableColumn.

9.20.50 `MouseDown(x as Integer, y as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.20.51 `MouseDown(x as Integer, y as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.20.52 `namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns an array of filenames for the created files that the receiver promises to create.

Notes: dropDestination: The drop location where the files are created.

items: The items being dragged.

Returns an array of filenames (not full paths) for the created files that the receiver promises to create.

For more information on file promise dragging, see documentation on the NSDraggingSource protocol and `namesOfPromisedFilesDroppedAtDestination`.

9.20.53 `nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the first item that matches the searchString from within the range of startItem to endItem.

Notes: startItem: The first item to search.

endItem: The item before which to stop searching. It is possible for endItem to be less than startItem if the search will wrap.

searchString: The string for which to search.

Returns the first item—from within the range of startItem to endItem—that matches the searchString, or nil if there is no match.

Implement this method if you want to control how type selection works. You should include startItem as a possible match, but do not include endItem.

It is not necessary to implement this event in order to support type select.

9.20.54 `numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the number of child items encompassed by a given item.

Notes: item: An item in the data source.

Returns the number of child items encompassed by item. If item is nil, this method should return the number of children for the top-level item.

The numberOfChildrenOfItem method is called very frequently, so it must be efficient.

Do not call reloadData from this method.

9.20.55 `objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked by outlineView to return the data object associated with the specified item.

Notes: tableColumn: A column in outlineView.

item: An item in the data source in the specified tableColumn of the view.

Returns the item is located in the specified tableColumn of the view.

Do not call reloadData from this method.

9.20.56 OtherMouseDown(e as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed a mouse button other than the left or right one.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.57 OtherMouseDragged(e as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with a button other than the left or right button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.58 OtherMouseUp(e as NSEventMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released a mouse button other than the left or right button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.59 pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to enable the table to be an NSDraggingSource that supports dragging multiple items.

Notes: item: The item for which to return a pasteboard writer.

Returns a `NSPasteboardItem` object.

If this method is implemented, then `writeItems` is not called.

9.20.60 `persistentObjectForItem(item as NSOutlineViewItemMBS)` as Variant

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked by `outlineView` to return an archived object for item.

Notes: item: The item for which to return an archived object.

Returns an archived representation of item. If the item is an archived object, this method may return the item.

When the outline view is saving the expanded items, this method is called for each expanded item, to translate the outline view item to an archived object.

You must implement this method if you are automatically saving expanded items (that is, if `autosaveExpandedItems` returns true).

9.20.61 `RightMouseDown(e as NSEventMBS)` as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the view that the user has pressed the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.62 `RightMouseDragged(e as NSEventMBS)` as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the right button pressed .

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.63 `RightMouseUp(e as NSEventMBS)` as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.20.64 rowViewForItem(item as NSOutlineViewItemMBS) as NSTableViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to return a custom NSTableView for a particular item.

Notes: item: The item displayed by the returned table row view.

Return an instance or subclass of NSTableView. If nil is returned, a NSTableView instance is created and used.

This method, if implemented, is only invoked for NSView-based outline views.

9.20.65 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.20.66 SelectionDidChange(notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when the selection did change notification is posted—that is, immediately after the outline view’s selection has changed.

9.20.67 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to modify the proposed selection.

Notes: proposedSelectionIndexes: An index set containing the indexes of the proposed selection.

Return an NSIndexSet instance containing the indexes of the new selection. Return proposedSelectionIndexes if the proposed selection is acceptable, or the value of the table view’s existing selection to avoid

changing the selection.

This method may be called multiple times with one new index added to the existing selection to find out if a particular index can be selected when the user is extending the selection with the keyboard or mouse.

Implementation of this method is optional. If implemented, this method will be called instead of `willDisplayOutlineCell`.

If not implemented or returns `nil`, the plugin will return `proposedSelectionIndexes`.

9.20.68 SelectionIsChanging(notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the outline view’s selection changes.

9.20.69 selectionShouldChangeInOutlineView as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should change its selection.

Notes: Return true to permit `outlineView` to change its selection (typically a row being edited), false to deny permission.

For example, if the user is editing a cell and enters an improper value, the delegate can prevent the user from selecting or editing any other cells until a proper value has been entered into the original cell. You can implement this method for complex validation of edited rows based on the values of any of their cells.

9.20.70 setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Set the data object for a given item in a given column.

Notes: `object`: The new value for the item.

`tableColumn`: A column in `outlineView`.

`item`: An item in the data source in the specified `tableColumn` of the view.

The item is located in the specified `tableColumn` of the view.

Do not call reloadData from this method.

9.20.71 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether auto-expanded items should return to their original collapsed state.

Notes: deposited: If true, the drop terminated successfully; if false the drop failed.

Return true if auto-expanded items should return to their original collapsed state; otherwise false.

Implement this event to provide custom behavior. If the target of a drop is not auto-expanded (by hovering long enough) the drop target still gets expanded after a successful drop unless this method returns true. The default implementation returns false after a successful drop.

This method is called in a variety of situations. For example, it is called shortly after the acceptDrop method is called and also if the drag exits the outline view (exiting the view is treated the same as a failed drop). The return value of the acceptDrop method determines the incoming value of the deposited parameter.

9.20.72 shouldCollapseItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should collapse a given item.

Notes: item: The item that should collapse.

Return true to permit outlineView to collapse item, false to deny permission.

You can implement this method to disallow collapsing of specific items. For example, if the first row of your outline view should not be collapsed, your delegate method could contain this line of code:

```
return rowForItem(item) <>0
```

9.20.73 shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should allow editing of a given

item in a given table column.

Notes: tableColumn: The table column.
item: The item.

Returns true to permit outlineView to edit the cell specified by tableColumn and item, false to deny permission.

If this method returns true, the cell may still not be editable—for example, if you have set up a custom NSTextFieldCell as a data cell, it must return true for isEditable to allow editing.

You can implement this method to disallow editing of specific cells.

9.20.74 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should expand a given item.

Notes: item: The item that should expand.

Returns true to permit outlineView to expand item, false to deny permission.

You can implement this method to disallow expanding of specific items.

9.20.75 shouldReorderColumn(columnIndex as Integer, newColumnIndex as Integer) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent to the delegate to allow or prohibit the specified column to be dragged to a new location.

Notes: columnIndex: The index of the column being dragged.

newColumnIndex: The proposed target index of the column.

Returns true if the column reordering should be allowed, otherwise false.

When a column is initially dragged by the user, the delegate is first called with a newColumnIndex value of -1. Returning false will disallow that column from being reordered at all. Returning true allows it to be reordered, and the delegate will be called again when the column reaches a new location.

The actual NSTableColumn instance can be retrieved from the tableColumns array.

If this method is not implemented, all columns are considered reorderable.

9.20.76 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should select a given item.

Notes: item: The item.

Return true to permit outlineView to select item, false to deny permission.

You implement this event to disallow selection of particular items.

For better performance and finer grain control over the selection, use dataCell.

9.20.77 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should select a given table column.

Notes: tableColumn: The table column.

Return true to permit outlineView to select tableColumn, false to deny permission.

You can implement this method to disallow selection of specific columns.

9.20.78 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to control cell expansion for a specific column and item.

Notes: tableColumn: A table column in the outline view.

item: An item in the outline view.

Returns true to allow an expansion tooltip to appear in the column tableColumn for item item, otherwise false.

Cell expansion can occur when the mouse hovers over the specified cell and the cell contents are unable to be fully displayed within the cell. If this method returns true, the full cell contents will be shown in a special floating tool tip view, otherwise the content is truncated.

9.20.79 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns whether the specified item should display the outline cell (the disclosure triangle).

Notes: item: An item in the outline view.

Returns true if the outline cell should be displayed, otherwise false.

Returning false causes `frameOfOutlineCellAtRow` to return `NSZeroRect`, hiding the cell. In addition, the row will not be collapsible by keyboard shortcuts.

This method is called only for expandable rows.

9.20.80 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a given cell should be tracked.

Notes: cell: The cell used to display item item in column tableColumn

tableColumn: A table column in the outline view.

item: An item in the outline view.

Returns true if the cell should be tracked for the item item in column tableColumn, otherwise false.

Normally, only selectable or selected cells can be tracked. If you implement this method, cells which are not selectable or selected can be tracked (and vice-versa). For example, this allows you to have a button cell in a table which does not change the selection, but can still be clicked on and tracked.

9.20.81 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether type select should proceed for a given event and search string.

Notes: e: The event that caused the message to be sent.

searchString: The string for which searching is to proceed. The search string is nil if no type select has begun.

Return true if type select should proceed, otherwise false.

Generally, this method will be called from `keyDown` and the event will be a key event.

9.20.82 `sizeToFitWidthOfColumn(Column as Integer) as Double`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to provide custom sizing behavior when a column's resize divider is double clicked.

Notes: `column`: The index of the column.

Returns the width of the specified column.

By default, `NSOutlineView` iterates every row in the table, accesses a cell via `preparedCellAtColumn`, and requests the `cellSize` to find the appropriate largest width to use.

For accurate results and performance, it is recommended that this method is implemented when using large tables. By default, large tables use a monte carlo simulation instead of iterating every row.

9.20.83 `sortDescriptorsDidChange(oldDescriptors()) as NSSortDescriptorMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked by an outline view to notify the data source that the descriptors changed and the data may need to be resorted.

Notes: `oldDescriptors`: An array that contains the previous descriptors.

The data source typically sorts and reloads the data, and adjusts the selections accordingly. If you need to know the current sort descriptors and the data source does not itself manage them, you can get `outlineView`'s current sort descriptors by sending it a `sortDescriptors` message.

9.20.84 `textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow.

9.20.85 `textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow.

9.20.86 `toolTipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: When the cursor pauses over a given cell, the value returned from this method is displayed in a tooltip.

Notes: cell: The cell for which to generate a tooltip.

rect: The proposed active area of the tooltip. To control the default active area, you can modify the rect parameter. By default, rect is computed as `cell.drawingRectForBounds(cellFrame)`.

tc: The table column that contains cell.

item: The item for which to display a tooltip.

mouseLocation: The current mouse location in view coordinates.

If you don't want a tooltip at that location, return nil or the empty string.

9.20.87 `typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns the string that is used for type selection for a given column and item.

Notes: tableColumn: A table column in the outline view.

item: An item in the outline view.

Return the string that is used for type selection. You may want to change what is searched for based on what is displayed, or simply return nil for that row and/or column to not be searched

Implement this method if you want to control the string that is used for type selection. You may want to change what is searched for based on what is displayed, or simply return nil to specify that the given row and/or column should not be searched. By default, all cells with text in them are searched.

The default value when this delegate method is not implemented is:

`outlineView.preparedCellAtColumn(tableColumn, outlineView.rowForItem(item)).stringValue` and you can return this value from the event if you wish.

9.20.88 `updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to enable the table to update dragging items as they are dragged over the view.

Notes: `draggingInfo`: The dragging info object.

9.20.89 `validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Used by an outline view to determine a valid drop target.

Notes: `info`: An object that contains more information about this dragging operation.

`item`: The proposed parent.

`index`: The proposed child location.

Returns a value that indicates which dragging operation the data source will perform.

Based on the mouse position, the outline view will suggest a proposed drop location. The data source may, if desired, retarget a drop by calling `setDropItem` and returning something other than `NSDragOperationNone`. You may choose to retarget for various reasons (for example, for better visual feedback when inserting into a sorted position).

Implementation of this method is optional.

9.20.90 `view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implemented to return the view used to display the specified item and column.

Notes: `tableColumn`: The table column, or nil if the row is a group row.

`item`: The item displayed by the returned view.

Return the view to display the specified column and row. Returning nil is acceptable, in which case a view is not shown at that location.

This method is required if you wish to use `NSView` objects instead of `NSCell` objects for the cells within an outline view. Cells and views cannot be mixed within the same outline view.

It is recommended that the implementation of this method first call the `NSTableView` method `makeViewWithIdentifier` passing, respectively, the `tableColumn` parameter, the identifier and `self` as the owner to attempt to reuse a view that is no longer visible. The frame of the view returned by this method is not important, and is automatically set by the outline view.

The view's properties should be properly set up before returning the result.

When using Cocoa bindings, this method is optional if at least one identifier has been associated with the table view at design time. If this method is not implemented, the outline view automatically calls `makeViewWithIdentifier` with the `tableColumn` parameter, the identifier and the outline view, the delegate as parameters, to attempt to reuse a previous view or automatically unarchive a prototype associated with the table view.

The `autoresizingMask` of the returned view is automatically set to `NSViewHeightSizable` to resize properly on row height changes.

See also:

- 9.20.13 View as `NSOutlineViewMBS`

376

9.20.91 `willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the delegate that the cell specified by the column and item will be displayed.

Notes: `cell`: The cell.

`tableColumn`: The table column.

`item`: The item.

You can implement this method to modify `cell` to provide further setup for the cell in `tableColumn` and `item`. It is not safe to do drawing inside this method—you should only set up state for `cell`.

9.20.92 `willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the delegate that an outline view is about to display a cell used to draw the expansion symbol.

Notes: `cell`: The cell.

`tableColumn`: The table column.

`item`: The item.

Informs the event that `outlineView` is about to display `cell`—an expandable cell (a cell that has the expansion symbol)—for the column and item specified by `tableColumn` and `item`. The delegate can modify `cell` to alter

its display attributes.

This method is not invoked when outlineView is about to display a non-expandable cell.

9.20.93 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.20.94 willTile

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The tableview will tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.20.95 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a drag operation is allowed.

Notes: items: An array of the items participating in the drag.

pasteboard: The pasteboard to which to write the drag data.

Returns true if the drag operation is allowed, otherwise false.

Invoked by outlineView after it has been determined that a drag should begin, but before the drag has been started.

To refuse the drag, return false. To start a drag, return true and place the drag data onto the pboard (data, owner, and so on). The drag image and other drag-related information will be set up and provided by the outline view once this call returns with true.

9.21 control DesktopNSPopUpButtonControlMBS

9.21.1 control DesktopNSPopUpButtonControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The Xojo control for a NSPopUpButtonControl.

Notes: This control embeds a special NSPopUpButtonControl subclass.
Designed for Xojo 2021r3 and newer.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo / Real Studio Plugins, version 13.2pr6](#)

9.21.2 Properties

9.21.3 View as NSPopUpButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.
(Read only property)

9.21.4 Events

9.21.5 Action

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The action event.

9.21.6 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.21.7 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.21.8 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.21.9 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.21.10 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.21.11 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.
In older Xojo versions, this event is named `EnableMenuItems`.

9.21.12 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the `MouseDown`. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the `MouseDown` and `MouseUp` events.

If you return False, the system handles the `MouseDown` so the above event handlers do not get called.

9.21.13 `MouseDown(x as Integer, y as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.21.14 `MouseUp(x as Integer, y as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.21.15 `ScaleFactorChanged(NewFactor as Double)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.21.16 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.22 control DesktopNSSwitchControlMBS

9.22.1 control DesktopNSSwitchControlMBS

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The control to host a NSSwitch as control in Xojo.

Notes: Available on macOS 10.15 or newer.

Blog Entries

- [News from the MBS Xojo Plugins Version 22.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 22.1](#)
- [MBS Xojo Plugins, version 22.1pr3](#)
- [Adding NSSwitch to Xojo](#)
- [MBS Xojo Plugins, version 22.1pr2](#)

Xojo Developer Magazine

- [20.4, page 10: News](#)
- [20.3, page 10: News](#)

9.22.2 Properties

9.22.3 State as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The current state.

Notes: Set to true for toggle on and set to false for toggle off.
(Read and Write property)

9.22.4 View as NSSwitchMBS

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.
(Read only property)

9.22.5 Events

9.22.6 Action

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The action event.

9.22.7 BoundsChanged

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.22.8 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.22.9 FocusLost

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.22.10 FocusReceived

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.22.11 FrameChanged

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.22.12 MenuBarSelected

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In older Xojo versions, this event is named EnableMenuItems.

9.22.13MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.22.14 MouseDrag(x as Integer, y as Integer)

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the

mouse has really moved.

9.22.15 MouseUp(x As Integer, y As Integer)

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.22.16 ScaleFactorChanged(NewFactor as double)

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.22.17 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.23 control DesktopNSTableControlMBS

9.23.1 control DesktopNSTableControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The control for a NSTableView.

Notes: Please use NSOutlineControlMBS for hierarchical lists and NSTableControlMBS for normal lists.

Blog Entries

- [MBS Xojo Plugins, version 22.5pr3](#)
- [Xojo 2022r3 released](#)
- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.1](#)
- [MBS Xojo Plugins, version 17.1pr5](#)
- [MBS Xojo Plugins, version 17.1pr3](#)
- [MBS Xojo Plugins, version 17.1pr2](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr2](#)

Videos

- [Presentation from London conference about MBS Plugins.](#)

Xojo Developer Magazine

- [15.3, page 10: News](#)
- [14.4, page 22: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.23.2 Properties

9.23.3 AcceptTabs as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether the control should accept tab keys.

Notes: If true, the plugin will not forward the tab keydown/keyup events to Xojo, because Xojo would do switch to next control.

(Read and Write property)

9.23.4 allowsColumnReordering as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to rearrange columns by dragging their headers.

Notes: The default value of this property is true, which allows the user to rearrange the table view's columns. You can rearrange columns programmatically regardless of this setting.
(Read and Write property)

9.23.5 allowsColumnResizing as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to resize columns by dragging between their headers.

Notes: The default of this property is true, which allows the user to resize the table view's columns. You can resize columns programmatically regardless of this setting.
(Read and Write property)

9.23.6 allowsColumnSelection as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select columns by clicking their headers.

Notes: The default is false, which prevents the user from selecting columns (if you create the table view in Interface Builder, the default value is true). You can select columns programmatically regardless of this setting.
(Read and Write property)

9.23.7 allowsEmptySelection as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select zero columns or rows.

Notes: The default is true, which allows the user to select zero columns or rows.
(Read and Write property)

9.23.8 allowsMultipleSelection as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select more than one column or row at a time.

Notes: The default is false, which allows the user to select only one column or row at a time. You can select multiple columns or rows programmatically regardless of this setting.
(Read and Write property)

9.23.9 autohidesScrollers as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view automatically hides its scroll bars when they are not needed.

Notes: The horizontal and vertical scroll bars are hidden independently of each other. When the value of this property is YES and the content of the scroll view doesn't extend beyond the size of the clip view on a given axis, the scroller on that axis is removed to leave more room for the content.
(Read and Write property)

9.23.10 disableCellEvents as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether to disable cell based events.

Notes: The table view can work with cell modes and use NSCell to disable cells.
Or since OS X 10.7 it can work with NSView to display cells or rows.
This property lets you explicitly disable cells and use only views.
(Read and Write property)

9.23.11 disableViewEvents as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Whether to disable view based events.

Notes: The table view can work with cell modes and use NSCell to disable cells.
Or since OS X 10.7 it can work with NSView to display cells or rows.
This property lets you explicitly disable views and use only cells.
(Read and Write property)

9.23.12 hasHorizontalScroller as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a horizontal scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a horizontal scroller as needed. The default value of this property is false.

(Read and Write property)

9.23.13 hasVerticalScroller as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a vertical scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a vertical scroller as needed. The default value of this property is false.

(Read and Write property)

9.23.14 ScrollView as NSScrollViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The scroll view used in this control.

Notes: (Read only property)

9.23.15 View as NSTableViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The table view used in this control.

Notes: (Read only property)

See also:

- 9.23.78 view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS

432

9.23.16 Events**9.23.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean**

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called by `TableView` when the mouse button is released over a table view that previously decided to allow a drop.

Notes: `info`: An object that contains more information about this dragging operation.

`row`: The index of the proposed target row.

`operation`: The type of dragging operation.

Returns true if the drop operation was successful, otherwise false.

The data source should incorporate the data from the dragging pasteboard in the implementation of this method. You can use the `draggingPasteboard` method to get the data for the drop operation from `info`.

To accept a drop on the second row, `row` would be 2 and `operation` would be `NSTableViewDropOn`. To accept a drop below the last row, `row` would be `TableView.numberOfRows` and `operation` would be `NSTableViewDropAbove`.

9.23.18 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.23.19 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event informs the delegate that a column was moved by user action in the table view.

9.23.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event informs you that a column was resized in the table view.

9.23.21 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Optional return a different cells for each row.

Notes: A different data cell can be returned for any particular `tableColumn` and `row`, or a cell that will

be used for the entire row (a full width cell). The returned cell should properly implement `copyWithZone:`, since the cell may be copied by `NSTableView`. If the `tableColumn` is non-nil, and nil is returned, then the table will use the default cell from `tableColumn.dataCellForRow(Row)`.

When each row is being drawn, this method will first be called with a nil `tableColumn`. At this time, you can return a cell that will be used to draw the entire row, acting like a group. If you do return a cell for the 'nil' `tableColumn`, be prepared to have the other corresponding datasource and delegate methods to be called with a 'nil' `tableColumn` value. If don't return a cell, the method will be called once for each `tableColumn` in the `tableView`, as usual.

9.23.22 `didAddRowView(rowView as NSTableRowViewMBS, row as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Tells the delegate that a row view was added at the specified row.

Notes: `rowView`: The row view.

`row`: The index of the row.

At this point, the delegate can add extra views, or modify the properties of `rowView`.

This method is only valid for `NSView`-based table views.

9.23.23 `didClickTableColumn(tableColumn as NSTableColumnMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called if a table column was clicked on.

9.23.24 `didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)`

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the `willShowContextualMenu` event.

9.23.25 `didDragTableColumn(tableColumn as NSTableColumnMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent at the time the mouse button goes up in tableView and tableColumn has been dragged during the time the mouse button was down.

Notes: tableColumn: The table column.

The behavior of this method on Mac OS X v10.5 is different from prior versions. On Mac OS X v 10.5 the dragged column is sent to the subclass. In earlier versions the table column that is currently located at the dragged column's original index is sent.

9.23.26 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Tells the delegate that a row view was removed from the table at the specified row.

Notes: rowView: The row view.

row: The index of the row.

If row equals -1, the row is being deleted from the table and is no longer a valid row; otherwise row is a valid row that is being removed by being moved off screen.

This method is only valid for NSView-based table views.

9.23.27 didTile

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The tableview did tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.23.28 DoubleClick

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse made a double click.

9.23.29 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to determine when a dragging session has ended.

Notes: session: The dragging session.

screenPoint: The ending drag location in screen coordinates.

operation: The drag operation. See NSDragOperation for supported values.

This delegate method can be used to determine when the dragging source operation ended at a specific location, such as the trash, by checking for an operation of NSDragOperationDelete.

9.23.30 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to determine when a dragging session will begin.

Notes: session: The dragging session.

screenPoint: The initial drag location in screen coordinates.

rowIndexes: The indexes of the rows to be dragged, excluding rows that were not dragged due to pasteboardItemForRow returning nil.

Implement this method to know when the dragging session is about to begin and to potentially modify the dragging session.

The dragged item order will directly match the pasteboard writer array used to begin the dragging session with the NSView method beginDraggingSessionWithItems. Hence, the order is deterministic, and can be used in acceptDrop when enumerating the NSDraggingInfo pasteboard classes.

9.23.31 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.23.32 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

9.23.33 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.23.34 heightForRow(row as Int64) as Double

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this event to support a table with varying row heights.

Notes: The height returned by this method should not include intercell spacing and must be greater than zero. Performance Considerations: For large tables in particular, you should make sure that this method is efficient. NSTableView may cache the values this method returns, but this should NOT be depended on, as all values may not be cached. To signal a row height change, call noteHeightOfRowsWithIndexesChanged. For a given row, the same row height should always be returned until noteHeightOfRowsWithIndexesChanged is called, otherwise unpredictable results will happen. NSTableView automatically invalidates its entire row height cache in reloadData, and noteNumberOfRowsChanged.

9.23.35 isGroupRow(row as Int64) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to indicate that a specified row is a group row.

Notes: row: The row index.

Return true if the specified row should have the group row style drawn, false otherwise.

If the cell in row is an NSTextFieldCell and contains only a string, the group row style attributes will automatically be applied to the cell.

Group rows in view-based table views can be made to visually ‚float‘ by setting the tableview method `setFloatsGroupRows` to true.

Note: When configured as a source list style table view, rows identified as group rows draw with a specific style unique for source lists.

Available in Mac OS X v10.5 and later.

9.23.36 LeftMouseDown(e as NSEventMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.37 LeftMouseDragged(e as NSEventMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the left button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.38 LeftMouseUp(e as NSEventMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.39 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In older Xojo versions, this event is named `EnableMenuItems`.

9.23.40 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.23.41 `mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumn, nMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent to the subclass whenever the mouse button is clicked in the table view's header column.

Notes: tableColumn: The table column.

9.23.42 `MouseDrag(x as Integer, y as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.23.43 `MouseUp(x as Integer, y as Integer)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.23.44 `namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns an array of filenames that represent the indexSet rows for a drag to dropDestination.

Notes: dropDestination: The drop location where the files are created.

indexSet: The indexes of the items being dragged.

Returns an array of filenames (not full paths) for the created files that the receiver promises to create.

This method is called when a destination has accepted a promise drag.

For more information on file promise dragging, see documentation on the NSDraggingSource protocol and `namesOfPromisedFilesDroppedAtDestination`.

9.23.45 `nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to modify how type selection works.

Notes: startRow: The starting row of the search range.

endRow: The ending row of the search range.

searchString: A string containing the typed selection.

Return the first row in the range of startRow through endRow (excluding endRow itself) that matches searchString. Return -1 if no match is found.

It is possible for endRow to be less than startRow if the search will wrap.

Available in Mac OS X v10.5 and later.

9.23.46 `numberOfRowsInTableView as Integer`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called when the table view needs to know the number of rows.

Notes: numberOfRowsInTableView is called very frequently, so it must be efficient.

9.23.47 `objectValue(column as NSTableColumnMBS, row as Integer) as Variant`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called when a value is required for a given cell.

Notes: Please implement your own arrays to store values.

9.23.48 `OtherMouseDown(e as NSEventMBS) as boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed a mouse button other than the left or right one.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.49 `OtherMouseDragged(e as NSEventMBS) as boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with a button other than the left or right button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.50 `OtherMouseUp(e as NSEventMBS) as boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released a mouse button other than the left or right button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.51 `pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called to allow the table to support multiple item dragging.

Notes: row: The row.

Returns an instance of `NSPasteboardItem`. Returning `nil` excludes the row from being dragged.

This method is required for multi-image dragging.

If this method is implemented, then `writeRowsWithIndexes` will not be called.

This is `pasteboardWriterForRow` method.

9.23.52 `RightMouseDown(e as NSEventMBS)` as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the view that the user has pressed the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.53 `RightMouseDragged(e as NSEventMBS)` as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the right button pressed .

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.54 `RightMouseUp(e as NSEventMBS)` as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.23.55 `rowActionsForRow(row as Integer, edge as Integer)` as `NSTableViewRowActionMBS()`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the control to provide an array of row actions to be attached to the specified edge of a table row and displayed when the user swipes horizontally across the row.

Notes: `row`: The index of the target row.

`edge`: The edge (`NSTableRowActionEdgeLeading` or `NSTableRowActionEdgeTrailing`) for which row actions

are requested. This is based on the direction in which the user swiped on the row. Swiping to the right results in an edge value of leading. Swiping to the left results in an edge value of trailing.

Returns an array of row actions (of class `NSTableViewRowActionMBS`) to be enabled on the specified edge of the table row.

Implement this method if your table row supports actions that are displayed when the user swipes horizontally across the row. For example, your table view could use this method to implement a swipe left to delete function in your table rows. When called, this method receives the table view, the index of the row the user swiped, and an edge of type `NSTableRowActionEdge`. The method should return an array of any row actions of class `NSTableViewRowAction` that are supported for the specified edge. If no row actions are available, an empty array should be returned.

If this method isn't implemented, then the table row displays no actions when the user swipes horizontally away from the specified edge.

9.23.56 `rowViewForRow(row as Integer)` as `NSTableRowViewMBS`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the delegate for a view to display the specified row.

Notes: row: The row index.

Return an instance or subclass of `NSTableRowView`. If nil is returned, an `NSTableRowView` instance will be created and used.

You can implement this event to return a custom `NSTableRowView` for row.

The reuse queue can be used in the same way as documented in `tableView:view:row:`. The returned view will have attributes properly set to it before it's added to the `tableView`.

This method is only valid for `NSView`-based table views.

9.23.57 `ScaleFactorChanged(NewFactor as Double)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.23.58 SelectionDidChange(notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event informs you that the table view's selection has changed.

9.23.59 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to modify the proposed selection.

Notes: proposedSelectionIndexes: An index set containing the indexes of the proposed selection.

Return an NSIndexSet instance containing the indexes of the new selection. Return proposedSelectionIndexes if the proposed selection is acceptable, or the value of the table view's existing selection to avoid changing the selection.

This method may be called multiple times with one new index added to the existing selection to find out if a particular index can be selected when the user is extending the selection with the keyboard or mouse.

Implementation of this method is optional. If implemented, this method will be called instead of shouldSelectRow.

Available in Mac OS X v10.5 and later.

9.23.60 SelectionIsChanging(notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event informs you that the table view's selection is in the process of changing (typically because the user is dragging the mouse across a number of rows).

9.23.61 selectionShouldChangeInTableView as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns whether the selection should change.

Notes: Return true to allow the table view to change its selection (typically a row being edited), false to deny selection change.

The user can select and edit different cells within the same row, but can't select another row unless the

delegate approves. The subclass can implement this method for complex validation of edited rows based on the values of any of their cells.

9.23.62 setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called when a cell value is saved to the datasource.

9.23.63 shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns whether the cell at the specified row and column can be edited.

Notes: TableColumn: The table column.

rowIndex: The row index.

Return true to allow editing the cell, false to deny editing.

The subclass can implement this method to disallow editing of specific cells.

Note: This method is only valid for cell-based table views.

9.23.64 shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent to the subclass to allow or prohibit the specified column to be dragged to a new location.

Notes: columnIndex: The index of the column being dragged.

newColumnIndex: The proposed target index of the column.

Return true if the column reordering should be allowed, otherwise false.

When a column is initially dragged by the user, the delegate is first called with a newColumnIndex value of -1. Returning false will disallow that column from being reordered at all. Returning true allows it to be reordered, and the delegate will be called again when the column reaches a new location.

The actual NSTableColumn instance can be retrieved from the tableColumns array.

If this method is not implemented, all columns are considered reorderable.
Available in Mac OS X v10.6 and later.

9.23.65 shouldSelectRow(row as Int64) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns whether the table view should allow selection of the specified row.

Notes: rowIndex: The row index.

Return true to permit selection of the row, false to deny selection.

The delegate can implement this method to disallow selection of particular rows.
For better performance and finer-grain control over the selection, use selectionIndexesForProposedSelection.

9.23.66 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns whether the specified table column can be selected.

Notes: TableColumn: The table column.

Return true to permit selection, otherwise false.

The subclass can implement this event to disallow selection of particular columns.

9.23.67 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control cell expansion for a specific row and column.

Notes: TableColumn: The table column.

row: The row index.

Return true if the tooltip cell should expand, false otherwise.

Cell expansion can occur when the mouse hovers over the specified cell and the cell contents are unable to be fully displayed within the cell. If this method returns true, the full cell contents will be shown in a special

floating tool tip view, otherwise the content is truncated.

Note: This method is only valid for cell-based table views.

Available in Mac OS X v10.5 and later.

9.23.68 `shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control the tracking behavior for a specific cell.

Notes: cell: The cell to track.

tableColumn: The table column.

row: A row in tableView.

Returns true if the cell should track, false otherwise.

Normally, only selectable or selected cells can be tracked. If you implement this method, cells which are not selectable or selected can be tracked, and vice-versa.

For example, this allows you to have an NSButtonCell in a table which does not change the selection, but can still be clicked on and tracked.

Note: This method is only valid for cell-based table views.

Available in Mac OS X v10.5 and later.

9.23.69 `shouldTypeSelectForEvent(e as NSEventMBS, searchString as string) as Boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control type select for a specific event.

Notes: event: The event.

searchString: The search string or nil if no type select has begun.

Return true to allow type select for event, false otherwise.

Typically, this is called from the table view keyDown implementation and the event will be a key event. Available in Mac OS X v10.5 and later.

9.23.70 sizeToFitWidthOfColumn(column as Int64) as Double

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to provide custom sizing behavior when a column's resize divider is double clicked.

Notes: column: The index of the column.

Returns the width of the specified column.

By default, NSTableView iterates every row in the table, accesses a cell via preparedCellAtColumn, and requests the cellSize to find the appropriate largest width to use.

For accurate results and performance, it is recommended that this method is implemented when using large tables. By default, large tables use a monte carlo simulation instead of iterating every row.

Available in Mac OS X v10.6 and later.

9.23.71 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Called by TableView to indicate that sorting may need to be done.

Notes: The data source typically sorts and reloads the data, and adjusts the selections accordingly. If you need to know the current sort descriptors and the data source doesn't manage them itself, you can get the current sort descriptors by calling TableView.sortDescriptors function.

9.23.72 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow.

9.23.73 `textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow.

9.23.74 `toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a string that is displayed as a tooltip for the specified cell in the column and row.

Notes: Cell: The cell.

r: The proposed active area of the tooltip. You can modify rect to provide an alternative active area.

TableColumn: The table column.

row: The row index.

mouseLocation: The mouse location.

Return a string containing the tooltip. Return empty string if no tooltip is desired.

By default, rect is computed as `cell.drawingRectForBounds(cellFrame)`.

Available in Mac OS X v10.4 and later.

9.23.75 `typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to provide an alternate text value used for type selection for a specified row and column.

Notes: tableColumn: The table column.

row: The row index.

Returns a string that is used in type select comparison for row and tableColumn. Return "" if the row or tableColumn should not be searched.

Implement this method to change the string value that is searched for based on what is displayed. By default, all cells with text in them are searched.

If this event is not implemented the string value is the cell string value. Implementation of this event is optional.

Available in Mac OS X v10.5 and later.

9.23.76 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Implement this method to allow the table to update dragging items as they are dragged over a view.

Notes: The dragging information.

Required for multi-image dragging. Typically this will involve invoking `enumerateDraggingItemsWithOptions` on the `draggingInfo` parameter value and setting the `draggingItem` object, `imageComponentsProvider` to a proper image based on the content.

For view-based table views, you can use the `NSTableCellView` method `draggingImageComponents`. For cell-based tables, use the `NSCell` method `draggingImageComponentsWithFrame`.

9.23.77 validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer) as Integer

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Used by a `TableView` to determine a valid drop target.

Notes: `info`: An object that contains more information about this dragging operation.

`row`: The index of the proposed target row.

`operation`: The type of dragging operation proposed.

Returns the dragging operation the data source will perform.

The data source may retarget a drop by calling `setDropRow` and returning something other than `NSDraggingOperationNone`. A data source might retarget for various reasons, such as to provide better visual feedback when inserting into a sorted position.

To propose a drop on the second row, `row` would be 2 and `operation` would be `NSTableViewDropOn`. To propose a drop below the last row, `row` would be `TableView.numberOfRows` and `operation` would be `NSTableViewDropAbove`.

9.23.78 view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Asks the delegate for a view to display the specified row and column.

Notes: see also

<https://developer.apple.com/reference/appkit/nstableviewdelegate/1527449-tableview?language=objc>

See also:

- 9.23.15 View as NSTableViewMBS

413

9.23.79 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Informs you that the tableview will display the specified cell at the row in the column.

Notes: Cell: The cell to be displayed.

TableColumn: The table column.

row: The row index.

The event can modify the display attributes of cell to alter the appearance of the cell.

Because cell is reused for every row in tableColumn, the event must set the display attributes both when drawing special cells and when drawing normal cells.

Note: The implementation of this method must not draw portions of the cell. It should only alter the state of the passed in cell.

9.23.80 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.23.81 willTile

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The tableview will tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.23.82 writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a drag operation is allowed.

Notes: rowIndexes: An index set of row numbers that will be participating in the drag.

pboard: The pasteboard to which to write the drag data.

Returns true if the drag operation is allowed, false otherwise.

Called by TableView after it has been determined that a drag should begin, but before the drag has been started.

To refuse the drag, return false. To start a drag, return true and place the drag data onto pboard (data, owner, and so on). The drag image and other drag-related information will be set up and provided by the table view once this call returns with true.

9.24 control DesktopNSTokenFieldControlMBS

9.24.1 control DesktopNSTokenFieldControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The control to host a NSTokenField.

Blog Entries

- [MBS Xojo Plugins, version 24.1pr5](#)
- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo Plugins, version 21.4pr1](#)
- [MBS Xojo Plugins, version 21.1pr2](#)

9.24.2 Properties

9.24.3 View as NSTokenFieldMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The view used for the control.

Notes: (Read only property)

9.24.4 Events

9.24.5 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.24.6 completionsForSubstring(substring as string, tokenIndex as Integer, byref selectedIndex as Integer) as Variant()

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide an array of appropriate completions for the contents of the receiver.

Notes: substring: The partial string that is to be completed.

tokenIndex: The index of the token being edited.

selectedIndex: Optionally, you can return by-reference an index into the returned array that specifies which of the completions should be initially selected. If none are to be selected, return by reference -1.

Returns an array of strings that are possible completions.

If the delegate does not implement this method, no completions are provided.

Available in OS X v10.4 and later.

9.24.7 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.24.8 displayStringForRepresentedObject(representedObject as Variant) as string

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a string to be displayed as a proxy for the given represented object.

Notes: representedObject: A represented object of the token field.

Returns the string to be used as a proxy for representedObject. If you return nil or do not implement this method, then representedObject is displayed as the string.

9.24.9 editingStringForRepresentedObject(representedObject as Variant) as string

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a string to be edited as a proxy for a represented object.

Notes: representedObject: A represented object of the token field.

Returns a string that's an editable proxy of the represented object, or nil if the token should not be editable.

9.24.10 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named `LostFocus`.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.24.11 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named `GotFocus`.

Notes:

This only fires if the control itself got focus and not a sub control.

9.24.12 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.24.13 hasMenuForRepresentedObject(representedObject as Variant) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to specify whether the given represented object provides a menu.

Notes: `representedObject`: A represented object of the token field.

Returns true if the represented object has a menu, false otherwise.

By default tokens in a token field have no menus.

9.24.14 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.
In older Xojo versions, this event is named EnableMenuItems.

9.24.15 menuForRepresentedObject(representedObject as Variant) as NSMenuMBS

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a menu for the specified represented object.

Notes: representedObject: A represented object of the token field.

Returns the menu associated with the represented object.
By default tokens in a token field do not return menus.

9.24.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.24.17 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.24.18 MouseUp(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the `x` and `y` parameters to determine if the mouse button was released within the control's boundaries.

9.24.19 readFromPasteboard(pboard as NSPasteboardMBS) as Variant()

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to return an array of objects representing the data read from the specified pasteboard.

Notes: `pboard`: The pasteboard from which to read the represented objects.

Returns an array of represented objects created from the pasteboard data.

9.24.20 representedObjectForEditingString(editingString as string) as Variant

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to provide a represented object for the given editing string.

Notes: `editingString`: The edited string representation of a represented object.

Returns a represented object that is displayed rather than the editing string.

Note: In OS X v10.4, NSTokenField trims whitespace around tokens but it does not trim whitespace in OS X versions 10.5.0 and 10.5.1. In OS X v10.5.2, you get whitespace-trimming behavior by either linking against the v10.4 binary or linking against the v10.5 binary and not implementing the this method. If you do not want the whitespace-trimming behavior, link against the v10.5 binary and implement this method, returning the editing string if you have no represented object.

9.24.21 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.24.22 shouldAddObjects(tokens() as Variant, index as Integer) as Variant()

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows to validate the tokens to be added to the receiver at a particular location.

Notes: tokens: An array of tokens to be inserted in the receiver at index.

index: The index of the receiver in which the array of tokens to be validated (tokens) will be inserted.

Returns an array of validated tokens.

The event can return the array unchanged or return a modified array of tokens. To reject the add completely, return an empty array. Returning nil causes an error.

9.24.23 styleForRepresentedObject(representedObject as Variant) as Integer

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Allows the delegate to return the token style for editing the specified represented object.

Notes: representedObject: A represented object of the token field.

Returns the style that should be used to display the representedObject. Possible values are shown in NSTokenStyle Values.

If the event implements this method and returns an NSTokenStyle that differs from the style set by setTokenStyle:, the value the event returns is preferred.

If you don't implement this method, the token field's tokenStyle is used.

9.24.24 TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent when a control with editable text begins an editing session.

Notes: Notification: The notification object. The name of the notification is always NSControlTextDidBeginEditingNotification.

This event is invoked when the user begins editing text in a control such as a text field or a form field. The control posts a NSControlTextDidBeginEditingNotification notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is also delivered for inspection.

See TextDidEndEditing for an explanation of why you may not always get one invocation of TextDidBeginEditing for each invocation of TextDidEndEditing.

9.24.25 `TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent when the text in the receiving control changes.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidChangeNotification`.

This event is invoked when text in a control such as a text field or form changes. The control posts a `NSControlTextDidChangeNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is provided as parameter for inspection.

9.24.26 `TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent when a control with editable text ends an editing session.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidEndEditingNotification`.

This event is invoked when the user stops editing text in a control such as a text field or form. The control posts a `NSControlTextDidEndEditingNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is also provided for inspection.

Warning: In some cases, such as when editing within an instance of `NSOutlineView`, this method may be invoked without a previous invocation of `TextDidBeginEditing`. You will only get the `TextDidBeginEditing` notification if the user actually types something, but you can get the `TextDidEndEditing` notification if the user just double-clicks the field and then clicks outside the field, without typing.

9.24.27 `textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean`

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.24.28 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow end of text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.24.29 tokenFieldAction

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The control's action was triggered.

Notes: For a button if it was pressed.

9.24.30 tokenFieldTextShouldBeginEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing.

9.24.31 tokenFieldTextShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow text editing.

9.24.32 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.24.33 writeRepresentedObjects(objects() as Variant, pboard as NSPasteboardMBS) as boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Sent so the delegate can write represented objects to the pasteboard corresponding to a given array of display strings.

Notes: objects: An array of represented objects associated with the token field.

pboard: The pasteboard to which to write the represented objects.

Return true if you writes the represented objects to the pasteboard, false otherwise. If false, the token field writes the display strings to the NSStringPboardType pasteboard.

9.25 class DesktopPopupMenu

9.25.1 class DesktopPopupMenu

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Gives access to the NSMenuMBS of the PopupMenu.

9.25.2 Methods

9.25.3 NSButtonMBS as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox PopupMenu1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.25.4 NSPopUpButtonMBS as NSPopUpButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSPopUpButtonMBS object for the given control.

Example:

```
// get cocoa view for the popupmenu
dim p as NSPopUpButtonMBS = PopupMenu1.NSPopUpButtonMBS

// find a menu entry
dim it as NSMenuItemMBS = p.itemAtIndex(0)

// get a picture
dim pic as Picture = LogoMBS(500)
dim img as new NSImageMBS(pic)
img.setSize 16,16

// and assign icon
it.image = img
```

Notes: This way you can manipulate Cocoa controls directly.

9.26 class DesktopProgressWheel

9.26.1 class DesktopProgressWheel

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in ProgressWheel class in Xojo.

9.26.2 Methods

9.26.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSProgressIndicatorMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.27 class DesktopRadioButton

9.27.1 class DesktopRadioButton

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.27.2 Methods

9.27.3 NSButtonMBS as NSButtonMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox RadioButton1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.28 class DesktopRadioGroup

9.28.1 class DesktopRadioGroup

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.28.2 Methods

9.28.3 NSButtonsMBS as NSButtonMBS()

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Queries the buttons within a radio button group.

Example:

```
Dim buttons() As NSButtonMBS = RadioGroup1.NSButtonsMBS
```

```
For Each button As NSButtonMBS In buttons
```

```
button.alphaValue = 0.5 // half transparent
```

```
Next
```

```
Break
```

9.29 class DesktopScrollBar

9.29.1 class DesktopScrollBar

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in Scrollbar class in Xojo.

9.29.2 Methods

9.29.3 NSScrollerMBS as NSScrollerMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSScrollerMBS object for the given control.

Example:

```
MsgBox ScrollBar1.NSScrollerMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.30 class DesktopSeparator

9.30.1 class DesktopSeparator

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in Separator class in Xojo.

9.30.2 Methods

9.30.3 NSBoxMBS as NSBoxMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSBoxMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.31 class DesktopSlider

9.31.1 class DesktopSlider

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The built in slider class in Xojo.

9.31.2 Methods

9.31.3 NSSliderMBS as NSSliderMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSSliderMBS object for the given control.

Example:

```
MsgBox Slider1.NSSliderMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.32 class DesktopTextArea

9.32.1 class DesktopTextArea

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in textarea class in Xojo.

Example:

```
// make a PDF from a textarea in Cocoa Xojo target:

// find view
dim n as NSViewMBS = TextArea1.NSViewMBS
if n = nil then
  MsgBox "Only in Cocoa!"
  Return
end if

// make pdf data
dim s as string = n.dataWithPDFInsideRect(0,0,n.frame.Width, n.frame.Height)

// save
dim f as FolderItem = GetSaveFolderItem("", "test.pdf")

if f<>Nil then

  dim b as BinaryStream = BinaryStream.Create(f, true)

  b.Write s

end if
```

Notes: Requires Xojo 2021r3 or newer.

9.32.2 Methods

9.32.3 NSScrollViewMBS as NSScrollViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSScrollViewMBS object for the given control.

Example:

```
MsgBox TextArea1.NSScrollViewMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.33 class DesktopUpDownArrows

9.33.1 class DesktopUpDownArrows

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The build in control class in Xojo.

9.33.2 Methods

9.33.3 NSStepperMBS as NSStepperMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSStepperMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.34 class DisclosureTriangle

9.34.1 class DisclosureTriangle

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: The built in DisclosureTriangle class in Xojo.

9.34.2 Methods

9.34.3 NSButtonMBS as NSButtonMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox DisclosureTriangle1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.35 class Groupbox

9.35.1 class Groupbox

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The built in Groupbox class in Xojo.

9.35.2 Methods

9.35.3 NSBoxMBS as NSBoxMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSBoxMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.36 class ImageWell

9.36.1 class ImageWell

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The built in ImageWell class in Xojo.

9.36.2 Methods

9.36.3 NSImageViewMBS as NSImageViewMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSImageViewMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.37 class KeyValueCollectionMBS

9.37.1 class KeyValueCollectionMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The class for key value coding and using NSSortDescriptor class.

Notes: Events are only raised if called on main thread to avoid threading trouble.

Blog Entries

- [MBS Xojo Plugins, version 17.1pr3](#)

9.37.2 Methods

9.37.3 Constructor

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The constructor.

9.37.4 sortedArrayUsingDescriptor(values() as KeyValueCollectionMBS, sortDescriptor as NSSortDescriptorMBS) as KeyValueCollectionMBS()

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a copy of the receiving array sorted as specified by a given a sort descriptor.

Notes: sortDescriptor: The NSSortDescriptor objects.

Returns a copy of the receiving array sorted as specified by sortDescriptor.

9.37.5 sortedArrayUsingDescriptors(values() as KeyValueCollectionMBS, sortDescriptor() as NSSortDescriptorMBS) as KeyValueCollectionMBS()

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a copy of the receiving array sorted as specified by a given array of sort descriptors.

Notes: sortDescriptors: An array of NSSortDescriptor objects.

Returns a copy of the receiving array sorted as specified by sortDescriptors.

The first descriptor specifies the primary key path to be used in sorting the receiving array,Ãs contents. Any subsequent descriptors are used to further refine sorting of objects with duplicate values. See NSSortDescriptor for additional information.

9.37.6 Properties

9.37.7 Description as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The description text property.

Notes: (Read only property)

See also:

- 9.37.12 Description as String

459

9.37.8 Handle as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The internal object handle.

Notes: (Read and Write property)

9.37.9 Tag as Variant

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The tag value.

Notes: Store anything you need, e.g. a dictionary.

(Read and Write property)

9.37.10 valueForKey(key as String) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Get/set the value for the property identified by a given key.

Notes: key: The name of one of the receiver's properties.

(Read and Write computed property)

See also:

- 9.37.15 valueForKey(key as string) as Variant

459

9.37.11 Events**9.37.12 Description as String**

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event when system queries description for object.

See also:

- 9.37.7 Description as String

458

9.37.13 setValueForKey(key as string, value as Variant)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Sets the property of the receiver specified by a given key to a given value.

Notes: value: The value for the property identified by key.

key: The name of one of the receiver's properties.

If key identifies a to-one relationship, relate the object specified by value to the receiver, unrelating the previously related object if there was one. Given a collection object and a key that identifies a to-many relationship, relate the objects contained in the collection to the receiver, unrelating previously related objects if there were any.

The search pattern that setValueForKey uses is described in Accessor Search Patterns in Key-Value Coding Programming Guide.

In a reference-counted environment, if the instance variable is accessed directly, value is retained.

9.37.14 setValueForUndefinedKey(key as string, value as Variant)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by setValueForKey when it finds no property for a given key.

Notes: value: The value for the key identified by key.

key: A string that is not equal to the name of any of the receiver's properties.

Subclasses can override this method to handle the request in some other way. The default implementation raises an NSUndefinedKeyException.

9.37.15 valueForKey(key as string) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Return the value for the property identified by a given key.

Notes: key: The name of one of the receiver's properties.

Returns the value for the property identified by key.

If event is not implemented, the search pattern that `valueForKey` uses to find the correct value to return is described in Accessor Search Patterns in Key-Value Coding Programming Guide.

See also:

- 9.37.10 `valueForKey(key as String) as Variant` 458

9.37.16 `valueForKey(key as string) as Variant`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by `valueForKey` when it finds no property corresponding to a given key.

Notes: key: A string that is not equal to the name of any of the receiver's properties.

Subclasses can override this method to return an alternate value for undefined keys. The default implementation raises an `NSUndefinedKeyException`.

9.38 class NSBoxMBS

9.38.1 class NSBoxMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The NSBox class implements simple views that can title themselves and draw a border around their content.

Notes: These objects are known as boxes. You can use box to group, visually, some number of other views.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard.

Subclass of the NSViewMBS class.

Blog Entries

- [MBS Plugins 10.3 Release Notes](#)
- [MBS REALbasic Plugins, version 10.3pr5](#)

9.38.2 Methods

9.38.3 borderRect as NSRectMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle in which the receiver's border is drawn.

Notes: The rectangle in which the border of the NSBox is drawn.

9.38.4 Constructor

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates a new box view with size 100/100 and position 0/0

Example:

```
dim x as new NSBoxMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.38.5 Constructor(Handle as Integer) 462
- 9.38.6 Constructor(left as Double, top as Double, width as Double, height as Double) 462

9.38.5 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSView handle.

Example:

```
dim t as new NSBoxMBS(0, 0, 100, 100)
dim v as new NSBoxMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSBox and the plugin retains this handle.

See also:

- 9.38.4 Constructor 461
- 9.38.6 Constructor(left as Double, top as Double, width as Double, height as Double) 462

9.38.6 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates a new control with the given size and position.

Example:

```
dim left,top,width,height as Integer
// define rectangle
dim x as new NSBoxMBS(left, top, width, height)
```

Notes: On success the handle property is not zero.

See also:

- 9.38.4 Constructor 461
- 9.38.5 Constructor(Handle as Integer) 462

9.38.7 setFrameFromContentFrame(contentFrame as NSRectMBS)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Places the receiver so its content view lies on the specified frame.

Notes: contentFrame: The rectangle specifying the frame of the box's content view, reckoned in the coor-

dinate system of the box's superview. The box is marked for redisplay.

9.38.8 setTitleWithMnemonic(stringWithAmpersand as string)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Sets the title of the receiver with a character denoted as an access key.

Notes: Mnemonics are not supported in Mac OS X.

By default, a box's title is "Title." The content view is not automatically resized, and the box is not marked for redisplay.

9.38.9 sizeToFit

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Resizes and moves the receiver's content view so it just encloses its subviews.

Example:

```
dim n as new NSProgressIndicatorMBS
n.sizeToFit
```

```
MsgBox n.frame.String
```

Notes: The receiver is then moved and resized to wrap around the content view. The receiver's width is constrained so its title will be fully displayed.

You should invoke this method after:

- Adding a subview (to the content view)
- Altering the size or location of such a subview
- Setting the margins around the content view

The mechanism by which the content view is moved and resized depends on whether the object responds to its own `sizeToFit` message: If it does respond, then that message is sent, and the content view is expected to be so modified. If the content view doesn't respond, the box moves and resizes the content view itself.

9.38.10 titleCell as NSCellMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the cell used to display the receiver's title.

9.38.11 titleRect as NSRectMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle in which the receiver's title is drawn.

Notes: The rectangle in which the title is drawn.

9.38.12 Properties

9.38.13 borderColor as NSColorMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the color of the receiver's border when the receiver is a custom box with a simple line border.

Notes: The receiver's border color. It must be a custom box—that is, it has a type of `NSBoxCustom`—and it must have a border style of `NSLineBorder`.

Available in Mac OS X v10.5 and later.
(Read and Write computed property)

9.38.14 borderType as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The border type to aType, which must be a valid border type.

Notes: A constant describing the type of border. Border types are defined in `NSView.h`. Currently, the following border types are defined: `NSNoBorder`, `NSLineBorder`, `NSBezelBorder`, `NSGrooveBorder`.

If the size of the new border is different from that of the old border, the content view is resized to absorb the difference, and the box is marked for redisplay.
(Read and Write computed property)

9.38.15 `borderWidth` as Double

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The border width.

Notes: Functional only when the receiver's box type (`boxType`) is `NSBoxCustom` and its border type (`borderType`) is `NSLineBorder`.

Available in Mac OS X v10.5 and later.
(Read and Write computed property)

9.38.16 `boxType` as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The box type.

Notes: Use the `NSBox*` constants.
(Read and Write computed property)

9.38.17 `contentView` as `NSViewMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The receiver's content view.

Notes: On settings the `NSView` object is resized to fit within the box's current content area and the box is marked for redisplay.
(Read and Write computed property)

9.38.18 `contentViewMargins` as `NSSizeMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The horizontal and vertical distance between the border of the receiver and its content view.

Notes: Value: The width and height of the offset between the box's border and content view. The horizontal value is applied (reckoned in the box's coordinate system) fully and equally to the left and right sides of the box. The vertical value is similarly applied to the top and bottom.

Unlike changing a box's other attributes, such as its title position or border type, changing the offsets doesn't automatically resize the content view. In general, you should send a `sizeToFit` message to the box after changing the size of its offsets. This message causes the content view to remain unchanged while the box is sized to fit around it.

(Read and Write computed property)

9.38.19 `cornerRadius` as `Double`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The corner radius for the box.

Notes: Functional only when the receiver's box type (`boxType`) is `NSBoxCustom` and its border type (`borderType`) is `NSLineBorder`.

(Read and Write computed property)

9.38.20 `fillColor` as `NSColorMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The color of the receiver's background when the receiver is a custom box with a simple line border.

Notes: The receiver's fill color. It must be a custom box—that is, it has a type of `NSBoxCustom`—and it must have a border style of `NSLineBorder`.

Available in Mac OS X v10.5 and later.

(Read and Write computed property)

9.38.21 `title` as `string`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The title for the box.

Notes: By default, a box's title is "Title."

(Read and Write computed property)

9.38.22 `titleFont` as `NSFontMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The font object used to draw the receiver's title.

Notes: By default, the title is drawn using the small system font (obtained using (`smallSystemFontSize` as the parameter of `systemFontOfSize:`, both `NSFont` class methods). If the size of the new font is different from that of the old font, the content view is resized to absorb the difference.

(Read and Write computed property)

9.38.23 titlePosition as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The position of the box's title.

Notes: If the new title position changes the size of the box's border area, the content view is resized to absorb the difference, and the box is marked as needing redisplay.

Use this constants: NSNoTitle, NSAboveTop, NSAtTop, NSBelowTop, NSAboveBottom, NSAtBottom and NSBelowBottom.

(Read and Write computed property)

9.38.24 Transparent as Boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver is transparent.

Notes: True makes the receiver transparent.

False makes the receiver opaque.

(Read and Write computed property)

9.38.25 Constants

Constants

Constant	Value	Description
NSAboveBottom	4	One of the constants for the title position. Title positioned above the box's bottom border.
NSAboveTop	1	One of the constants for the title position. Title positioned above the box's top border.
NSAtBottom	5	One of the constants for the title position. Title positioned within the box's bottom border.
NSAtTop	2	One of the constants for the title position. Title positioned within the box's top border.
NSBelowBottom	6	One of the constants for the title position. Title positioned below the box's bottom border.
NSBelowTop	3	One of the constants for the title position. Title positioned below the box's top border.
NSBoxCustom	4	One of the constants and data type identifies box types, which, in conjunction with a box's border type, define the appearance of the box. Specifies that the appearance of the box is determined entirely by the by box-configuration methods, without automatically applying Apple human interface guidelines.
NSBoxOldStyle	3	One of the constants and data type identifies box types, which, in conjunction with a box's border type, define the appearance of the box. Specifies that the box is a Mac OS X v10.2-style box.
NSBoxPrimary	0	One of the constants and data type identifies box types, which, in conjunction with a box's border type, define the appearance of the box. Specifies the primary box appearance. This is the default box type.
NSBoxSecondary	1	One of the constants and data type identifies box types, which, in conjunction with a box's border type, define the appearance of the box. Specifies the secondary box appearance.
NSBoxSeparator	2	One of the constants and data type identifies box types, which, in conjunction with a box's border type, define the appearance of the box. Specifies the secondary box appearance.
NSNoTitle	0	One of the constants for the title position. The box has no title.

9.39 control NSButtonControlMBS

9.39.1 control NSButtonControlMBS

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The Xojo control for a NSButton.

Notes: This control embeds a special NSButton subclass.

Designed for Xojo 2013r1 and newer. May work on Xojo 2012, but not perfectly.

Please use view property to access the underlying object and set properties.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo / Real Studio Plugins, version 13.2pr6](#)

9.39.2 Properties

9.39.3 AlternateTitle as String

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The title that the button displays when it's in its alternate state.

Notes: (Read and Write property)

9.39.4 BezelStyle as Integer

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The appearance of the border, if the view has one.

Notes: Use this constants:

(Read and Write property)

9.39.5 ButtonType as Integer

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The button type.

Notes: See also NSButtonMBS.ButtonType and the constants there like NSPushOnPushOffButton.

(Read and Write property)

NSRoundedBezelStyle	= 1
NSRegularSquareBezelStyle	= 2
NSThickSquareBezelStyle	= 3
NSThickerSquareBezelStyle	= 4
NSDisclosureBezelStyle	= 5
NSShadowlessSquareBezelStyle	= 6
NSCircularBezelStyle	= 7
NSTexturedSquareBezelStyle	= 8
NSHelpButtonBezelStyle	= 9
NSSmallSquareBezelStyle	= 10
NSTexturedRoundedBezelStyle	= 11
NSRoundRectBezelStyle	= 12
NSRecessedBezelStyle	= 13
NSRoundedDisclosureBezelStyle	= 14

9.39.6 Title as String

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The title displayed on the button when it's in its normal state.

Notes: (Read and Write property)

9.39.7 View as NSButtonMBS

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.
(Read only property)

9.39.8 Events

9.39.9 Action

Plugin Version: 13.2, Platform: macOS, Targets: .

Function: The action event.

9.39.10 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.39.11 Close

Plugin Version: 13.2, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.39.12 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.39.13 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.39.14 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.39.15 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.
In Xojo version 2021r3 and newer this event is named `MenuBarSelected`.

9.39.16 `FrameChanged`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.39.17 `GotFocus`

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named `FocusReceived`.

Notes:

This only fires if the control itself got focus and not a sub control.

9.39.18 `LostFocus`

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named `FocusLost`.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.39.19 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the `MouseDown`. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.39.20 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.39.21 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.39.22 Open

Plugin Version: 13.2, Platform: macOS, Targets: .

Function:

The control is about to was created and you can initialize it.

In Xojo version 2021r3 and newer this event is named Opening.

9.39.23 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.39.24 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.40 class NSButtonMBS

9.40.1 class NSButtonMBS

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The Cocoa class for a button control.

Notes: Subclass of the NSControlMBS class.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 22.5](#)
- [MBS Xojo Plugins, version 22.5pr1](#)
- [MBS Xojo / Real Studio Plugins, version 16.5pr5](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr3](#)
- [MBS Xojo / Real Studio Plugins, version 14.2pr8](#)
- [Geoff's Cocoa update](#)

Xojo Developer Magazine

- [15.5, page 35: Touch/©, Using Apple's Touch Bar with Xojo by Marc Zeedar](#)

9.40.2 Methods

9.40.3 Constructor

Plugin Version: 8.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with size 100/100 and position 0/0

Example:

```
dim t as new NSButtonMBS
```

Notes: On success the handle property is not zero.

See also:

- [9.40.4 Constructor\(Handle as Integer\)](#) 476
- [9.40.5 Constructor\(left as Double, top as Double, width as Double, height as Double\)](#) 476
- [9.40.6 Constructor\(Title as String, Image as NSImageMBS = nil, Type as Integer = 0\)](#) 476

9.40.4 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSButton handle.

Example:

```
dim t as new NSButtonMBS(0, 0, 100, 100)
dim v as new NSButtonMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSButton and the plugin retains this handle.

See also:

- 9.40.3 Constructor 475
- 9.40.5 Constructor(left as Double, top as Double, width as Double, height as Double) 476
- 9.40.6 Constructor(Title as String, Image as NSImageMBS = nil, Type as Integer = 0) 476

9.40.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with the given size and position.

Example:

```
dim x as new NSButtonMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.40.3 Constructor 475
- 9.40.4 Constructor(Handle as Integer) 476
- 9.40.6 Constructor(Title as String, Image as NSImageMBS = nil, Type as Integer = 0) 476

9.40.6 Constructor(Title as String, Image as NSImageMBS = nil, Type as Integer = 0)

Plugin Version: 16.5, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with default settings.

Notes: For macOS 10.12, we use the convenience functions from Apple. For older systems, we use our own code.

Title is the title to use. Can be empty.

Image is the image to use. Can be nil.

Type is the type, e.g. NSMomentaryLightButton.

See also:

- 9.40.3 Constructor 475
- 9.40.4 Constructor(Handle as Integer) 476
- 9.40.5 Constructor(left as Double, top as Double, width as Double, height as Double) 476

9.40.7 setButtonType(buttonType as Integer)

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Sets how the button highlights while pressed and how it shows its state.

Notes: Use this constants:

NSMomentaryLightButton	= 0
NSPushOnPushOffButton	= 1
NSToggleButton	= 2
NSSwitchButton	= 3
NSRadioButton	= 4
NSMomentaryChangeButton	= 5
NSOnOffButton	= 6
NSMomentaryPushInButton	= 7

9.40.8 setNextState

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Sets the view to its next state.

Notes: If the button has three states, it cycles through them in this order: on, off, mixed, on, and so forth. If the button has two states, it toggles between them.

9.40.9 Properties

9.40.10 allowsMixedState as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the button allows a mixed state.

Notes: (Read and Write property)

9.40.11 `alternateImage` as `NSImageMBS`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The image that appears on the button when it's in its alternate state.

Notes: The image displayed by the button when it's in its alternate state, or nil if there is no alternate image. Note that some button types don't display an alternate image. Buttons don't display images by default.

(Read and Write property)

9.40.12 `alternateTitle` as string

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The title that the button displays when it's in its alternate state.

Notes: (Read and Write property)

9.40.13 `attributedAlternateTitle` as `NSAttributedStringMBS`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The title that appears on the button when it's in its alternate state to the given attributed string.

Notes: (Read and Write property)

9.40.14 `attributedTitle` as `NSAttributedStringMBS`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The title that the button displays in its normal state as an attributed string.

Notes: (Read and Write property)

9.40.15 `backgroundColor` as `NSColorMBS`

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: The background color.

Notes: The background color is used only when drawing borderless buttons.
(Read and Write property)

9.40.16 bezelColor as NSColorMBS

Plugin Version: 16.5, Platform: macOS, Targets: Desktop only.

Function: The color of the button's bezel, in appearances that support it.

Notes: (Read and Write property)

9.40.17 bezelStyle as Integer

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The appearance of the border, if the view has one.

Notes: Use this constants:

NSRoundedBezelStyle	= 1
NSRegularSquareBezelStyle	= 2
NSThickSquareBezelStyle	= 3
NSThickerSquareBezelStyle	= 4
NSDisclosureBezelStyle	= 5
NSShadowlessSquareBezelStyle	= 6
NSCircularBezelStyle	= 7
NSTexturedSquareBezelStyle	= 8
NSHelpButtonBezelStyle	= 9
NSSmallSquareBezelStyle	= 10
NSTexturedRoundedBezelStyle	= 11
NSRoundRectBezelStyle	= 12
NSRecessedBezelStyle	= 13
NSRoundedDisclosureBezelStyle	= 14

(Read and Write property)

9.40.18 hasDestructiveAction as Boolean

Plugin Version: 22.5, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that defines whether a button's action has a destructive effect.

Notes: Available on macOS 11.0 and newer.

The default value of `hasDestructiveAction` is `false`. Setting this to `true` allows the system to guard a destructive-action button against accidental presses, and can give the button a special appearance in certain contexts to caution against unintentional use.

(Read and Write property)

9.40.19 `image` as `NSImageMBS`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The view's image.

Notes: A button's image is displayed when the button is in its normal state, or all the time for a button that doesn't change its contents when highlighting or displaying its alternate state.

(Read and Write property)

9.40.20 `imageDimsWhenDisabled` as `Boolean`

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver's image appears "dim" when the button cell is disabled.

Notes: True to indicate that the button's image should dim when the button is disabled.

By default, all button types except `NSSwitchButton` and `NSRadioButton` do dim when disabled. When `NSSwitchButtons` and `NSRadioButtons` are disabled, only the associated text dims. The default setting for this condition is reasserted whenever you set `ButtonType`, so be sure to specify the button cell's type before you set `ImageDimsWhenDisabled`.

(Read and Write property)

9.40.21 `imageHugsTitle` as `Boolean`

Plugin Version: 16.5, Platform: macOS, Targets: Desktop only.

Function: Whether image hugs title.

Notes: Available in macOS 10.12 and newer.

(Read and Write property)

9.40.22 `imagePosition` as `Integer`

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: The position of the button's image relative to its title.

Notes: Value can be:

NSNoImage	0	The cell doesn't display an image.
NSImageOnly	1	The cell displays an image, but not a title.
NSImageLeft	2	The image is to the left of the title.
NSImageRight	3	The image is to the right of the title.
NSImageBelow	4	The image is below the title.
NSImageAbove	5	The image is above the title.
NSImageOverlaps	6	The image overlaps the title.

(Read and Write property)

9.40.23 imageScaling as Integer

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: The scale factor for the receiver's image.

Notes: These constants specify a cell's image scaling behavior.

NSImageScaleProportionallyDown	0	If it is too large for the destination, scale the image down while preserving the aspect ratio.
NSImageScaleAxesIndependently	1	Scale each dimension to exactly fit destination. This setting does not preserve the aspect ratio of the image.
NSImageScaleNone	2	Do not scale the image.
NSImageScaleProportionallyUpOrDown	3	Scale the image to its maximum possible dimensions while both staying within the destination area and preserving its aspect ratio.

Available in OS X v10.5 and later.

(Read and Write property)

9.40.24 isBordered as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the button has a border.

Notes: (Read and Write property)

9.40.25 isTransparent as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the button is transparent.

Notes: (Read and Write property)

9.40.26 keyEquivalent as string

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The key-equivalent character of the view.

Notes: (Read and Write property)

9.40.27 keyEquivalentModifierMask as Integer

Plugin Version: 8.0, Platform: macOS, Targets: Desktop only.

Function: the controls's keyboard equivalent modifier mask.

Example:

```
const NSShiftKeyMask=131.072
const NSControlKeyMask=262144
const NSAlternateKeyMask=524288
const NSCommandKeyMask=1048576
```

```
dim d as NSButtonMBS // your button
d.KeyEquivalent="A"
d.KeyEquivalentModifierMask=NSShiftKeyMask+NSCommandKeyMask+NSAlternateKeyMask // command-
option-shift
```

Notes: Constants for the mask:

NSAlphaShiftKeyMask = 65536
Set if Caps Lock key is pressed.
Available in Mac OS X v10.0 and later.

NSShiftKeyMask = 131072
Set if Shift key is pressed.
Available in Mac OS X v10.0 and later.

NSControlKeyMask = 262144
Set if Control key is pressed.
Available in Mac OS X v10.0 and later.

NSAlternateKeyMask = 524288

Set if Option or Alternate key is pressed.
Available in Mac OS X v10.0 and later.

NSCommandKeyMask = 1048576

Set if Command key is pressed.
Available in Mac OS X v10.0 and later.

NSNumericPadKeyMask = 2097152

Set if any key in the numeric keypad is pressed. The numeric keypad is generally on the right side of the keyboard. This is also set if any of the arrow keys are pressed (NSUpArrowFunctionKey, NSDownArrowFunctionKey, NSLeftArrowFunctionKey, and NSRightArrowFunctionKey).
Available in Mac OS X v10.0 and later.

NSHelpKeyMask = 4194304

Set if the Help key is pressed.
Available in Mac OS X v10.0 and later.

NSFunctionKeyMask = 8388608

Set if any function key is pressed. The function keys include the F keys at the top of most keyboards (F1, F2, and so on) and the navigation keys in the center of most keyboards (Help, Forward Delete, Home, End, Page Up, Page Down, and the arrow keys).
Available in Mac OS X v10.0 and later.

NSDeviceIndependentModifierFlagsMask = 16777216

Used to retrieve only the device-independent modifier flags, allowing applications to mask off the device-dependent modifier flags, including event coalescing information.
Available in Mac OS X v10.4.
(Read and Write property)

9.40.28 maxAcceleratorLevel as Integer

Plugin Version: 15.2, Platform: macOS, Targets: Desktop only.

Function: Configures the maximum allowed level for an NSMultiLevelAcceleratorButton, allowed values range from [1,5] .

Notes: Defaults to 2.

Available on Mac OS X 10.10.3.
(Read and Write property)

9.40.29 `showsBorderOnlyWhileMouseInside` as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Whether the view's border is displayed only when the cursor is over the button.

Notes: If `isBordered` returns false, the border is never displayed, regardless of what this method returns.
(Read and Write property)

9.40.30 `sound` as Variant

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: The sound played when the user presses the button.

Notes: The sound that should be played when the user presses the button. The sound is played during a mouse-down event, such as `NSLeftMouseDown`.

Value is `NSSoundMBS` object.

(Read and Write property)

9.40.31 `SpringLoaded` as Boolean

Plugin Version: 15.2, Platform: macOS, Targets: Desktop only.

Function: Sends action on deep-press or extended hover while dragging.

Notes: Defaults to false.

Available on Mac OS X 10.10.3.

(Read and Write property)

9.40.32 `state` as Integer

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The state of the button.

Notes: This can be `NSOnState` (1), `NSOffState` (0), `NSMixedState` (-1).

(Read and Write property)

9.40.33 `title` as string

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The title displayed on the button when it's in its normal state.

Notes: (Read and Write property)

9.40.34 Constants

Constants

Constant	Value	Description
NSCircularBezelStyle	7	One of the button styles that can be specified using <code>bezelStyle</code> . A round button with room for a small icon or a single character. This style has both regular and small variants, but the large variant is available only in gray at this time.
NSDisclosureBezelStyle	5	One of the button styles that can be specified using <code>bezelStyle</code> . A bezel style for use with a disclosure triangle. To create the disclosure triangle, set the button bezel style to <code>NSDisclosureBezelStyle</code> and the button type to <code>NSOnOffButton</code> .
NSHelpButtonBezelStyle	9	One of the button styles that can be specified using <code>bezelStyle</code> . A round button with a question mark providing the standard help button look.
NSInlineBezelStyle	15	One of the button styles that can be specified using <code>bezelStyle</code> . Inline Style.
NSMomentaryChangeButton	5	One of the button types that can be specified using <code>ButtonType</code> . While the button is held down, the alternate image and alternate title are displayed. The normal image and title are displayed when the button isn't pressed. This option is called "Momentary Change" in Interface Builder's Button Inspector.
NSMomentaryLightButton	0	One of the button types that can be specified using <code>ButtonType</code> . While the button is held down it's shown as "lit," and also "pushed in" to the screen if the button is bordered. This type of button is best for simply triggering actions, as it doesn't show its state; it always displays its normal image or title. This option is called "Momentary Light" in Interface Builder's Button Inspector.
NSMomentaryPushInButton	7	One of the button types that can be specified using <code>ButtonType</code> . While the button is held down it's shown as "lit." This type of button is best for simply triggering actions, as it doesn't show its state; it always displays its normal image or title. This option is called "Momentary Push In" in Interface Builder's Button Inspector.
NSOnOffButton	6	One of the button types that can be specified using <code>ButtonType</code> . The first click highlights the button; a second click returns it to the normal (unhighlighted) state.
NSPushOnPushOffButton	1	One of the button types that can be specified using <code>ButtonType</code> . The first click both highlights and causes the button to be "pushed in" if the button is bordered; a second click returns it to its normal state.
NSRadioButton	4	One of the button types that can be specified using <code>ButtonType</code> . This style is similar to <code>NSSwitchButton</code> , but it used to constrain a selection to a single element from several.
NSRecessedBezelStyle	13	One of the button styles that can be specified using <code>bezelStyle</code> . A bezel style that matches the recessed buttons in Mail, Finder and Safari.
NSRegularSquareBezelStyle	2	One of the button styles that can be specified using <code>bezelStyle</code> . A rectangular button with a 2 point border, designed for icons.
NSRoundedBezelStyle	1	One of the button styles that can be specified using <code>bezelStyle</code> . A rounded rectangle button, designed for text.
NSRoundedDisclosureBezelStyle	14	One of the button styles that can be specified using <code>bezelStyle</code> . A bezel style that matches the disclosure style used in the standard Save panel.
NSRoundRectBezelStyle	12	One of the button styles that can be specified using <code>bezelStyle</code> . A bezel style that matches the search buttons in Finder and Mail.
NSShadowlessSquareBezelStyle	6	One of the button styles that can be specified using <code>bezelStyle</code> . Similar to <code>NSRegularSquareBezelStyle</code> , but has no shadow so you can abut the cells without overlapping shadows. This style would be used in a tool palette, for example.
NSSmallSquareBezelStyle	10	One of the button styles that can be specified using <code>bezelStyle</code> . A simple square bezel style. Buttons using this style can be scaled to any size.
NSSwitchButton	3	One of the button types that can be specified using <code>ButtonType</code> . This style is a variant of <code>NSToggleButton</code> that has no border and is used to represent a checkbox.
NSTexturedRoundedBezelStyle	11	One of the button styles that can be specified using <code>bezelStyle</code> . A textured (metal) bezel style similar in appearance to the Finder's action

9.41 class NSClipViewMBS

9.41.1 class NSClipViewMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: An NSClipView contains and scrolls the document view displayed by an NSScrollView.

Notes: You normally don't need to program with NSClipViews, as NSScrollView handles most of the details of their operation.

An NSClipView holds the document view of an NSScrollView, clipping the document view to its frame, handling the details of scrolling in an efficient manner, and updating the NSScrollView when the document view's size or position changes. You don't normally use the NSClipView class directly; it's provided primarily as the scrolling machinery for the NSScrollView class. However, you might use the NSClipView class to implement a class similar to NSScrollView.

Interaction With NSScrollView

When using an NSClipView within an NSScrollView (the usual configuration), you should issue messages that control background drawing state to the NSScrollView, rather than messaging the NSClipView directly. This recommendation applies to the following methods:

- backgroundColor
- drawsBackground

The NSClipView methods are intended for when the NSClipView is used independently of a containing NSScrollView. In the usual case, NSScrollView should be allowed to manage the background-drawing properties of its associated NSClipView.

There is only one background-drawing state per NSScrollView/NSClipView pair. The two objects do not maintain independent and distinct drawsBackground and backgroundColor properties; rather, NSScrollView's accessors for these properties largely defer to the associated NSClipView and allow the NSClipView to maintain the state. In Mac OS X v10.2 and earlier system versions, NSScrollView maintained a cache of the last state it set for its NSClipView. If the NSClipView was sent a setDrawsBackground: message directly, the cache might not reflect the state accurately. This caching of state has been removed in Mac OS X v10.3.

It is also important to note that sending a setDrawsBackground message with a parameter of false to an NSScrollView has the added effect of sending the NSClipView a setCopiesOnScroll message with a parameter of false. The side effect of sending the setDrawsBackground message directly to the NSClipView is the appearance of "trails" (vestiges of previous drawing) in the document view as it is scrolled.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard. Subclass of the NSViewMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr1](#)

9.41.2 Methods**9.41.3 autoscroll(theEvent as NSEventMBS) as boolean**

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Scrolls the receiver proportionally to theEvent’s distance outside of it.

Notes: theEvent’s location should be expressed in the window’s base coordinate system (which it normally is), not the receiving NSClipView’s. Returns true if any scrolling is performed; otherwise returns false.

Never invoke this method directly; instead, the NSClipView’s document view should repeatedly send itself autoscroll: messages when the cursor is dragged outside the NSClipView’s frame during a modal event loop initiated by a mouse-down event. The NSView class implements autoscroll to forward the message to the receiver’s superview; thus the message is ultimately forwarded to the NSClipView.

9.41.4 constrainScrollPoint(newOrigin as NSPointMBS) as NSPointMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns a scroll point adjusted from proposedNewOrigin, if necessary, to guarantee the receiver will still lie within its document view.

Notes: For example, if proposedNewOrigin’s y coordinate lies to the left of the document view’s origin, then the y coordinate returned is set to that of the document view’s origin.

9.41.5 Constructor

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Creates a new clipview with the given size and position.

Example:

```
dim x as new NSClipViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.41.6 Constructor(Handle as Integer) 489
- 9.41.7 Constructor(left as Double, top as Double, width as Double, height as Double) 489

9.41.6 Constructor(Handle as Integer)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSClipView handle.

Example:

```
dim t as new NSClipViewMBS(0, 0, 100, 100)
dim v as new NSClipViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSClipView and the plugin retains this handle.

See also:

- 9.41.5 Constructor 488
- 9.41.7 Constructor(left as Double, top as Double, width as Double, height as Double) 489

9.41.7 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Creates a new NSClipView with the given size and position.

Example:

```
dim x as new NSClipViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.41.5 Constructor 488
- 9.41.6 Constructor(Handle as Integer) 489

9.41.8 documentRect as NSRectMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle defining the document view's frame, adjusted to the size of the receiver if the document view is smaller.

Notes: In other words, this rectangle is always at least as large as the receiver itself.

The document rectangle is used in conjunction with an `NSClipView`'s bounds rectangle to determine values for the indicators of relative position and size between the `NSClipView` and its document view. For example, `NSScrollView` uses these rectangles to set the size and position of the knobs in its scrollers. When the document view is much larger than the `NSClipView`, the knob is small; when the document view is near the same size, the knob is large; and when the document view is the same size or smaller, there is no knob.

9.41.9 `documentVisibleRect` as `NSRectMBS`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the exposed rectangle of the receiver's document view, in the document view's own coordinate system.

Notes: Note that this rectangle doesn't reflect the effects of any clipping that may occur above the `NSClipView` itself. To get the portion of the document view that's guaranteed to be visible, send it a `visibleRect` message.

9.41.10 `reflectScrolledClipView(clipView as NSClipViewMBS)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Adjusts the receiver's scrollers to reflect the size and positioning of its content view.

9.41.11 `scrollClipView(clipview as NSClipViewMBS, toPoint as NSPointMBS)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Notifies the superview of a clip view that the clip view needs to reset the origin of its bounds rectangle.

9.41.12 `scrollToPoint(newOrigin as NSPointMBS)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Changes the origin of the receiver's bounds rectangle to `newOrigin`.

9.41.13 `viewBoundsChanged(notification as NSNotificationMBS)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Handles an `NSViewBoundsDidChangeNotification`, passed in the `aNotification` argument, by updating a containing `NSScrollView` based on the new bounds.

9.41.14 `viewFrameChanged(notification as NSNotificationMBS)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Handles an `NSViewFrameDidChangeNotification`, passed in the `aNotification` argument, by updating a containing `NSScrollView` based on the new frame.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.41.15 Properties

9.41.16 `backgroundColor` as `NSColorMBS`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The color of the receiver's background.

Notes: (Read and Write computed property)

9.41.17 `copiesOnScroll` as `boolean`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver copies rendered images while scrolling.

Notes: If true, the receiver copies the existing rendered image to its new location while scrolling and only draws exposed portions of its document view. If false, the receiver always forces its document view to draw itself on scrolling.

(Read and Write computed property)

9.41.18 `documentCursor` as `Variant`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The cursor object used over the receiver.

Notes: Value is an `NSCursorMBS` object.

(Read and Write computed property)

9.41.19 documentView as NSViewMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The document view.

Notes: Gets or sets the receiver's document view. On set removes any previous document view, and sets the origin of the receiver's bounds rectangle to the origin of the new view's frame rectangle.

If the receiver is contained in an NSScrollView, you should set the NSScrollView DocumentView instead, so it can perform whatever updating it needs.

In the process of setting the document view, this method registers the receiver for the notifications NSViewFrameDidChangeNotification and NSViewBoundsDidChangeNotification, adjusts the key view loop to include the new document view, and updates a parent NSScrollView's display if needed using reflectScrolledClipView. (Read and Write computed property)

9.41.20 drawsBackground as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver draws its background color.

Notes: If your NSClipView is enclosed in an NSScrollView, you should set the DrawsBackground on the NSScrollView.

(Read and Write computed property)

9.41.21 Constants

Findbar position constants.

Constant	Value	Description
NSClipViewFindBarPositionAboveContent	1	The find bar is displayed above the scroll view content. Available in Mac OS X v10.7 and later.
NSClipViewFindBarPositionAboveHorizontalRuler	0	The find bar is displayed above the horizontal ruler, if visible. Available in Mac OS X v10.7 and later.
NSClipViewFindBarPositionBelowContent	2	The find bar is displayed below the scroll view content. Available in Mac OS X v10.7 and later.

Elasticity behavior constants.

Constant	Value	Description
NSScrollElasticityAllowed	2	Allow content to be scrolled past its bounds on this axis in an elastic fashion. Available in Mac OS X v10.7 and later.
NSScrollElasticityAutomatic	0	Automatically determine whether to allow elasticity on this axis. Available in Mac OS X v10.7 and later.
NSScrollElasticityNone	1	Disallow scrolling beyond document bounds on this axis. Available in Mac OS X v10.7 and later.

9.42 control NSComboBoxControlMBS

9.42.1 control NSComboBoxControlMBS

Plugin Version: 21.4, Platform: macOS, Targets: Desktop only.

Function: The Xojo control for a NSComboBox.

Notes: This control embeds a special NSComboBox subclass.

Designed for Xojo 2013r1 and newer. May work on Xojo 2012, but not perfectly.

Please use view property to access the underlying object and set properties.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.4](#)
- [MBS Xojo Plugins, version 21.4pr2](#)

Xojo Developer Magazine

- [19.6, page 10: News](#)

9.42.2 Properties

9.42.3 View as NSComboBoxMBS

Plugin Version: 21.4, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.

(Read only property)

9.42.4 Events

9.42.5 Action

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The action event run, when e.g. return key is pressed.

9.42.6 BoundsChanged

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.42.7 Close

Plugin Version: 21.4, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.42.8 ConstructContextMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.42.9 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: Called when a menuitem is choosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.42.10 didCloseContextMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextMenu event.

9.42.11 EnableMenuItems

Plugin Version: 21.4, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named MenuBarSelected.

9.42.12 FrameChanged

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.42.13 GotFocus

Plugin Version: 21.4, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named FocusReceived.

Notes:

This only fires if the control itself got focus and not a sub control.

9.42.14 LostFocus

Plugin Version: 21.4, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.42.15MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle theMouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles theMouseDown so the above event handlers do not get called.

9.42.16 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.42.17 MouseUp(x As Integer, y As Integer)

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.42.18 Open

Plugin Version: 21.4, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.

In Xojo version 2021r3 and newer this event is named Opening.

9.42.19 ScaleFactorChanged(NewFactor as double)

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.42.20 `TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: Sent when a control with editable text begins an editing session.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidBeginEditingNotification`.

This event is invoked when the user begins editing text in a control such as a text field or a form field. The control posts a `NSControlTextDidBeginEditingNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is also delivered for inspection.

9.42.21 `TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: Sent when the text in the receiving control changes.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidChangeNotification`.

This event is invoked when text in a control such as a text field or form changes. The control posts a `NSControlTextDidChangeNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is provided as parameter for inspection.

9.42.22 `TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: Sent when a control with editable text ends an editing session.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidEndEditingNotification`.

This event is invoked when the user stops editing text in a control such as a text field or form. The control posts a `NSControlTextDidEndEditingNotification` notification, and if the control's subclass implements this

event, it is automatically registered to receive the notification. The field editor is also provided for inspection.

9.42.23 textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.42.24 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 21.4, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow end of text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.42.25 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.43 class NSComboBoxMBS

9.43.1 class NSComboBoxMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The ComboBox control from Cocoa.

Notes: An NSComboBox is a kind of NSControl that allows you to either enter text directly (as you would with an NSTextField) or click the attached arrow at the right of the combo box and select from a displayed ("pop-up") list of items.

Subclass of the NSTextFieldMBS class.

9.43.2 Methods

9.43.3 addItemWithObjectValue(value as Variant)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Adds an object to the end of the receiver's internal item list.

Notes: anObject: The object to add to the internal item list.

This method logs a warning if usesDataSource returns true.

9.43.4 Constructor

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with size 100/100 and position 0/0

Example:

```
dim t as new NSComboBoxMBS(0, 0, 100, 20)
```

Notes: On success the handle property is not zero.

See also:

- 9.43.5 Constructor(Handle as Integer) 500
- 9.43.6 Constructor(left as Double, top as Double, width as Double, height as Double) 501

9.43.5 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSComboBox handle.

Example:

```
dim t as new NSComboBoxMBS(0, 0, 100, 100)
dim v as new NSComboBoxMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSComboBox and the plugin retains this handle.

See also:

- 9.43.4 Constructor 500
- 9.43.6 Constructor(left as Double, top as Double, width as Double, height as Double) 501

9.43.6 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new combo box with the given size and position.

Example:

```
dim left,top,width,height as Integer
// define rectangle
dim x as new NSComboBoxMBS(left, top, width, height)
```

Notes: On success the handle property is not zero.

See also:

- 9.43.4 Constructor 500
- 9.43.5 Constructor(Handle as Integer) 500

9.43.7 deselectItemAtIndex(index as Integer)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Deselects the pop-up list item at the specified index if it's selected.

Notes: index: The index of the item to deselect.

If the selection does in fact change, this method posts an NSComboBoxSelectionDidChangeNotification to the default notification center.

9.43.8 `indexOfItemWithObjectValue(value as Variant) as Integer`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Searches the receiver's internal item list for the specified object and returns the lowest matching index.

Notes: `anObject`: The object for which to return the index.

Returns the lowest index in the internal item list whose corresponding value is equal to that of the specified object. Objects are considered equal if they have the same id or if `isEqual` returns true.

If none of the objects in the receiver's internal item list are equal to `anObject`, `indexOfItemWithObjectValue` returns `NSNotFound` (-1).

This method logs a warning if `usesDataSource` returns true.

9.43.9 `indexOfSelectedItem as Integer`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the last item selected from the pop-up list.

Notes: Returns the index of the last item selected from the receiver's pop-up list or -1 if no item is selected.

Note that nothing is initially selected in a newly initialized combo box.

9.43.10 `noteNumberOfItemsChanged`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Informs the receiver that the number of items in its data source has changed.

Notes: This method allows the receiver to update the scrollers in its displayed pop-up list without actually reloading data into the receiver. It is particularly useful for a data source that continually receives data in the background over a period of time, in which case the `NSComboBox` can remain responsive to the user while the data is received.

See the `NSComboBoxDataSource` informal protocol specification for information on the messages an `NSComboBox` sends to its data source.

9.43.11 numberOfItems as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the total number of items in the pop-up list.

9.43.12 reloadData

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Marks the receiver as needing redisplay, so that it will reload the data for visible pop-up items and draw the new values.

9.43.13 removeAllItems

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Removes all items from the receiver's internal item list.

Notes: This method logs a warning if usesDataSource returns true.

9.43.14 removeItemAtIndex(index as Integer)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Removes the object at the specified location from the receiver's internal item list.

Notes: index: The index of the object to remove. All items beyond index are moved up one slot to fill the gap.

The removed object receives a release message. This method raises an NSRangeException if index is beyond the end of the list and logs a warning if usesDataSource returns true.

9.43.15 removeItemWithObjectValue(value as Variant)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Removes all occurrences of the given object from the receiver's internal item list.

Notes: anObject: The object to remove from the internal item list. Objects are considered equal if they have the same id or if isEqual: returns true.

This method logs a warning if `usesDataSource` returns true.

9.43.16 `scrollItemAtIndexToTop(index as Integer)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Scrolls the receiver’s pop-up list vertically so that the item at the specified index is as close to the top as possible.

Notes: `index`: The index of the item to scroll to the top.

The pop-up list need not be displayed at the time this method is invoked.

9.43.17 `scrollItemAtIndexToVisible(index as Integer)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Scrolls the receiver’s pop-up list vertically so that the item at the specified index is visible.

Notes: `index`: The index of the item to make visible.

The pop-up list need not be displayed at the time this method is invoked.

9.43.18 `selectItemAtIndex(index as Integer)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Selects the pop-up list row at the given index.

Notes: `index`: The index of the item to select in the pop-up list.

Posts an `NSComboBoxSelectionDidChangeNotification` to the default notification center if the selection does in fact change. Note that this method does not alter the contents of the combo box’s text field—see [Setting the Combo Box’s Value](#) for more information.

9.43.19 `selectItemWithObjectValue(value as Variant)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Selects the first pop-up list item that corresponds to the given object.

Notes: `anObject`: The object to select in the pop-up list. Objects are considered equal if they have the same id or if `isEqual`: returns true.

This method logs a warning if `usesDataSource` returns true. Posts an `NSComboBoxSelectionDidChangeNotification` to the default notification center if the selection does in fact change. Note that this method doesn't alter the contents of the combo box's text field—see [Setting the Combo Box's Value](#) for more information.

9.43.20 Properties

9.43.21 `completes as boolean`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value indicating whether the receiver tries to complete what the user types in the text field.

Notes: Returns true if the receiver tries to complete what the user types in the text field; otherwise false. (Read and Write computed property)

9.43.22 `hasVerticalScroller as boolean`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value indicating whether the receiver will display a vertical scroller.

Notes: Returns true if the receiver will display a vertical scroller; otherwise false.

Note that the scroller will be displayed even if the pop-up list contains fewer items than will fit in the area specified for display.

(Read and Write computed property)

9.43.23 `intercellSpacing as NSSizeMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the horizontal and vertical spacing between cells in the receiver's pop-up list.

Notes: Returns the space between cells in the pop-up list. The default spacing is (3.0, 2.0).

(Read and Write computed property)

9.43.24 `isButtonBordered as boolean`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Whether the combo box button is set to display a border.

Notes: True if the button has a border; otherwise false.

Available in Mac OS X v10.3 and later
(Read and Write computed property)

9.43.25 `itemHeight` as `Double`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the height of each item in the receiver's pop-up list.

Notes: The default item height is 16.0.

(Read and Write computed property)

9.43.26 `numberOfVisibleItems` as `Integer`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the maximum number of items visible in the pop-up list.

Notes: The maximum number of items visible at any one time in the pop-up list.

(Read and Write computed property)

9.43.27 `usesDataSource` as `boolean`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the receiver uses an external data source to populate its pop-up list.

Notes: True if the receiver uses an external data source to populate the receiver's pop-up list, false if it uses an internal item list.

(Read and Write computed property)

9.44 control NSDatePickerControlMBS

9.44.1 control NSDatePickerControlMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The control to put a Mac OS X date picker on a Xojo window.

Notes: This control embeds a special NSDatePicker subclass.

Designed for Xojo 2013r1 and newer. May work on Xojo 2012, but not perfectly.

Please use view property to access the underlying object and set properties.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo Plugins, version 19.2pr1](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr7](#)

9.44.2 Properties

9.44.3 AcceptTabs as Boolean

Plugin Version: 19.2, Platform: macOS, Targets: Desktop only.

Function: Whether the control should accept tab keys.

Notes: If true, the plugin will not forward the tab keydown/keyup events to Xojo, because Xojo would do switch to next control.

(Read and Write property)

9.44.4 View as NSDatePickerMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The reference to the underlying NSDatePicker.

Notes: (Read only property)

9.44.5 Events

9.44.6 Action

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: This event is called when user clicks on a date/time and changes something.

9.44.7 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.44.8 Close

Plugin Version: 15.0, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.44.9 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.44.10 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is choosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.44.11 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.44.12 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named MenuBarSelected.

9.44.13 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.44.14 GotFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named FocusReceived.

Notes:

This only fires if the control itself got focus and not a sub control.

9.44.15 LostFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.44.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.44.17 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.44.18 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.44.19 Open

Plugin Version: 15.0, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.

In Xojo version 2021r3 and newer this event is named Opening.

9.44.20 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.44.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.45 class NSDatePickerMBS

9.45.1 class NSDatePickerMBS

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The plugin class for a Cocoa date picker.

Notes: NSDatePicker is a subclass of NSControl that provides a user interface for displaying and editing an NSDate object.

Subclass of the NSControlMBS class.

Blog Entries

- [MBS Xojo Plugins, version 20.5pr7](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr1](#)
- [MBS Real Studio Plugins, version 12.2pr4](#)

9.45.2 Methods

9.45.3 Constructor

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new date picker with size 100/100 and position 0/0

Example:

```
dim t as new NSDatePickerMBS
```

Notes: On success the handle property is not zero.

See also:

- [9.45.4 Constructor\(Handle as Integer\)](#) 512
- [9.45.5 Constructor\(left as Double, top as Double, width as Double, height as Double\)](#) 513

9.45.4 Constructor(Handle as Integer)

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSDatePicker handle.

Example:

```
dim t as new NSDatePickerMBS(0, 0, 100, 100)
dim v as new NSDatePickerMBS(t.handle)
```

9.45. CLASS NSDATEPICKERMBS

513

MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)

Notes: The handle is casted to a NSDatePicker and the plugin retains this handle.
See also:

- 9.45.3 Constructor 512
- 9.45.5 Constructor(left as Double, top as Double, width as Double, height as Double) 513

9.45.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new date picker with the given size and position.

Example:

```
dim x as new NSDatePickerMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.
See also:

- 9.45.3 Constructor 512
- 9.45.4 Constructor(Handle as Integer) 512

9.45.6 Properties

9.45.7 backgroundColor as NSColorMBS

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The background color.

Notes: See also drawsBackground property.
(Read and Write property)

9.45.8 Bezeled as Boolean

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver has a bezeled border.

Notes: True if the receiver has a bezeled border, false otherwise.

(Read and Write property)

9.45.9 Bordered as Boolean

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver has a plain border.

Notes: True if the receiver has a plain border, false otherwise.
(Read and Write property)

9.45.10 calendar as NSCalendarMBS

Plugin Version: 13.4, Platform: macOS, Targets: Desktop only.

Function: The calendar.

Notes: (Read and Write property)

9.45.11 datePickerElements as Integer

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: A bitmask that indicates which visual elements of the date picker are currently shown, and which won't be usable because they are hidden.

Notes: (Read and Write property)

9.45.12 datePickerMode as Integer

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The date picker mode.

Notes: (Read and Write property)

9.45.13 datePickerStyle as Integer

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The date picker style.

Notes: (Read and Write property)

9.45.14 `dateTimeValue` as `dateTime`

Plugin Version: 20.5, Platform: macOS, Targets: Desktop only.

Function: The current date value.

Notes: (Read and Write property)

9.45.15 `dateValue` as `date`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The current date value.

Notes: For a range of dates, this is start date and you use `timeInterval` to get the second one.

(Read and Write property)

9.45.16 `drawsBackground` as `Boolean`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver draws the background.

Notes: True if the receiver draws the background, false otherwise.

(Read and Write property)

9.45.17 `locale` as `NSLocaleMBS`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The locale.

Notes: (Read and Write property)

9.45.18 `maxDate` as `date`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The maximum date.

Notes: nil indicates no maximum date.

(Read and Write property)

9.45.19 `maxDateTime` as `dateTime`

Plugin Version: 20.5, Platform: macOS, Targets: Desktop only.

Function: The maximum date.

Notes: nil indicates no maximum date.

(Read and Write property)

9.45.20 `minDate` as `date`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The minimum date.

Notes: nil indicates no minimum date.

(Read and Write property)

9.45.21 `minDateTime` as `dateTime`

Plugin Version: 20.5, Platform: macOS, Targets: Desktop only.

Function: The minimum date.

Notes: nil indicates no minimum date.

(Read and Write property)

9.45.22 `textColor` as `NSColorMBS`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The text color.

Notes: (Read and Write property)

9.45.23 `timeInterval` as `Double`

Plugin Version: 12.2, Platform: macOS, Targets: Desktop only.

Function: The time interval that represents the date range.

Notes: The time interval that represents the receiver's date range. The date range begins at the date returned by `dateValue`. This method returns 0 when the receiver is not in the `NSRangeDateMode` mode.

(Read and Write property)

9.45.24 timeZone as NSTimeZoneMBS

Plugin Version: 13.4, Platform: macOS, Targets: Desktop only.

Function: The time zone.

Notes: (Read and Write property)

9.45.25 Constants

Date Picker Style Constants

Constant	Value	Description
NSClockAndCalendarDatePickerStyle	1	Provide a visual clock and calendar style interface.
NSTextFieldAndStepperDatePickerStyle	0	Provide a text field and stepper style interface.
NSTextFieldDatePickerStyle	2	Provide a text field interface.

Element Flag Constants

Constant	Value	Description
NSEraDatePickerElementFlag	&h100	Display and allow editing of the era of the date, if applicable. This flag has been declared for possible future use, and does not have any effect.
NSHourMinuteDatePickerElementFlag	&hC	Display and allow editing of the hour and minute elements of the date.
NSHourMinuteSecondDatePickerElementFlag	&hE	Display and allow editing of the hour, minute and second elements of the date.
NSTimeZoneDatePickerElementFlag	&h10	Display and allow editing of the time zone. This flag has been declared for possible future use, and does not have any effect.
NSYearMonthDatePickerElementFlag	&hC0	Display and allow editing of the year and month elements of the date.
NSYearMonthDayDatePickerElementFlag	&hE0	Display and allow editing of the year, month and day elements of the date.

Date Picker Mode Constants

Constant	Value	Description
NSRangeDateMode	1	Allow selection of a range of dates. (First implemented in Mac OS X v 10.5.)
NSSingleDateMode	0	Allow selection of a single date.

9.46 class `NSImageViewMBS`

9.46.1 class `NSImageViewMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: An `NSImageView` object displays a single image from an `NSImage` object in a frame and can optionally allow a user to drag an image to it.

Example:

```
ImageWell1.NSImageViewMBS.alphaValue = 0.5
```

Notes: You can embed this view in a `CustomNSViewMBS` to get more events for mouse and keyboard. Subclass of the `NSControlMBS` class.

Blog Entries

- [News from the MBS Xojo Plugins Version 22.1](#)
- [Several ways for picture to PDF in MBS Plugins](#)
- [MBS Xojo Plugins, version 21.2pr1](#)
- [Fun with ImageWell and NSImageViewMBS](#)

9.46.2 Methods

9.46.3 Constructor

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new imageview with size 100/100 and position 0/0

Example:

```
dim t as new NSImageViewMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.46.4 Constructor(Handle as Integer) 518
- 9.46.5 Constructor(left as Double, top as Double, width as Double, height as Double) 519

9.46.4 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSImageView handle.

Example:

```
dim t as new NSImageViewMBS(0, 0, 100, 100)
dim v as new NSImageViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSImageView and the plugin retains this handle.

See also:

- 9.46.3 Constructor 518
- 9.46.5 Constructor(left as Double, top as Double, width as Double, height as Double) 519

9.46.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new image view with the given size and position.

Example:

```
dim x as new NSImageViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.46.3 Constructor 518
- 9.46.4 Constructor(Handle as Integer) 518

9.46.6 Properties

9.46.7 allowsCutCopyPaste as Boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver allows the user to cut, copy and paste the image contents.

Notes: True if the user can cut, copy, and paste the image contents; otherwise, false to prevent the use of pasteboard operations.

(Read and Write computed property)

9.46.8 animates as Boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the receiver automatically plays animated images.

Notes: True if the receiver automatically plays animated images; otherwise, false. The default value is true for UIImageView objects you create programmatically. For UIImageView objects loaded from a nib file, the control takes the value set in Interface Builder.

The timing and looping characteristics of the animation are taken from the image data. If this method returns false, the receiver displays the first frame of the animation only.

(Read and Write computed property)

9.46.9 image as NSImageMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the NSImage object displayed by the receiver.

Notes: (Read and Write computed property)

9.46.10 imageAlignment as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The alignment of the receiver's image relative to its frame.

Notes: For a list of possible values, see NSImageAlign* constants. The default value is NSImageAlignCenter.

(Read and Write computed property)

9.46.11 imageFrameStyle as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The style of the frame that borders the image.

Example:

```
ImageWell1.NSImageViewMBS.imageFrameStyle = NSImageViewMBS.NSImageFramePhoto
```

Notes: Value is one of the frame style constants. For a list of frame styles, see NSImageFrame* constants.

(Read and Write computed property)

9.46.12 imageScaling as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The scaling mode used to fit the receiver's image into the frame.

Example:

```
ImageWell1.NSImageViewMBS.imageScaling = NSImageViewMBS.NSScaleNone
```

Notes: Value is one of the image scaling constants. For a list of possible values, see NSScale* constants. (Read and Write computed property)

9.46.13 isEditable as Boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Whether the user can drag a new image into the frame.

Notes: True if the user can drag an image into the receiver's frame; otherwise, false. (Read and Write computed property)

9.46.14 Constants

Constants

Constant	Value	Description
<code>NSImageAlignBottom</code>	5	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Align the image with the bottom edge of the cell.
<code>NSImageAlignBottomLeft</code>	6	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Align the image with the bottom and left edges of the cell.
<code>NSImageAlignBottomRight</code>	7	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Align the image with the bottom and right edges of the cell.
<code>NSImageAlignCenter</code>	0	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Center the image in the cell.
<code>NSImageAlignLeft</code>	4	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Align the image with the left edge of the cell.
<code>NSImageAlignRight</code>	8	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Position the image along the right edge of the cell.
<code>NSImageAlignTop</code>	1	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Position the image along the top edge of the cell.
<code>NSImageAlignTopLeft</code>	2	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Align the image with the top and left edges of the cell.
<code>NSImageAlignTopRight</code>	3	One of the constants to specify the location of the image in the frame using the <code>imageAlignment</code> property. Align the image with the top and right edges of the cell.
<code>NSImageFrameButton</code>	4	One of the frame constants for the <code>imageFrameStyle</code> property. A convex bezel that makes the image stand out in relief, like a button
<code>NSImageFrameGrayBezel</code>	2	One of the frame constants for the <code>imageFrameStyle</code> property. A gray, concave bezel that makes the image look sunken
<code>NSImageFrameGroove</code>	3	One of the frame constants for the <code>imageFrameStyle</code> property. A thin groove that looks etched around the image
<code>NSImageFrameNone</code>	0	One of the frame constants for the <code>imageFrameStyle</code> property. An invisible frame
<code>NSImageFramePhoto</code>	1	One of the frame constants for the <code>imageFrameStyle</code> property. A thin black outline and a dropped shadow

Scale Methods

Constant	Value	Description
NSImageScaleAxesIndependently	1	Scale each dimension to exactly fit destination. This setting does not preserve the aspect ratio of the image.
NSImageScaleNone	2	Do not scale the image.
NSImageScaleProportionallyDown	0	If it is too large for the destination, scale the image down while preserving aspect ratio.
NSImageScaleProportionallyUpOrDown	3	Scale the image to its maximum possible dimensions while both staying within the destination area and preserving its aspect ratio.
NSScaleNone	2	Do not scale the image.
NSScaleProportionally	0	If it is too large for the destination, scale the image down while preserving aspect ratio.
NSScaleToFit	1	Scale each dimension to exactly fit destination. This setting does not preserve the aspect ratio of the image.

9.47 control NSOutlineControlMBS

9.47.1 control NSOutlineControlMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The control for a NSOutlineView.

Notes: Please use NSOutlineControlMBS for hierarchical lists and NSTableControlMBS for normal lists.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.1](#)
- [MBS Xojo Plugins, version 17.1pr4](#)

Videos

- [Presentation from London conference about MBS Plugins.](#)

Xojo Developer Magazine

- [15.3, page 10: News](#)

9.47.2 Properties

9.47.3 AcceptTabs as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Whether the control should accept tab keys.

Notes: If true, the plugin will not forward the tab keydown/keyup events to Xojo, because Xojo would do switch to next control.

(Read and Write property)

9.47.4 allowsColumnReordering as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to rearrange columns by dragging their headers.

Notes: The default value of this property is true, which allows the user to rearrange the table view,Äôs columns. You can rearrange columns programmatically regardless of this setting.

(Read and Write property)

9.47.5 allowsColumnResizing as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to resize columns by dragging between their headers.

Notes: The default of this property is true, which allows the user to resize the table view's columns. You can resize columns programmatically regardless of this setting.
(Read and Write property)

9.47.6 allowsColumnSelection as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select columns by clicking their headers.

Notes: The default is false, which prevents the user from selecting columns (if you create the table view in Interface Builder, the default value is true). You can select columns programmatically regardless of this setting.
(Read and Write property)

9.47.7 allowsEmptySelection as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select zero columns or rows.

Notes: The default is true, which allows the user to select zero columns or rows.
(Read and Write property)

9.47.8 allowsMultipleSelection as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select more than one column or row at a time.

Notes: The default is false, which allows the user to select only one column or row at a time. You can select multiple columns or rows programmatically regardless of this setting.
(Read and Write property)

9.47.9 autohidesScrollers as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view automatically hides its scroll bars when they are not needed.

Notes: The horizontal and vertical scroll bars are hidden independently of each other. When the value of this property is YES and the content of the scroll view doesn't extend beyond the size of the clip view on a given axis, the scroller on that axis is removed to leave more room for the content.

(Read and Write property)

9.47.10 hasHorizontalScroller as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a horizontal scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a horizontal scroller as needed. The default value of this property is false.

(Read and Write property)

9.47.11 hasVerticalScroller as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a vertical scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a vertical scroller as needed. The default value of this property is false.

(Read and Write property)

9.47.12 ScrollView as NSScrollViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The scroll view used in this control.

Notes: (Read only property)

9.47.13 View as NSOutlineViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The outline view used in this control.

Notes: (Read only property)

See also:

- 9.47.94 view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS
550

9.47.14 Events

9.47.15 acceptDrop(info as NSDraggingInfoMBS, item as NSOutlineViewItemMBS, index as Integer) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a drop operation was successful.

Notes: info: An object that contains more information about this dragging operation.

item: The parent of the item over which the cursor was placed when the mouse button was released.

index: The index of the child of item over which the cursor was placed when the mouse button was released.

Return true if the drop operation was successful, otherwise false.

The data source should incorporate the data from the dragging pasteboard in the implementation of this method. You can get the data for the drop operation from info using the draggingPasteboard method. The return value indicates success or failure of the drag operation to the system.

9.47.16 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.47.17 childOfItem(index as Integer, item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns the child item at the specified index of a given item.

Notes: index: The index of the child item from item to return.

item: An item in the data source.

Return the child item at index of item. If item is nil, returns the appropriate child item of the root object.

Children of a given parent item are accessed sequentially. In order for the collapsed state of the outline view to remain consistent when it is reloaded you must always return the same object for a specified child and item.

Do not call reloadData from this method.

This event is called very frequently, so it must be efficient.

9.47.18 Close

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The control is about to close.
In Xojo version 2021r3 and newer this event is named Closing.

9.47.19 ColumnDidMove(notification as NSNotificationMBS, OldColumn as Integer, NewColumn as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked whenever the user moves a column in the outline view.

9.47.20 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked whenever the user resizes a column in the outline view.

9.47.21 concludeDragOperation(info as NSDraggingInfoMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when the dragging operation is complete, signaling the receiver to perform any necessary clean-up.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

For this method to be invoked, the previous performDragOperation must have returned true.

The destination implements this method to perform any tidying up that it needs to do, such as updating its visual representation now that it has incorporated the dragged data. This message is the last message sent from sender to the destination during a dragging session.

If the sender object's `animatesToDestination` property was set to true in `prepareForDragOperation`, then the drag image is still visible. At this point you should draw the final visual representation in the view. When this method returns, the drag image is removed from the screen. If your final visual representation matches the visual representation in the drag, this is a seamless transition.

9.47.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.47.23 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.47.24 dataCell(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSCellMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns the cell to use in a given column for a given item.

Notes: `tableColumn`: The table column for which the cell is required. This value may be nil.
`item`: The item for which the cell is required.

Return the cell to use in column `tableColumn` for item `item`, or nil. The cell must properly implement `copyWithZone` (since it may be copied by the outline view).

You can return a different data cell for any particular combination of table column and item, or a cell that will be used for the entire row (a full-width cell). If `tableColumn` is non-nil, you should return a cell. Typically, you should default to returning the result from [`tableColumn dataCellForRow:row`] .

When each row (identified by the item) is being drawn, this method is first called with a nil value for `tableColumn`. At this time, you can return a cell that is used to draw the entire row, acting like a group. If

you do return a cell for the nil table column, your implementations of the other corresponding data source and delegate methods must be prepared to be invoked with a nil value for tableColumn. If do not return a cell for the nil table column, the method is called once for each column in the outline view, as usual.

9.47.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implemented to know when a new row view is added to the table.

Notes: rowView: The new row view.

row: The row to which the view was added.

This event is for NSView-based outline views. At this point, you can choose to add in extra views or modify any properties on rowView.

9.47.26 didClickTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Sent at the time the mouse button subsequently goes up in outlineView and tableColumn has been ,Ä¼clicked,Ä½ without having been dragged anywhere.

9.47.27 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.47.28 didDragTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Sent at the time the mouse button goes up in outlineView and tableColumn has been dragged during the time the mouse button was down.

9.47.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implemented to know when a row view is removed from the table

Notes: rowView: The row view that was removed.

row: The number of the row that was removed due to being moved offscreen, or -1 if the row was removed from the table so it is no longer valid.

The removed rowView may be reused by the table, so any additionally inserted views should be removed at this point.

9.47.30 didTile

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The tableview did tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.47.31 DoubleClick

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse made a double click.

9.47.32 draggingEnded(info as NSDraggingInfoMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this event to be notified when a drag operation ends in some other destination.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

This method might be used by a destination doing auto-expansion in order to collapse any auto-expands.

9.47.33 draggingExited(info as NSDraggingInfoMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when the dragged image enters destination bounds or frame; delegate returns dragging operation to perform.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in `NSDragOperation` in the `NSDraggingInfo` reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous `draggingEntered` message.

Invoked when a dragged image enters the destination but only if the destination has registered for the pasteboard data type involved in the drag operation. Specifically, this method is invoked when the mouse pointer enters the destination's bounds rectangle (if it is a view object) or its frame rectangle (if it is a window object).

This method must return a value that indicates which dragging operation the destination will perform when the image is released. In deciding which dragging operation to return, the method should evaluate the overlap between both the dragging operations allowed by the source (obtained from sender with the `draggingSourceOperationMask` method) and the dragging operations and pasteboard data types the destination itself supports.

If none of the operations is appropriate, this method should return `NSDragOperationNone` (this is the default response if the method is not implemented by the destination). A destination will still receive `draggingUpdated` and `draggingExited` even if `NSDragOperationNone` is returned by this method.

9.47.34 `draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to know when the given dragging session has ended.

Notes: session: The dragging session that ended.

screenPoint: The point onscreen at which the drag ended.

operation: A mask specifying the types of drag operations permitted by the dragging source.

You can implement this optional delegate method to know when the dragging source operation ended at a specific location, such as the trash (by checking for an operation of `NSDragOperationDelete`).

9.47.35 `draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, draggedItems() as NSOutlineViewItemMBS)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method know when the given dragging session is about to begin and potentially modify the dragging session.

Notes: session: The dragging session that is about to begin.

screenPoint: The point onscreen at which the drag is to begin.

draggedItems: A array of items to be dragged, excluding items for which pasteboardWriterForItem returns nil.

The draggedItems array directly matches the pasteboard writer array used to begin the dragging session with the NSView method beginDraggingSessionWithItems. Hence, the order is deterministic, and can be used in acceptDrop when enumerating the NSDraggingInfo protocol's pasteboard classes.

9.47.36 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named MenuBarSelected.

9.47.37 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.47.38 GotFocus

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named FocusReceived.

Notes:

This only fires if the control itself got focus and not a sub control.

9.47.39 heightOfRowByItem(item as NSOutlineViewItemMBS) as Double

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns the height in points of the row containing item.

Notes: item: The row item.

Return the height of the row.

Values returned by this method should not include intercell spacing and must be greater than 0. Implement this event to support an outline view with varying row heights.

For large tables in particular, you should make sure that this method is efficient. `NSOutlineView` may cache the values this method returns, so if you would like to change a row's height make sure to invalidate the row height by calling `noteHeightOfRowsWithIndexesChanged`. `NSOutlineView` automatically invalidates its entire row height cache in `reloadData` and `noteNumberOfRowsChanged`.

If you call `viewAtColumn` or `rowViewAtRow` within your implementation of this method, an exception is thrown.

To avoid the possibility of a hang due to unexpected recursion, don't call geometry-calculating methods such as `bounds`, `rectOfColumn`, or any `NSTableView` method that calls `tile` within your implementation of this method.

9.47.40 `isGroupItem(item as NSOutlineViewItemMBS)` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean that indicates whether a given row should be drawn in the „group row“ style.

Notes: item: An item in the outline view.

Return true to indicate a particular row should have the "group row" style drawn for that row, otherwise false.

If the cell in that row is an instance of `NSTextFieldCell` and contains only a string value, the „group row“ style attributes are automatically applied for that cell.

9.47.41 `isItemExpandable(item as NSOutlineViewItemMBS)` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the a given item is expandable.

Notes: item: An item in the data source.

Returns true if item can be expanded to display its children, otherwise NO.

This method may be called quite often, so it must be efficient.

Do not call reloadData from this method.

9.47.42 ItemDidCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when the did collapse notification is posted—that is, whenever the user collapses an item in the outline view.

9.47.43 ItemDidExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the user expands an item in the outline view.

9.47.44 itemForPersistentObject(PersistentObject as Variant) as NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by outlineView to return the item for the archived object.

Notes: object: An archived representation of an item in outlineView’s data source.

Return the unarchived item corresponding to object. If the item is an archived object, this method may return the object.

When the outline view is restoring the saved expanded items, this method is called for each expanded item, to translate the archived object to an outline view item.

You must implement this method if you are automatically saving expanded items (that is, if autosaveExpandedItems returns true).

9.47.45 ItemWillCollapse(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the user is about to collapse an item in the outline view.

9.47.46 ItemWillExpand(notification as NSNotificationMBS, item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the user is about to expand an item in the outline view.

9.47.47 LeftMouseDown(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.48 LeftMouseDragged(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the left button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.49 LeftMouseUp(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.50 LostFocus

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.47.51MouseDown(x as Integer, y as Integer, Modifiers as Integer) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.47.52 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Event sent whenever the mouse button is clicked in outlineView while the cursor is in a column header tableColumn.

9.47.53 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.47.54 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.47.55 namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedItems() as NSOutlineViewItemMBS) as string()

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns an array of filenames for the created files that the receiver promises to create.

Notes: dropDestination: The drop location where the files are created.

items: The items being dragged.

Returns an array of filenames (not full paths) for the created files that the receiver promises to create.

For more information on file promise dragging, see documentation on the NSDraggingSource protocol and namesOfPromisedFilesDroppedAtDestination.

9.47.56 nextTypeSelectMatchFromItem(startItem as NSOutlineViewItemMBS, endItem as NSOutlineViewItemMBS, searchString as String) as NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns the first item that matches the searchString from within the range of startItem to endItem.

Notes: startItem: The first item to search.

endItem: The item before which to stop searching. It is possible for endItem to be less than startItem if the search will wrap.

searchString: The string for which to search.

Returns the first item—from within the range of startItem to endItem—that matches the searchString, or nil if there is no match.

Implement this method if you want to control how type selection works. You should include startItem as a possible match, but do not include endItem.

It is not necessary to implement this event in order to support type select.

9.47.57 numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns the number of child items encompassed by a given item.

Notes: item: An item in the data source.

Returns the number of child items encompassed by item. If item is nil, this method should return the number of children for the top-level item.

The numberOfChildrenOfItem method is called very frequently, so it must be efficient.

Do not call reloadData from this method.

9.47.58 objectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by outlineView to return the data object associated with the specified item.

Notes: tableColumn: A column in outlineView.

item: An item in the data source in the specified tableColumn of the view.

Returns the item is located in the specified tableColumn of the view.

Do not call reloadData from this method.

9.47.59 Open

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The control is about to was created and you can initialize it.

In Xojo version 2021r3 and newer this event is named Opening.

9.47.60 OtherMouseDown(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed a mouse button other than the left or right one.
Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.61 OtherMouseDragged(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with a button other than the left or right button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.62 OtherMouseUp(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released a mouse button other than the left or right button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.63 pasteboardWriterForItem(item as NSOutlineViewItemMBS) as NSPasteboardItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to enable the table to be an NSDraggingSource that supports dragging multiple items.

Notes: item: The item for which to return a pasteboard writer.

Returns a NSPasteboardItem object.

If this method is implemented, then writeItems is not called.

9.47.64 persistentObjectForItem(item as NSOutlineViewItemMBS) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by outlineView to return an archived object for item.

Notes: item: The item for which to return an archived object.

Returns an archived representation of item. If the item is an archived object, this method may return the item.

When the outline view is saving the expanded items, this method is called for each expanded item, to translate the outline view item to an archived object.

You must implement this method if you are automatically saving expanded items (that is, if `autosaveExpandedItems` returns true).

9.47.65 RightMouseDown(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the view that the user has pressed the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.66 RightMouseDragged(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the right button pressed .

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.67 RightMouseUp(e as NSEventMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.47.68 rowViewForItem(item as NSOutlineViewItemMBS) as NSTableViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to return a custom `NSTableView` for a particular item.

Notes: item: The item displayed by the returned table row view.

Return an instance or subclass of `NSTableView`. If `nil` is returned, a `NSTableView` instance is created and used.

This method, if implemented, is only invoked for `NSView`-based outline views.

9.47.69 `ScaleFactorChanged(NewFactor as Double)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.47.70 `SelectionDidChange(notification as NSNotificationMBS)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when the selection did change notification is posted—that is, immediately after the outline view, “selection has changed.

9.47.71 `selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to modify the proposed selection.

Notes: `proposedSelectionIndexes`: An index set containing the indexes of the proposed selection.

Return an `NSIndexSet` instance containing the indexes of the new selection. Return `proposedSelectionIndexes` if the proposed selection is acceptable, or the value of the table view, “existing selection to avoid changing the selection.

This method may be called multiple times with one new index added to the existing selection to find out if a particular index can be selected when the user is extending the selection with the keyboard or mouse.

Implementation of this method is optional. If implemented, this method will be called instead of `willDisplayOutlineCell`.

If not implemented or returns `nil`, the plugin will return `proposedSelectionIndexes`.

9.47.72 SelectionIsChanging(notification as NSNotificationMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked when notification is posted—that is, whenever the outline view,Äôs selection changes.

9.47.73 selectionShouldChangeInOutlineView as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should change its selection.

Notes: Return true to permit outlineView to change its selection (typically a row being edited), false to deny permission.

For example, if the user is editing a cell and enters an improper value, the delegate can prevent the user from selecting or editing any other cells until a proper value has been entered into the original cell. You can implement this method for complex validation of edited rows based on the values of any of their cells.

9.47.74 setObjectValue(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, value as Variant)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Set the data object for a given item in a given column.

Notes: object: The new value for the item.

tableColumn: A column in outlineView.

item: An item in the data source in the specified tableColumn of the view.

The item is located in the specified tableColumn of the view.

Do not call reloadData from this method.

9.47.75 shouldCollapseAutoExpandedItemsForDeposited(deposited as Boolean, superResult as Boolean) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether auto-expanded items should return to their original collapsed state.

Notes: deposited: If true, the drop terminated successfully; if false the drop failed.

Return true if auto-expanded items should return to their original collapsed state; otherwise false.

Implement this event to provide custom behavior. If the target of a drop is not auto-expanded (by hovering long enough) the drop target still gets expanded after a successful drop unless this method returns true. The default implementation returns false after a successful drop.

This method is called in a variety of situations. For example, it is called shortly after the `acceptDrop` method is called and also if the drag exits the outline view (exiting the view is treated the same as a failed drop). The return value of the `acceptDrop` method determines the incoming value of the deposited parameter.

9.47.76 `shouldCollapseItem(item as NSOutlineViewItemMBS)` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should collapse a given item.

Notes: `item`: The item that should collapse.

Return true to permit `outlineView` to collapse `item`, false to deny permission.

You can implement this method to disallow collapsing of specific items. For example, if the first row of your outline view should not be collapsed, your delegate method could contain this line of code:

```
return rowForItem(item) <> 0
```

9.47.77 `shouldEdit(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should allow editing of a given item in a given table column.

Notes: `tableColumn`: The table column.

`item`: The item.

Returns true to permit `outlineView` to edit the cell specified by `tableColumn` and `item`, false to deny permission.

If this method returns true, the cell may still not be editable—for example, if you have set up a custom `NSTextFieldCell` as a data cell, it must return true for `isEditable` to allow editing.

You can implement this method to disallow editing of specific cells.

9.47.78 shouldExpandItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should expand a given item.

Notes: item: The item that should expand.

Returns true to permit outlineView to expand item, false to deny permission.
You can implement this method to disallow expanding of specific items.

9.47.79 shouldReorderColumn(columnIndex as Integer, newColumnIndex as Integer) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Sent to the delegate to allow or prohibit the specified column to be dragged to a new location.

Notes: columnIndex: The index of the column being dragged.

newColumnIndex: The proposed target index of the column.

Returns true if the column reordering should be allowed, otherwise false.

When a column is initially dragged by the user, the delegate is first called with a newColumnIndex value of -1. Returning false will disallow that column from being reordered at all. Returning true allows it to be reordered, and the delegate will be called again when the column reaches a new location.

The actual NSTableColumn instance can be retrieved from the tableColumns array.
If this method is not implemented, all columns are considered reorderable.

9.47.80 shouldSelectItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should select a given item.

Notes: item: The item.

Return true to permit outlineView to select item, false to deny permission.

You implement this event to disallow selection of particular items.
For better performance and finer grain control over the selection, use dataCell.

9.47.81 shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the outline view should select a given table column.

Notes: tableColumn: The table column.

Return true to permit outlineView to select tableColumn, false to deny permission.

You can implement this method to disallow selection of specific columns.

9.47.82 shouldShowCellExpansion(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to control cell expansion for a specific column and item.

Notes: tableColumn: A table column in the outline view.

item: An item in the outline view.

Returns true to allow an expansion tooltip to appear in the column tableColumn for item item, otherwise false.

Cell expansion can occur when the mouse hovers over the specified cell and the cell contents are unable to be fully displayed within the cell. If this method returns true, the full cell contents will be shown in a special floating tool tip view, otherwise the content is truncated.

9.47.83 shouldShowOutlineCellForItem(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns whether the specified item should display the outline cell (the disclosure triangle).

Notes: item: An item in the outline view.

Returns true if the outline cell should be displayed, otherwise false.

Returning false causes frameOfOutlineCellAtRow to return NSZeroRect, hiding the cell. In addition, the row will not be collapsible by keyboard shortcuts.

This method is called only for expandable rows.

9.47.84 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a given cell should be tracked.

Notes: cell: The cell used to display item item in column tableColumn

tableColumn: A table column in the outline view.

item: An item in the outline view.

Returns true if the cell should be tracked for the item item in column tableColumn, otherwise false.

Normally, only selectable or selected cells can be tracked. If you implement this method, cells which are not selectable or selected can be tracked (and vice-versa). For example, this allows you to have a button cell in a table which does not change the selection, but can still be clicked on and tracked.

9.47.85 shouldTypeSelectForEvent(e as NSEventMBS, searchString as String) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether type select should proceed for a given event and search string.

Notes: e: The event that caused the message to be sent.

searchString: The string for which searching is to proceed. The search string is nil if no type select has begun.

Return true if type select should proceed, otherwise false.

Generally, this method will be called from keyDown and the event will be a key event.

9.47.86 sizeToFitWidthOfColumn(Column as Integer) as Double

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to provide custom sizing behavior when a column's resize divider is double clicked.

Notes: column: The index of the column.

Returns the width of the specified column.

By default, NSOutlineView iterates every row in the table, accesses a cell via preparedCellAtColumn, and

requests the `cellSize` to find the appropriate largest width to use.

For accurate results and performance, it is recommended that this method is implemented when using large tables. By default, large tables use a monte carlo simulation instead of iterating every row.

9.47.87 `sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by an outline view to notify the data source that the descriptors changed and the data may need to be resorted.

Notes: `oldDescriptors`: An array that contains the previous descriptors.

The data source typically sorts and reloads the data, and adjusts the selections accordingly. If you need to know the current sort descriptors and the data source does not itself manage them, you can get `outlineView`'s current sort descriptors by sending it a `sortDescriptors` message.

9.47.88 `textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow.

9.47.89 `textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as Boolean`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow.

9.47.90 `toolTipForCell(cell as NSCellMBS, byref rect as NSRectMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS, mouseLocation as NSPointMBS) as String`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: When the cursor pauses over a given cell, the value returned from this method is displayed in a tooltip.

Notes: cell: The cell for which to generate a tooltip.
 rect: The proposed active area of the tooltip. To control the default active area, you can modify the rect parameter. By default, rect is computed as `cell.drawingRectForBounds(cellFrame)`.
 tc: The table column that contains cell.
 item: The item for which to display a tooltip.
 mouseLocation: The current mouse location in view coordinates.

If you don't want a tooltip at that location, return nil or the empty string.

9.47.91 typeSelectString(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as String

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns the string that is used for type selection for a given column and item.

Notes: tableColumn: A table column in the outline view.

item: An item in the outline view.

Return the string that is used for type selection. You may want to change what is searched for based on what is displayed, or simply return nil for that row and/or column to not be searched

Implement this method if you want to control the string that is used for type selection. You may want to change what is searched for based on what is displayed, or simply return nil to specify that the given row and/or column should not be searched. By default, all cells with text in them are searched.

The default value when this delegate method is not implemented is:

```
outlineView.preparedCellAtColumn(tableColumn, outlineView.rowForItem(item)).stringValue
```

and you can return this value from the event if you wish.

9.47.92 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to enable the table to update dragging items as they are dragged over the view.

Notes: draggingInfo: The dragging info object.

9.47.93 `validateDrop(info as NSDraggingInfoMBS, proposedItem as NSOutlineViewItemMBS, proposedChildIndex as Integer) as Integer`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Used by an outline view to determine a valid drop target.

Notes: info: An object that contains more information about this dragging operation.

item: The proposed parent.

index: The proposed child location.

Returns a value that indicates which dragging operation the data source will perform.

Based on the mouse position, the outline view will suggest a proposed drop location. The data source may ,Åüretarget,Åü a drop if desired by calling `setDropItem` and returning something other than `NSDragOperationNone`. You may choose to retarget for various reasons (for example, for better visual feedback when inserting into a sorted position).

Implementation of this method is optional.

9.47.94 `view(tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS) as NSViewMBS`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implemented to return the view used to display the specified item and column.

Notes: tableColumn: The table column, or nil if the row is a group row.

item: The item displayed by the returned view.

Return the view to display the specified column and row. Returning nil is acceptable, in which case a view is not shown at that location.

This method is required if you wish to use `NSView` objects instead of `NSCell` objects for the cells within an outline view. Cells and views cannot be mixed within the same outline view.

It is recommended that the implementation of this method first call the `NSTableView` method `makeViewWithIdentifier` passing, respectively, the `tableColumn` parameter,Åôs identifier and self as the owner to attempt to reuse a view that is no longer visible. The frame of the view returned by this method is not important, and is automatically set by the outline view.

The view's properties should be properly set up before returning the result.

When using Cocoa bindings, this method is optional if at least one identifier has been associated with the table view at design time. If this method is not implemented, the outline view automatically calls `makeViewWithIdentifier` with the `tableColumn` parameter,Åôs identifier and the outline view,Åôs delegate as parameters, to attempt to reuse a previous view or automatically unarchive a prototype associated with the table view.

The `autoresizingMask` of the returned view is automatically set to `NSViewHeightSizable` to resize properly on row height changes.

See also:

- 9.47.13 View as NSOutlineViewMBS

526

9.47.95 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the delegate that the cell specified by the column and item will be displayed.

Notes: cell: The cell.

tableColumn: The table column.

item: The item.

You can implement this method to modify cell to provide further setup for the cell in tableColumn and item. It is not safe to do drawing inside this method—you should only set up state for cell.

9.47.96 willDisplayOutlineCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the delegate that an outline view is about to display a cell used to draw the expansion symbol.

Notes: cell: The cell.

tableColumn: The table column.

item: The item.

Informs the event that outlineView is about to display cell—an expandable cell (a cell that has the expansion symbol)—for the column and item specified by tableColumn and item. The delegate can modify cell to alter its display attributes.

This method is not invoked when outlineView is about to display a non-expandable cell.

9.47.97 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.47.98 willTile

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The tableview will tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.47.99 writeItems(items() as NSOutlineViewItemMBS, pasteboard as NSPasteboardMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a drag operation is allowed.

Notes: items: An array of the items participating in the drag.

pasteboard: The pasteboard to which to write the drag data.

Returns true if the drag operation is allowed, otherwise false.

Invoked by outlineView after it has been determined that a drag should begin, but before the drag has been started.

To refuse the drag, return false. To start a drag, return true and place the drag data onto the pboard (data, owner, and so on). The drag image and other drag-related information will be set up and provided by the outline view once this call returns with true.

9.48 class NSOutlineViewItemMBS

9.48.1 class NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The class to subclass for your items.

Blog Entries

- [MBS Xojo Plugins, version 17.1pr4](#)

9.48.2 Methods

9.48.3 Constructor

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The constructor.

9.48.4 `sortedArrayUsingDescriptor(values() as NSOutlineViewItemMBS, sortDescriptor as NSSortDescriptorMBS) as NSOutlineViewItemMBS()`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a copy of the receiving array sorted as specified by a given a sort descriptor.

Notes: `sortDescriptor`: The `NSSortDescriptor` objects.

Returns a copy of the receiving array sorted as specified by `sortDescriptor`.

9.48.5 `sortedArrayUsingDescriptors(values() as NSOutlineViewItemMBS, sortDescriptors() as NSSortDescriptorMBS) as NSOutlineViewItemMBS()`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a copy of the receiving array sorted as specified by a given array of sort descriptors.

Notes: `sortDescriptors`: An array of `NSSortDescriptor` objects.

Returns a copy of the receiving array sorted as specified by `sortDescriptors`.

The first descriptor specifies the primary key path to be used in sorting the receiving array,Â’s contents. Any subsequent descriptors are used to further refine sorting of objects with duplicate values. See `NSSort-`

Descriptor for additional information.

9.48.6 Properties

9.48.7 Description as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The description text property.

Notes: (Read only property)

See also:

- 9.48.11 Description as String

554

9.48.8 Handle as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal object reference.

Notes: (Read and Write property)

9.48.9 valueForKey(key as String) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Get/set the value for the property identified by a given key.

Notes: key: The name of one of the receiver's properties.

(Read and Write computed property)

See also:

- 9.48.14 valueForKey(key as string) as Variant

555

9.48.10 Events

9.48.11 Description as String

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event when system queries description for object.

See also:

- 9.48.7 Description as String

554

9.48.12 setValueForKey(key as string, value as Variant)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Sets the property of the receiver specified by a given key to a given value.

Notes: value: The value for the property identified by key.

key: The name of one of the receiver's properties.

If key identifies a to-one relationship, relate the object specified by value to the receiver, unrelating the previously related object if there was one. Given a collection object and a key that identifies a to-many relationship, relate the objects contained in the collection to the receiver, unrelating previously related objects if there were any.

The search pattern that setValueForKey uses is described in Accessor Search Patterns in Key-Value Coding Programming Guide.

In a reference-counted environment, if the instance variable is accessed directly, value is retained.

9.48.13 setValueForUndefinedKey(key as string, value as Variant)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by setValueForKey when it finds no property for a given key.

Notes: value: The value for the key identified by key.

key: A string that is not equal to the name of any of the receiver's properties.

Subclasses can override this method to handle the request in some other way. The default implementation raises an NSUndefinedKeyException.

9.48.14 valueForKey(key as string) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Return the value for the property identified by a given key.

Notes: key: The name of one of the receiver's properties.

Returns the value for the property identified by key.

If event is not implemented, the search pattern that valueForKey uses to find the correct value to return is described in Accessor Search Patterns in Key-Value Coding Programming Guide.

See also:

- 9.48.9 valueForKey(key as String) as Variant

9.48.15 valueForUndefinedKey(key as string) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Invoked by `valueForKey` when it finds no property corresponding to a given key.

Notes: `key`: A string that is not equal to the name of any of the receiver's properties.

Subclasses can override this method to return an alternate value for undefined keys. The default implementation raises an `NSUndefinedKeyException`.

9.49 class NSOutlineViewMBS

9.49.1 class NSOutlineViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The class for a hierarchical list.

Notes: NSOutlineView is a subclass of NSTableView that uses a row-and-column format to display hierarchical data that can be expanded and collapsed, such as directories and files in a file system. A user can expand and collapse rows, edit values, and resize and rearrange columns.

Like a table view, an outline view does not store its own data, instead it retrieves data values as needed from a data source to which it has a weak reference.

MBS Plugin provides all events in NSOutlineControlMBS control.

An outline view has the following features:

- A user can expand and collapse rows.
- Each item in the outline view must be unique. In order for the collapsed state to remain consistent between reloads the item's pointer must remain the same and the item must maintain isEqual:sameness.
- The view gets data from a data source (see NSOutlineControlMBS).
- The view retrieves only the data that needs to be displayed.

Subclass of the NSTableViewMBS class.

Blog Entries

- [MBS Xojo Plugins, version 17.1pr4](#)

9.49.2 Methods

9.49.3 child(index as Integer, ofItem as NSOutlineViewItemMBS) as NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the specified child of an item.

Notes: index: The index of the child item in the parent.

item: The parent item whose child item you want to retrieve.

Returns the child item or nil if the item could not be found.

You can call this method on an outline view with either a static or dynamic data source. For an outline view whose contents are dynamic, this method may call out to the child event of the associated data source.

9.49.4 childIndexForItem(item as NSOutlineViewItemMBS) as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the child index of the specified item within its parent.

Notes: The performance of this method is O(1) at best and O(n) at worst.

9.49.5 collapseItem(item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Collapses a given item.

Notes: If item is not expanded or not expandable, does nothing
If collapsing takes place, posts item collapse notification.

See also:

- 9.49.6 collapseItem(item as NSOutlineViewItemMBS, collapseChildren as Boolean) 558

9.49.6 collapseItem(item as NSOutlineViewItemMBS, collapseChildren as Boolean)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Collapses a given item and, optionally, its children.

Notes: item: An item in the receiver.

Starting in OS X version 10.5, passing 'nil' will collapse each item under the root in the outline view.

collapseChildren: If true, recursively collapses item and its children. If NO, collapses item only (identical to collapseItem:).

For example, this method is invoked with the collapseChildren parameter set to true when a user Option-clicks the disclosure triangle for an item in the outline view (to collapse the item and all its contained items). For each item collapsed, posts an item collapsed notification.

See also:

- 9.49.5 collapseItem(item as NSOutlineViewItemMBS) 558

9.49.7 Constructor

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new outline view with size 100/100 and position 0/0

Example:

```
dim t as new NSOutlineViewMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.49.8 Constructor(Handle as Integer) 559
- 9.49.9 Constructor(left as Double, top as Double, width as Double, height as Double) 559

9.49.8 Constructor(Handle as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSOutlineView handle.

Example:

```
dim t as new NSOutlineViewMBS(0, 0, 100, 100)
```

```
dim v as new NSOutlineViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSOutlineViewMBS and the plugin retains this handle.

See also:

- 9.49.7 Constructor 559
- 9.49.9 Constructor(left as Double, top as Double, width as Double, height as Double) 559

9.49.9 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new outline view with the given size and position.

Example:

```
dim x as new NSOutlineViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.49.7 Constructor 559
- 9.49.8 Constructor(Handle as Integer) 559

9.49.10 `expandItem(item as NSOutlineViewItemMBS)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Expands a given item.

Notes: If item is not expandable or is already expanded, does nothing.

If expanding takes place, posts an item expanded notification.

See also:

- 9.49.11 `expandItem(item as NSOutlineViewItemMBS, expandChildren as Boolean)` 560

9.49.11 `expandItem(item as NSOutlineViewItemMBS, expandChildren as Boolean)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Expands a specified item and, optionally, its children.

Notes: item: An item in the receiver.

Starting in OS X version 10.5, passing 'nil' will expand each item under the root in the outline view.

expandChildren: If true, recursively expands item and its children. If false, expands item only (identical to `expandItem`).

For example, this method is invoked with the `expandChildren` parameter set to YES when a user Option-clicks the disclosure triangle for an item in the outline view (to expand the item and all its contained items). For each item expanded, posts an item expanded notification.

See also:

- 9.49.10 `expandItem(item as NSOutlineViewItemMBS)` 560

9.49.12 `frameOfOutlineCellAtRow(row as Integer) as NSRectMBS`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the frame of the outline cell for a given row.

Notes: row: The index of the row for which to return the frame.

Returns the frame of the outline cell for the row at index row, considering the current indentation and the value in the `indentationMarkerFollowsCell` property. If the row at index row is not an expandable row,

returns NSZeroRect.

You can override this method in a subclass to return a custom frame for the outline button cell. If your override returns an empty rect, no outline cell is drawn for that row. You might do that, for example, so that the disclosure triangle will not be shown for a row that should never be expanded.

9.49.13 insertItemsAtIndexes(indexes as NSIndexSetMBS, Parent as NSOutlineViewItemMBS, animationOptions as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Inserts new items at the given indexes in the given parent with the specified optional animations.

Notes: indexes: Indexes at which to insert items.

parent: The parent for the items, or nil if the parent is the root.

animationOptions: Animated slide effects used when inserting items.

This method parallels the insertRowsAtIndexes method of NSTableView and is used in a way similar to the insertObjects method of NSMutableArray. The method does nothing if parent is not expanded. The actual item values are determined by the data source,Ãs child event (which is called only after endUpdates to ensure data source integrity).

NSCell-based outline views must first call beginUpdates before calling this method.

You can call this method multiple times within the same beginUpdates/endUpdates block; new insertions move previously inserted new items, just like modifying an array. Inserting an index beyond what is available throws an exception.

9.49.14 insertRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Private method to block you from calling this method in NSTableViewMBS.

9.49.15 isExpandable(item as NSOutlineViewItemMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether a given item is expandable.

Notes: True if item is expandable—that is, item can contain other items, otherwise false.

9.49.16 `isItemExpanded(item as NSOutlineViewItemMBS)` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether a given item is expanded.

Notes: True if item is expanded, otherwise false.

9.49.17 `itemAtRow(row as Integer)` as NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the item associated with a given row.

Notes: row: The index of a row in the receiver.

Returns the item associated with row.

9.49.18 `levelForItem(item as NSOutlineViewItemMBS)` as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the indentation level for a given item.

Notes: The indentation level for item. If item is nil (which is the root item), returns -1. The levels are zero-based—that is, the first level of displayed items is level 0.

9.49.19 `levelForRow(row as Integer)` as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the indentation level for a given row.

Notes: The indentation level for row. For an invalid row, returns -1. The levels are zero-based—that is, the first level of displayed items is level 0.

9.49.20 `moveItemAtIndex(oldIndex as Integer, oldParent as NSOutlineViewItemMBS, newIndex as Integer, newParent as NSOutlineViewItemMBS)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Moves an item at a given index in the given parent to a new index in a new parent.

Notes: fromIndex: Index of the item to be moved.

oldParent: The parent of the item to be moved.

toIndex: Index in the new parent to which the item is moved.

newParent: The parent of the item after it is moved.

This method parallels the `moveRowAtIndex` method of `NSTableView`. The `newParent` can be the same as `oldParent` to reorder an item within the same parent.

`NSCell`-based outline views must first call `beginUpdates` before calling this method.

You can call this method multiple times within the same `beginUpdates/endUpdates` block. Moving from an invalid index, or to an invalid index, throws an exception.

9.49.21 `moveRowAtIndex(oldIndex as Integer, newIndex as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Private method to block you from calling this method in `NSTableViewMBS`.

9.49.22 `NSNotification as String`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted whenever a column is moved by user action in an `NSOutlineView` object.

Notes: The notification object is the `NSOutlineView` object in which a column moved. The `userInfo` dictionary contains the following information:

`NSOldColumn`: An `NSNumber` object containing the integer value of the column,Ãs original index

`NSNewColumn`: An `NSNumber` object containing the integer value of the column,Ãs present index

9.49.23 `NSNotification as String`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted whenever a column is resized in an `NSOutlineView` object.

Notes: The notification object is the `NSOutlineView` object in which a column was resized. The `userInfo` dictionary contains the following information:

`NSTableColumn`: The column that was resized.

NSOldWidth: An NSNumber object containing the column's original width

9.49.24 NSOutlineViewDisclosureButtonKey as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: This key is used by the outline view to create disclosure buttons that collapse and expand items.

Notes: The normal triangle disclosure button.

9.49.25 NSOutlineViewItemDidCollapseNotification as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted whenever an item is collapsed in an NSOutlineView object.

Notes: The notification object is the NSOutlineView object in which an item was collapsed. A collapsed item's children lose their status as being selected. The userInfo dictionary contains the following information:

NSObject: The item that was collapsed (an id)

9.49.26 NSOutlineViewItemDidExpandNotification as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted whenever an item is expanded in an NSOutlineView object.

Notes: The notification object is the NSOutlineView object in which an item was expanded. The userInfo dictionary contains the following information:

NSObject: The item that was expanded (an id)

9.49.27 NSOutlineViewItemWillCollapseNotification as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted before an item is collapsed (after the user clicks the arrow but before the item is collapsed).

Notes: The notification object is the NSOutlineView object that contains the item about to be collapsed. A collapsed item's children will lose their status as being selected. The userInfo dictionary contains the

following information:

NSObject: The item about to be collapsed (an id)

9.49.28 NSOutlineViewItemWillExpandNotification as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted before an item is expanded (after the user clicks the arrow but before the item is collapsed).

Notes: The notification object is the outline view that contains an item about to be expanded. The userInfo dictionary contains the following information:

NSObject: The item that is to be expanded (an id)

9.49.29 NSOutlineViewSelectionDidChangeNotification as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted after the outline view's selection changes.

Notes: The notification object is the outline view whose selection changed. This notification does not contain a userInfo dictionary.

9.49.30 NSOutlineViewSelectionIsChangingNotification as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Posted as the outline view's selection changes (while the mouse button is still down).

Notes: The notification object is the outline view whose selection is changing. This notification does not contain a userInfo dictionary.

9.49.31 NSOutlineViewShowHideButtonKey as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: This key is used by the outline view to create disclosure buttons that collapse and expand items.

Notes: The Show/Hide button.

9.49.32 numberOfChildrenOfItem(item as NSOutlineViewItemMBS) as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the number of children for the specified parent item.

Notes: item: The parent item.

Returns the number of children associated with the parent.

You can call this method on an outline view with either a static or dynamic data source. For an outline view whose contents are dynamic, this method may call out to the `numberOfChildrenOfItem` event of the associated data source.

9.49.33 parentForItem(item as NSOutlineViewItemMBS) as NSOutlineViewItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the parent for a given item.

Notes: item: The item for which to return the parent.

Returns the parent for item, or nil if the parent is the root.

9.49.34 reloadItem(item as NSOutlineViewItemMBS)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Reloads and redisplay the data for the given item.

Notes: Reloading the cell views associated with item occurs only in apps that link against macOS 10.12 and later.

This method may cause the outline view to change its selection without calling the `SelectionDidChange` event.

See also:

- 9.49.35 `reloadItem(item as NSOutlineViewItemMBS, reloadChildren as Boolean)` 566

9.49.35 reloadItem(item as NSOutlineViewItemMBS, reloadChildren as Boolean)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Reloads a given item and, optionally, its children.

Notes: item: An item in the receiver.

Starting in OS X version 10.5, passing 'nil' will reload everything under the root in the outline view.

reloadChildren: If true, recursively reloads item and its children. If false, reloads item only (identical to reloadItem).

It is not necessary, or efficient, to reload children if the item is not expanded.
See also:

- 9.49.34 reloadItem(item as NSOutlineViewItemMBS) 566

9.49.36 removeItemsAtIndexes(indexes as NSIndexSetMBS, Parent as NSOutlineViewItemMBS, animationOptions as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Removes items at the given indexes in the given parent with the specified optional animations.

Notes: indexes: Indexes of the items to be removed.

parent: The parent of the items to be removed.

animationOptions: Animated slide effects used when removing items.

This method parallels the removeRowsAtIndexes method of NSTableView and is used in a way similar to the removeObjectAtIndexes method of NSMutableArray. The method does nothing if parent is not expanded. If any of the child items is expanded, then all of its child rows are also be removed.

NSCell-based outline views must first call beginUpdates before calling this method.

You can call this method multiple times within the same beginUpdates/endUpdates block; changes work just like modifying an array. Removing an item at an index beyond what is available throws an exception.

9.49.37 removeRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Private method to block you from calling this method in NSTableViewMBS.

9.49.38 rowForItem(item as NSOutlineViewItemMBS) as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the row associated with a given item.

Notes: The row associated with item, or -1 if item is nil or cannot be found.

9.49.39 `setDropItem(item as NSOutlineViewItemMBS, dropChildIndex as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Used to `setDropItem`, a proposed drop.

Notes: `item`: The target item.

`index`: The drop index.

For example, to specify a drop on `someOutlineItem`, you specify `item` as `someOutlineItem` and `index` as `NSOutlineViewDropOnItemIndex`. To specify a drop between child 2 and 3 of `someOutlineItem`, you specify `item` as `someOutlineItem` and `index` as 3 (children are a zero-based index). To specify a drop on an un-expandable `someOutlineItem`, you specify `item` as `someOutlineItem` and `index` as `NSOutlineViewDropOnItemIndex`.

9.49.40 Properties

9.49.41 `autoresizesOutlineColumn` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the outline view resizes its outline column when the user expands or collapses items.

Notes: The outline column contains the cells with the expansion symbols and is generally the first column. The default value of this property is `true`, which causes the outline column to be resized.

The outline column is resized based on how many indentation levels are exposed or hidden. For example, if expanding a row exposes a single indentation level, the outline column width is increased by one `indentationPerLevel`.

(Read and Write property)

9.49.42 `autosaveExpandedItems` as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the expanded items are automatically saved across launches of the app.

Notes: When the value of this property is `true`, the outline view saves the state of its expanded items and restores that state the next time the user launches the app. (If the outline view's `autosaveName` property is `nil`, or if you have not implemented the `itemForPersistentObject` and `persistentObjectForItem` events, this setting is ignored and outline information is not saved.) The configuration data is saved separately for each user and for each app. The default value of this property is `NO`.

You can have separate settings for the `autosaveExpandedItems` and `autosaveTableColumns` properties, so you could, for example, save expanded item information, but not table column positions.

(Read and Write property)

9.49.43 indentationMarkerFollowsCell as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the indentation marker symbol displayed in the outline column should be indented along with the cell contents.

Notes: When the value of this property is true, the indentation marker is indented along with the cell contents. When the value is false, the marker is always displayed left-justified in the column. The default value of this property is true.

(Read and Write property)

9.49.44 indentationPerLevel as Double

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The per-level indentation, measured in points.

Notes: (Read and Write property)

9.49.45 outlinetableColumn as NSTableColumnMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The table column in which hierarchical data is displayed.

Notes: Each level of hierarchical data is indented by the amount specified by the indentationPerLevel property (the default is 16.0), and decorated with the indentation marker (disclosure triangle) on rows that are expandable. Outline table column data is archived with the rest of the outline view's state information.

Attempts to set the value of this property to nil are silently ignored.

(Read and Write property)

9.49.46 stronglyReferencesItems as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the outline view retains and releases the objects returned from its data source.

Notes: For Xojo applications we highly recommend to keep all items in arrays in Xojo, so they are not released by Xojo too early!

When the value of this property is true, the outline view retains and releases the objects returned to it from dataSource. When the value is NO, the outline view treats the objects as opaque items and assumes that

the client has a retain on them. The default value is YES for applications linked on macOS 10.12 and later, and false for applications linked on earlier versions of macOS. If you require the legacy behavior and your app links in macOS 10.12 or later, the value of this property must be explicitly set to NO in code, because it is not encoded in the nib. In general, this is required if the items themselves create a retain cycle.
(Read and Write property)

9.49.47 `userInterfaceLayoutDirection` as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The user interface layout direction.

Notes: When set to `NSUserInterfaceLayoutDirectionRightToLeft`, the outline view displays the disclosure triangle to the right of the cell instead of the left. The default value is `NSUserInterfaceLayoutDirectionLeftToRight`.

(Read and Write property)

9.49.48 Constants

Constants

Constant	Value	Description
<code>NSOutlineViewDropOnItemIndex</code>	-1	May be used as a valid child index of a drop target item. In this case, the drop will happen directly on the target item.

9.50 control NSPopUpButtonControlMBS

9.50.1 control NSPopUpButtonControlMBS

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The Xojo control for a NSPopUpButtonControl.

Notes: This control embeds a special NSPopUpButtonControl subclass.

Designed for Xojo 2013r1 and newer. May work on Xojo 2012, but not perfectly.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo / Real Studio Plugins, version 13.2pr6](#)

9.50.2 Properties

9.50.3 View as NSPopUpButtonMBS

Plugin Version: 13.2, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.

(Read only property)

9.50.4 Events

9.50.5 Action

Plugin Version: 13.2, Platform: macOS, Targets: .

Function: The action event.

9.50.6 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.50.7 Close

Plugin Version: 13.2, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.50.8 ConstructContextMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.50.9 ContextualMenuItem(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.50.10 didCloseContextMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextMenu event.

9.50.11 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named MenuBarSelected.

9.50.12 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.50.13 GotFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named FocusReceived.

Notes:

This only fires if the control itself got focus and not a sub control.

9.50.14 LostFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.50.15MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return `False`, the system handles the `MouseDown` so the above event handlers do not get called.

9.50.16 `MouseDown(x as Integer, y as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to `x`, `y`.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.50.17 `MouseUp(x as Integer, y as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the `x` and `y` parameters to determine if the mouse button was released within the control's boundaries.

9.50.18 `Open`

Plugin Version: 13.2, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.

In Xojo version 2021r3 and newer this event is named `Opening`.

9.50.19 `ScaleFactorChanged(NewFactor as Double)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.50.20 `willShowContextMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)`

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.51 class NSPopUpButtonMBS

9.51.1 class NSPopUpButtonMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: An NSPopUpButton object controls a pop-up menu or a pull-down menu from which a user can select an item.

Example:

```
// get cocoa view for the popupmenu
dim p as NSPopUpButtonMBS = PopupMenu1.NSPopUpButtonMBS

// find a menu entry
dim it as NSMenuItemMBS = p.itemAtIndex(0)

// get a picture
dim pic as Picture = LogoMBS(500)
dim img as new NSImageMBS(pic)
img.setSize 16,16

// and assign icon
it.image = img
```

Notes: The NSPopUpButton class defines objects that implement the pop-up and pull-down menus of the graphical user interface.

An NSPopUpButton object uses an NSPopUpButtonCell object to implement its user interface.

Note that while a menu is tracking, adding, removing, or changing items on the menu is not reflected. Subclass of the NSButtonMBS class.

Blog Entries

- [Apply fonts to font PopupMenu](#)
- [MBS Xojo / Real Studio Plugins, version 14.3pr4](#)
- [More notes](#)

9.51.2 Methods

9.51.3 addItemWithTitle(titles() as string)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Adds multiple items to the end of the menu.

Notes: `temTitles`: An array of strings containing the titles of the items you want to add. Each string in the array should be unique. If an item with the same title already exists in the menu, the existing item is removed and the new one is added.

If you want to move an item, it's better to invoke `removeItemWithTitle:` explicitly and then send this method. After adding the items, this method uses the `synchronizeTitleAndSelectedItem` method to make sure the item being displayed matches the currently selected item.

Since this method searches for duplicate items, it should not be used if you are adding items to an already populated menu with more than a few hundred items. Add items directly to the receiver's menu instead.

9.51.4 addItemWithTitle(title as string)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Adds an item with the specified title to the end of the menu.

Notes: `title`: The title of the menu-item entry. If an item with the same title already exists in the menu, the existing item is removed and the new one is added.

If you want to move an item, it's better to invoke `removeItemWithTitle` explicitly and then send this method. After adding the item, this method calls the `synchronizeTitleAndSelectedItem` method to make sure the item being displayed matches the currently selected item.

Since this method searches for duplicate items, it should not be used if you are adding an item to an already populated menu with more than a few hundred items. Add items directly to the receiver's menu instead.

9.51.5 Constructor

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with size 100/100 and position 0/0

Example:

```
dim t as new NSPopUpButtonMBS
```

Notes: On success the `handle` property is not zero.

See also:

- 9.51.6 Constructor(Handle as Integer) 578
- 9.51.7 Constructor(left as Double, top as Double, width as Double, height as Double) 578

- 9.51.8 Constructor(left as Double, top as Double, width as Double, height as Double, pullsDown as boolean) 579

9.51.6 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSPopUpButton handle.

Example:

```
dim t as new NSPopUpButtonMBS(0, 0, 100, 100)
dim v as new NSPopUpButtonMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSPopUpButton and the plugin retains this handle.

See also:

- 9.51.5 Constructor 577
- 9.51.7 Constructor(left as Double, top as Double, width as Double, height as Double) 578
- 9.51.8 Constructor(left as Double, top as Double, width as Double, height as Double, pullsDown as boolean) 579

9.51.7 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new popup button with the given size and position.

Example:

```
dim x as new NSPopUpButtonMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.51.5 Constructor 577
- 9.51.6 Constructor(Handle as Integer) 578
- 9.51.8 Constructor(left as Double, top as Double, width as Double, height as Double, pullsDown as boolean) 579

9.51.8 Constructor(left as Double, top as Double, width as Double, height as Double, pullsDown as boolean)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates an NSPopUpButton object initialized to the specified dimensions.

Notes: left,top,width,height: The frame rectangle for the button, specified in the parent view's coordinate system.

pullsDown: true if you want the receiver to display a pull-down menu; otherwise, false if you want it to display a pop-up menu.

See also:

- 9.51.5 Constructor 577
- 9.51.6 Constructor(Handle as Integer) 578
- 9.51.7 Constructor(left as Double, top as Double, width as Double, height as Double) 578

9.51.9 indexOfItem(item as NSMenuItemMBS) as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the specified menu item.

Notes: item: The menu item whose index you want.

Returns the index of the item or -1 if no such item was found.

This method invokes the method of the same name of its NSPopUpButtonCell object.

9.51.10 indexOfItemWithTag(tag as Integer) as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the menu item with the specified tag.

Notes: tag: The tag of the menu item you want.

Returns the index of the item or -1 if no item with the specified tag was found.

This method invokes the method of the same name of its NSPopUpButtonCell object.

9.51.11 `indexOfItemWithTitle(title as string) as Integer`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the item with the specified title.

Notes: title: The title of the item you want.

Returns the index of the item or -1 if no item with the specified title was found.

9.51.12 `indexOfSelectedItem as Integer`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the item last selected by the user.

Notes: The index of the selected item, or -1 if no item is selected.

9.51.13 `insertItemWithTitle(title as string, atIndex as Integer)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Inserts an item at the specified position in the menu.

Notes: title: The title of the new item. If an item with the same title already exists in the menu, the existing item is removed and the new one is added

index: The zero-based index at which to insert the item. Specifying 0 inserts the item at the top of the menu.

If you want to move an item, it's better to invoke `removeItemWithTitle:` explicitly and then send this method. After adding the item, this method uses the `synchronizeTitleAndSelectedItem` method to make sure the item displayed matches the currently selected item.

Since this method searches for duplicate items, it should not be used if you are adding an item to an already populated menu with more than a few hundred items. Add items directly to the receiver's menu instead.

9.51.14 `itemAtIndex(index as Integer) as NSMenuItemMBS`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the menu item at the specified index.

Notes: index: The index of the item you want.

Returns the menu item, or nil if no item exists at the specified index.

9.51.15 itemWithTitle(title as string) as NSMenuItemMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the menu item with the specified title.

Notes: title: The title of the menu item you want.

The menu item, or nil if no item with the specified title exists in the menu.

9.51.16 lastItem as NSMenuItemMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the last item in the menu.

9.51.17 removeAllItems

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Removes all items in the receiver's item menu.

Notes: After removing the items, this method uses the synchronizeTitleAndSelectedItem method to refresh the menu.

9.51.18 removeItemAtIndex(index as Integer)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Removes the item at the specified index.

Notes: index: The zero-based index indicating which item to remove. Specifying 0 removes the item at the top of the menu.

After removing the item, this method uses the synchronizeTitleAndSelectedItem method to make sure the title displayed matches the currently selected item.

9.51.19 `removeItemWithTitle(title as string)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Removes the item with the specified title from the menu.

Notes: `title`: The title of the item you want to remove. If no menu item exists with the specified title, this method triggers an assertion.

This method removes the first item it finds with the specified name. This method then uses `synchronizeTitleAndSelectedItem` to refresh the menu.

9.51.20 `selectItem(item as NSMenuItemMBS)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Selects the specified menu item.

Notes: `item`: The menu item to select, or nil if you want to deselect all menu items.

9.51.21 `selectItemAtIndex(index as Integer)`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Selects the item in the menu at the specified index.

Notes: `index`: The index of the item you want to select, or -1 you want to deselect all menu items.

9.51.22 `selectItemWithTag(tag as Integer) as boolean`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Selects the menu item with the specified tag.

Notes: `tag`: The tag of the item you want to select.

Returns true if the item was successfully selected; otherwise, false.

If no item with the specified tag is found, this method returns false and leaves the menu state unchanged.

You typically assign tags to menu items from Interface Builder, but you can also assign them programmatically using the `setTag:` method of `NSMenuItem`.

Available in Mac OS X v10.4 and later.

9.51.23 selectItemWithTitle(title as string)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Selects the item with the specified title.

Notes: title: The title of the item to select. If you specify an empty string, or a string that does not match the title of a menu item, this method deselects the currently selected item.

9.51.24 setTitle(title as string)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Sets the string displayed in the receiver when the user isn't pressing the mouse button.

Notes: If the receiver displays a pop-up menu, this method changes the current item to be the item with the specified title, adding a new item by that name if one does not already exist. If the receiver displays a pull-down list, this method sets its title to the specified string.

9.51.25 synchronizeTitleAndSelectedItem

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Ensures that the item being displayed by the receiver agrees with the selected item.

Notes: If there's no selected item, this method selects the first item in the item menu and sets the receiver's item to match. For pull-down menus, this method makes sure that the first item is being displayed (the NSPopUpButtonCell object must be set to use the selected menu item, which happens by default).

9.51.26 titleOfSelectedItem as string

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the title of the item last selected by the user.

Notes: Returns the title of the selected menu item, or an empty string if no item is selected.

9.51.27 Properties

9.51.28 arrowPosition as Integer

Plugin Version: 14.3, Platform: macOS, Targets: Desktop only.

Function: The position of the arrow displayed.

Notes: If you specify `NSPopUpNoArrow`, the receiver displays no arrow. `NSPopUpArrowAtCenter` displays the arrow centered horizontally within the cell. `NSPopUpArrowAtBottom` displays the arrow at the edge of the cell. This method works with `setPreferredEdge:` to determine the exact location and orientation of the arrow. For more information, see `setPreferredEdge:`.

This method is ignored unless the receiver is a pull-down list with a beveled border.
(Read and Write property)

9.51.29 `autoenablesItems` as boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver automatically enables and disables its items every time a user event occurs.
Notes: true if the receiver automatically enables and disables items; otherwise, false. The default value is true.

(Read and Write property)

9.51.30 `menu` as `NSMenuItem`

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the pop-up button's associated menu.

Notes: (Read and Write property)

9.51.31 `numberOfItems` as Integer

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the number of items in the menu.

Notes: (Read only property)

9.51.32 `pullsDown` as boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating the behavior of the control's menu.

Notes: True if the menu behaves like a pull-down menu; otherwise, false if it behaves like a pop-up menu.
(Read and Write property)

9.51.33 selectedItem as NSMenuItemMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Returns the menu item last selected by the user.

Notes: Returns the menu item that is currently selected, or nil if no item is selected.

The last selected menu item is the one that was highlighted when the user released the mouse button. It is possible for a pull-down menu's selected item to be its first item.

(Read and Write computed property)

9.51.34 Constants

Arrow Position Constants

Constant	Value	Description
NSPopUpArrowAtBottom	2	Arrow is drawn at the edge of the button, pointing toward the preferredEdge.
NSPopUpArrowAtCenter	1	Arrow is centered vertically, pointing toward the preferredEdge.
NSPopUpNoArrow	0	Does not display any arrow in the receiver.

9.52 class NSProgressIndicatorMBS

9.52.1 class NSProgressIndicatorMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The NSProgressIndicator class lets an application display a progress indicator to show that a lengthy task is under way.

Notes: Some progress indicators are indeterminate and do nothing more than spin to show that the application is busy. Others are determinate and show the percentage of the task that has been completed.

This also eliminates the hung appearance is you move the window or open a menu. Unlike the RB progress controls that pause when you do these things, the Cocoa version continues to update.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard. Subclass of the NSViewMBS class.

Blog Entries

- [MonkeyBread Software Releases the MBS REALbasic plug-ins 8.4](#)

Xojo Developer Magazine

- [6.6, page 8: News](#)

9.52.2 Methods

9.52.3 Constructor

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new progress indicator with size 100/100 and position 0/0

Example:

```
dim t as new NSProgressIndicatorMBS
```

Notes: On success the handle property is not zero.

See also:

- [9.52.4 Constructor\(Handle as Integer\)](#) 587
- [9.52.5 Constructor\(left as Double, top as Double, width as Double, height as Double\)](#) 587

9.52.4 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSProgressIndicator handle.

Example:

```
dim t as new NSProgressIndicatorMBS(0, 0, 100, 100)
dim v as new NSProgressIndicatorMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSProgressIndicator and the plugin retains this handle.

See also:

- 9.52.3 Constructor 586
- 9.52.5 Constructor(left as Double, top as Double, width as Double, height as Double) 587

9.52.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new progress indicator with the given size and position.

Example:

```
dim x as new NSProgressIndicatorMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.52.3 Constructor 586
- 9.52.4 Constructor(Handle as Integer) 587

9.52.6 incrementBy(delta as Double)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Advances the progress bar of a determinate progress indicator by the specified amount.

Notes: The amount by which to increment the progress bar. For example, if you want to advance a progress bar from 0.0 to 100.0 in 20 steps, you would invoke incrementBy 20 times with a delta value of 5.0.

9.52.7 sizeToFit

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: This action method resizes the receiver to an appropriate size depending on what style returns.

Notes: Use this after you set style to re-size the receiver.

9.52.8 startAnimation

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Starts the animation of an indeterminate progress indicator.

Notes: Does nothing for a determinate progress indicator.

9.52.9 stopAnimation

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Stops the animation of an indeterminate progress indicator.

Notes: Does nothing for a determinate progress indicator.

9.52.10 Properties

9.52.11 controlSize as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The size of the receiver.

Notes: (Read and Write computed property)

9.52.12 controlTint as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The receiver's control tint.

Notes: (Read and Write computed property)

9.52.13 doubleValue as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: A value that indicates the current extent of the progress bar of a determinate progress indicator.

Notes: The value representing the current extent of a determinate progress bar. For example, a determinate progress indicator goes from 0.0 to 100.0 by default. If the progress bar has advanced halfway across the view, the value returned by doubleValue would be 50.0. An indeterminate progress indicator does not use this value.

(Read and Write computed property)

9.52.14 isBezeled as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the receiver's frame has a bezel.

Notes: True if the receiver's frame has a three-dimensional bezel; otherwise, false.

(Read and Write computed property)

9.52.15 isDisplayedWhenStopped as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the receiver shows itself even when it's not animating.

Notes: True if the progress indicator shows itself even when it's not animating. By default, this returns true if style is NSProgressIndicatorBarStyle and false if style is NSProgressIndicatorSpinningStyle.

(Read and Write computed property)

9.52.16 isIndeterminate as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver is indeterminate.

Notes: This method only has an effect if style returns NSProgressIndicatorBarStyle. If style returns NSProgressIndicatorSpinningStyle, the indicator is always indeterminate, regardless of what you pass to this method.

(Read and Write computed property)

9.52.17 maxValue as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The maximum value for the receiver.

Notes: The maximum value of the progress indicator. An indeterminate progress indicator does not use this value.

(Read and Write computed property)

9.52.18 `minValue` as `Double`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The minimum value for the receiver.

Notes: An indeterminate progress indicator does not use this value.

(Read and Write computed property)

9.52.19 `style` as `Integer`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The style of the progress indicator (bar or spinning).

Notes: Either `NSProgressIndicatorBarStyle` or `NSProgressIndicatorSpinningStyle`.

(Read and Write computed property)

9.52.20 `usesThreadedAnimation` as `boolean`

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: A hint as to whether the receiver should implement animation of the progress indicator in a separate thread.

Notes: Value is true to indicate that animation of the progress indicator should occur in a separate thread; otherwise, false. This value is only a hint and may be ignored.

If the application becomes multithreaded as a result of an invocation of this method, the application's performance could become noticeably slower.

(Read and Write computed property)

9.52.21 `Constants`

Constants

Constant	Value	Description
NSBlueControlTint	1	One of the constants to specify a the control tint. Aqua control tint
NSClearControlTint	7	One of the constants to specify a the control tint. Clear control tint
NSDefaultControlTint	0	One of the constants to specify a the control tint. The current default tint setting
NSGraphiteControlTint	6	One of the constants to specify a the control tint. Graphite control tint
NSMiniControlSize	2	One of the values for the ControlSize property. The control has a smaller size than NSSmallControlSize.
NSProgressIndicatorBarStyle	0	One of the constants for the style property. A rectangular indicator that can be determinate or indeterminate
NSProgressIndicatorPreferredAquaThickness	12	One of the constants to specify the height of a progress indicator.
NSProgressIndicatorPreferredLargeThickness	18	One of the constants to specify the height of a progress indicator.
NSProgressIndicatorPreferredSmallThickness	10	One of the constants to specify the height of a progress indicator.
NSProgressIndicatorPreferredThickness	14	One of the constants to specify the height of a progress indicator.
NSProgressIndicatorSpinningStyle	1	One of the constants for the style property. A small square indicator that can be indeterminate only .
NSRegularControlSize	0	One of the values for the ControlSize property. The control is sized as regular.
NSSmallControlSize	1	One of the values for the ControlSize property. This constant is for controls that cannot be resized in one direction. push buttons, radio buttons, checkboxes, sliders, scroll bars, pop-up menus, tabs, and progress indicators. You should use a small system font for these controls.

9.53 class NSScrollerMBS

9.53.1 class NSScrollerMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: An NSScroller object controls scrolling of a document view within an NSScrollView’s clip view (or potentially another kind of container view).

Notes: It typically displays a pair of buttons that the user can click to scroll by a small amount (called a line increment or decrement) and Alt-click to scroll by a large amount (called a page increment or decrement), plus a slot containing a knob that the user can drag directly to the desired location. The knob indicates both the position within the document view and, by varying in size within the slot, the amount visible relative to the size of the document view. You can configure whether an NSScroller object uses scroll buttons, but it always draws the knob when there’s room for it.

Don’t use an NSScroller when an NSSlider would be better. A slider represents a range of values for something in the application and lets the user choose a setting. A scroller represents the relative position of the visible portion of a view and lets the user choose which portion to view.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard.

Subclass of the NSControlMBS class.

Blog Entries

- [Tip of the day: Autohide scrollbar](#)
- [Custom Scrollbars for Real Studio Cocoa Apps](#)
- [MBS Real Studio Plugins, version 11.3pr1](#)
- [MonkeyBread Software Releases the MBS REALbasic plug-ins 8.4](#)

Xojo Developer Magazine

- [6.6, page 8: News](#)

9.53.2 Methods

9.53.3 checkSpaceForParts

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Checks to see if there is enough room in the receiver to display the knob and buttons.

Notes: usableParts returns the state calculated by this method. You should never need to invoke this method; it’s invoked automatically whenever the NSScroller’s size changes.

9.53.4 Constructor

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new scroller with size 100/100 and position 0/0

Example:

```
dim t as new NSScrollerMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.53.5 Constructor(Handle as Integer) 593
- 9.53.6 Constructor(left as Double, top as Double, width as Double, height as Double) 593

9.53.5 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSScroller handle.

Example:

```
dim t as new NSScrollerMBS(0, 0, 100, 100)
```

```
dim v as new NSScrollerMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSScroller and the plugin retains this handle.

See also:

- 9.53.4 Constructor 593
- 9.53.6 Constructor(left as Double, top as Double, width as Double, height as Double) 593

9.53.6 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new scroller with the given size and position.

Example:

```
dim x as new NSScrollerMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.53.4 Constructor 593
- 9.53.5 Constructor(Handle as Integer) 593

9.53.7 drawArrow(Arrow as Integer, highlight as boolean)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Draws the scroll button indicated by arrow, which is either NSScrollerIncrementArrow (the down or right scroll button) or NSScrollerDecrementArrow (up or left).

Notes: If flag is true, the button is drawn highlighted; otherwise it's drawn normally. You should never need to invoke this method directly, but may wish to override it to customize the appearance of scroll buttons. Calling this method on CustomNSScrollerMBS objects does not trigger the drawArrow event.

9.53.8 drawKnob

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Draws the knob.

Notes: You should never need to invoke this method directly, but may wish to override it to customize the appearance of the knob.

Calling this method on CustomNSScrollerMBS objects does not trigger the drawKnob event.

9.53.9 drawKnobSlotInRect(slotRect as NSRectMBS, highlight as boolean)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Draws the knob slot in the rectangle.

Notes: Calling this method on CustomNSScrollerMBS objects does not trigger the drawKnobSlotInRect event.

9.53.10 drawParts

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Caches images for the scroll buttons and knob.

Notes: It's invoked only once when the NSScroller is created. You may want to override this method if you alter the look of the NSScroller, but you should never invoke it directly.

Calling this method on CustomNSScrollerMBS objects does not trigger the drawParts event.

9.53.11 highlight(flag as boolean)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Highlights or unhighlights the scroll button the user clicked.

Notes: The receiver invokes this method while tracking the mouse; you should not invoke it directly. If flag is true, the appropriate part is drawn highlighted; otherwise it's drawn normally.

9.53.12 hitPart as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns a part code indicating the manner in which the scrolling should be performed.

Notes: This method is typically invoked by an NSScrollView to determine how to scroll its document view when it receives an action message from the NSScroller.

9.53.13 isCompatibleWithOverlayScrollers as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether the class is compatible with overlay scroller style and behavior.

Notes: Returns true if the the class is compatible with overlay scroller style and behavior, otherwise false. The plugin implements this so you can query the value.

9.53.14 NSPreferredScrollerStyleDidChangeNotification as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: A notification name.

Notes: Posted if the preferred scroller style changes.

Available in Mac OS X v10.7 and later.

9.53.15 preferredScrollerStyle as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the style of scrollers that applications should use wherever possible.

Notes: The preferred scroller style is determined by the Appearance preference panel’s “Show scroll bars” setting for the current user, and—when the user’s preference is set to “Automatically based on input device”—by the set of built-in and connected pointing devices and the user’s scroll capability preference settings for them. The preferred scroller style may therefore change over time, and applications should be prepared to adapt their user interfaces to the new scroller style if needed.

In most cases, updating to a new scroller style is automatic: When the preferred scroller style changes, AppKit notifies all NSScrollView instances, sending `setScrollerStyle` to each with the new style, which causes each scroll view to automatically re-tile (update its layout) to adapt to the new scroller style. Some NSScrollView instances may refuse the new scroller style setting if they cannot accommodate it for compatibility reasons (the presence of accessory views or legacy scroller subclasses prevent use of overlay scrollers), but most instances will switch to the specified new preferred scroller style.

If you need to be notified of changes to the preferred scroller style, you can register to receive `NSPreferredScrollerStyleDidChangeNotification` notifications.

Available in Mac OS X v10.7 and later.

9.53.16 `rectForPart(part as Integer)` as `NSRectMBS`

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle occupied by aPart, which for this method is interpreted literally rather than as an indicator of scrolling direction.

Notes: Note the interpretations of `NSScrollerDecrementPage` and `NSScrollerIncrementPage`. The actual part of an NSScroller that causes page-by-page scrolling varies, so as a convenience these part codes refer to useful parts different from the scroll buttons.

Returns `NSRectMBS.Zero` if the part requested isn’t present on the receiver.

9.53.17 `scrollerWidth` as `Double`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the width of “normal-sized” instances.

Notes: NSScrollView uses this value to lay out its components. Subclasses that use a different width should override this method.

9.53.18 scrollerWidthForControlSize(controlSize as Integer) as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the width of the scroller based on controlSize.

9.53.19 setFloatValue(aFloat as Double, proportion as Double)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Sets the position of the knob to aFloat, which is a value from 0.0 (indicating the top or left end) to 1.0 (the bottom or right end).

Notes: Also sets the proportion of the knob slot filled by the knob to knobProp, also a value from 0.0 (minimal size) to 1.0 (fills the slot).

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.5.

9.53.20 testPart(p as NSPointMBS) as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the part that would be hit by a mouse-down event at p (expressed in the window's coordinate system).

Notes: Note the interpretations of NSScrollerDecrementPage and NSScrollerIncrementPage. The actual part of an NSScroller that causes page-by-page scrolling varies, so as a convenience these part codes refer to useful parts different from the scroll buttons.

9.53.21 trackKnob(theEvent as NSEventMBS)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Tracks the knob and sends action messages to the receiver's target.

Notes: This method is invoked automatically when the receiver receives theEvent mouse-down event in the knob; you should not invoke it directly.

9.53.22 trackScrollButtons(theEvent as NSEventMBS)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Tracks the scroll buttons and sends action messages to the receiver's target.

Notes: This method is invoked automatically when the receiver receives theEvent mouse-down event in a scroll button; you should not invoke this method directly.

9.53.23 usableParts as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns a value indicating which parts of the receiver are displayed and usable.

9.53.24 Properties

9.53.25 arrowsPosition as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The location of the scroll buttons within the receiver.

Notes: Can be NSScrollerArrowsMaxEnd, NSScrollerArrowsMinEnd, NSScrollerArrowsDefaultSetting or NSScrollerArrowsNone.

(Read and Write computed property)

9.53.26 controlSize as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The size of the receiver.

Notes: Can be NSRegularControlSize, NSMiniControlSize or NSSmallControlSize.

(Read and Write computed property)

9.53.27 controlTint as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The receiver's control tint.

Notes: Can be NSGraphiteControlTint, NSBlueControlTint, NSClearControlTint or NSDefaultControlTint.

(Read and Write computed property)

9.53.28 knobProportion as Double

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: The portion of the knob slot the knob should fill, as a floating-point value from 0.0 (minimal size) to 1.0 (fills the slot).

Notes: Can be set on Mac OS X 10.5 directly.
(Read and Write computed property)

9.53.29 knobStyle as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The knob style.

Notes: The value of this property does not affect legacy scrollers. `NSScrollerKnobStyleDefault` is appropriate for a wide range of content, but in some cases choosing an alternative knob style may enhance visibility of the scroller knob atop some kinds of content.

Available in Mac OS X v10.7 and later.

use constants: `NSScrollerKnobStyleDefault`, `NSScrollerKnobStyleDark` or `NSScrollerKnobStyleLight`.
(Read and Write computed property)

9.53.30 scrollerStyle as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The scroller style.

Notes: or a scroller that's managed by an `NSScrollView` object, the setter is automatically invoked by the scroll view with the appropriate setting, according to the user's Appearance preference settings and possibly what pointing device(s) are present (see `preferredScrollerStyle`).

Available in Mac OS X v10.7 and later.
(Read and Write computed property)

9.53.31 Constants

Constants

Constant	Value	Description
<code>NSAllScrollerParts</code>	2	One of the constants to specify which parts of the scroller are visible. Scroller has at least a knob, possibly also scroll buttons.
<code>NSBlueControlTint</code>	1	One of the constants to specify a the control tint. Aqua control tint
<code>NSClearControlTint</code>	7	One of the constants to specify a the control tint. Clear control tint
<code>NSDefaultControlTint</code>	0	One of the constants to specify a the control tint. The current default tint setting
<code>NSGraphiteControlTint</code>	6	One of the constants to specify a the control tint. Graphite control tint
<code>NSMiniControlSize</code>	2	One of the values for the <code>ControlSize</code> property. The control has a smaller size than <code>NSSmallControlSize</code> .
<code>NSNoScrollerParts</code>	0	One of the constants to specify which parts of the scroller are visible. Scroller has neither a knob nor scroll buttons, only the knob slot.
<code>NSOnlyScrollerArrows</code>	1	One of the constants to specify which parts of the scroller are visible. Scroller has only scroll buttons, no knob.
<code>NSRegularControlSize</code>	0	One of the values for the <code>ControlSize</code> property. The control is sized as regular.
<code>NSScrollerArrowsDefaultSetting</code>	0	One of the constants to specify where the scroller's buttons appear with the <code>arrowsPosition</code> property. Contains the information from the <code>AppleScrollBarVariant</code> default value.
<code>NSScrollerArrowsMaxEnd</code>	0	One of the constants to specify where the scroller's buttons appear with the <code>arrowsPosition</code> property. Buttons at bottom or right. This constant has been deprecated.
<code>NSScrollerArrowsMinEnd</code>	1	One of the constants to specify where the scroller's buttons appear with the <code>arrowsPosition</code> property. Buttons at top or left. This has been deprecated.
<code>NSScrollerArrowsNone</code>	2	One of the constants to specify where the scroller's buttons appear with the <code>arrowsPosition</code> property. No buttons.
<code>NSScrollerDecrementArrow</code>	1	One of the constants describe the two scroller buttons and are used by <code>drawArrow</code> . The down or right scroll button.
<code>NSScrollerDecrementLine</code>	4	One of the constants to specify the different parts of the scroller. Up or left by a small amount.
<code>NSScrollerDecrementPage</code>	1	One of the constants to specify the different parts of the scroller. Up or left by a large amount.
<code>NSScrollerIncrementArrow</code>	0	One of the constants describe the two scroller buttons and are used by <code>drawArrow</code> . The up or left scroll button.
<code>NSScrollerIncrementLine</code>	5	One of the constants to specify the different parts of the scroller. Down or right by a small amount.
<code>NSScrollerIncrementPage</code>	3	One of the constants to specify the different parts of the scroller. Down or right by a large amount.
<code>NSScrollerKnob</code>	2	One of the constants to specify the different parts of the scroller. Directly to the <code>NSScroller</code> 's value, as given by <code>floatValue</code> .
<code>NSScrollerKnobSlot</code>	6	One of the constants to specify the different parts of the scroller. Directly to the <code>NSScroller</code> 's value, as given by <code>floatValue</code> .
<code>NSScrollerNoPart</code>	0	One of the constants to specify the different parts of the scroller. Don't scroll at all.
<code>NSSmallControlSize</code>	1	One of the values for the <code>ControlSize</code> property. This constant is for controls that cannot be resized in one direction, such as push buttons, radio buttons, checkboxes, sliders, scroll bars, pop-up buttons, tabs, and progress indicators. You should use a small system font with a small control.

Knob style constants

Constant	Value	Description
NSScrollerKnobStyleDark	1	Specifies a dark knob. This style is particularly good against a light background. Available in Mac OS X v10.7 and later.
NSScrollerKnobStyleDefault	0	Specifies a dark knob with a light border. This is the default style; it is good against any background. Available in Mac OS X v10.7 and later.
NSScrollerKnobStyleLight	2	Specifies a light knob. This style is particularly good against a dark background. Available in Mac OS X v10.7 and later.

Scroller Style Constants

Constant	Value	Description
NSScrollerStyleLegacy	0	Specifies legacy-style scrollers as prior to Mac OS X v10.7. Available in Mac OS X v10.7 and later.
NSScrollerStyleOverlay	1	Specifies overlay-style scrollers in Mac OS X v10.7 and later. Available in Mac OS X v10.7 and later.

9.54 class NSScrollViewMBS

9.54.1 class NSScrollViewMBS

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The Cocoa class for a view which has scrollbars.

Notes: Embed another view inside it to add scrolling.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard.

Subclass of the NSViewMBS class.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.2](#)
- [Smooth scrolling list control with ContainerControls](#)
- [Use Text Finder for TextArea in Xojo](#)
- [MBS Xojo Plugins, version 18.5pr3](#)
- [Line Wrap for Textarea in Xojo Mac applications](#)
- [MBS Xojo Plugins, version 17.1pr2](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr7](#)
- [MBS Xojo / Real Studio Plugins, version 14.3pr7](#)
- [Tip of the day: Autohide scrollbar](#)
- [Custom Scrollbars for Real Studio Cocoa Apps](#)

Xojo Developer Magazine

- [19.4, page 10: News](#)
- [14.4, page 24: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.54.2 Methods

9.54.3 Constructor

Plugin Version: 8.2, Platform: macOS, Targets: Desktop only.

Function: Creates a new scrollview with size 100/100 and position 0/0

Example:

9.54. CLASS NSSCROLLVIEWMBS 603

`dim t as new NSScrollViewMBS`

Notes: On success the handle property is not zero.
See also:

- 9.54.4 Constructor(Handle as Integer) 603
- 9.54.5 Constructor(left as Double, top as Double, width as Double, height as Double) 603

9.54.4 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSScrollView handle.

Example:

```
dim t as new NSScrollViewMBS(0, 0, 100, 100)
dim v as new NSScrollViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSScrollView and the plugin retains this handle.
See also:

- 9.54.3 Constructor 602
- 9.54.5 Constructor(left as Double, top as Double, width as Double, height as Double) 603

9.54.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Creates a new scrollview with the given size and position.

Example:

```
dim x as new NSScrollViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.
See also:

- 9.54.3 Constructor 602
- 9.54.4 Constructor(Handle as Integer) 603

9.54.6 flashScrollers

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Flash the overlay scroll bars.

Notes: This method is only to scroll views that use overlay scrollers.

This method can be invoked to cause the overlay scroller knobs to be momentarily shown. This may be desirable when changing a document view's size or swapping new content into the view, or to give the user a sense of the current position within the scrollable range at each step of an incremental search or similar operation.

Available in Mac OS X v10.7 and later.

9.54.7 reflectScrolledClipView(`clipView as NSClipViewMBS`)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Adjusts the receiver's scrollers to reflect the size and positioning of its content view.

Notes: `ClipView`: The clip view being adjusted to. If a `ClipView` is any view object other than the receiver's content view, the method does nothing.

This method is invoked automatically during scrolling and when an `NSClipView` object's relationship to its document view changes; you should rarely need to invoke it yourself, but may wish to override it for custom updating or other behavior. If you override this method, be sure to call the superclass implementation. If you do not, other controls (such as the current scrollers) may not be updated properly.

Available in Mac OS X v10.0 and later.

9.54.8 tile

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Lays out the components of the receiver: the content view, the scrollers, and the ruler views.

Notes: You rarely need to invoke this method, but subclasses may override it to manage additional components.

9.54.9 Properties

9.54.10 autohidesScrollers as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: True when autohiding is set for scroll bars in the scrollview.

Notes: (Read and Write property)

9.54.11 backgroundColor as NSColorMBS

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The background color.

Notes: (Read and Write property)

9.54.12 borderType as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The border type.

Notes: Specifies the appearance of the style of the scroll view's border. See `NSBorderType` for a list of possible values.

Use constants `NSNoBorder = 0`, `NSLineBorder = 1`, `NSBezelBorder = 2` or `NSGrooveBorder = 3`.
(Read and Write property)

9.54.13 contentSize as NSSizeMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the size of the receiver's content view.

Notes: (Read only property)

9.54.14 contentView as NSClipViewMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The content view.

Notes: If view has a document view, this method also sets the receiver's document view to be the document view of view. The original content view retains its document view.

(Read and Write property)

9.54.15 documentCursor as Variant

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The cursor used when the cursor is over the content view.

Notes: Value is an NSCursorMBS object.

(Read and Write property)

9.54.16 documentView as NSViewMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The view the receiver scrolls within its content view.

Notes: (Read and Write property)

9.54.17 documentVisibleRect as NSRectMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the portion of the document view, in its own coordinate system, visible through the receiver's content view.

Notes: (Read only property)

9.54.18 drawsBackground as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the scrollview draws its background.

Notes: (Read and Write property)

9.54.19 FindBarPosition as Integer

Plugin Version: 18.5, Platform: macOS, Targets: Desktop only.

Function: The position of the find bar.

Notes: See NSScrollViewFindBarPosition* constants for possible values.

(Read and Write property)

9.54.20 FindBarView as NSViewMBS

Plugin Version: 18.5, Platform: macOS, Targets: Desktop only.

Function: The view assigned by the text bar as the find bar view for the container.

Notes: This property is managed by NSTextFinderMBS and you must not set this property. The container may freely modify the view's width, but should not modify its height. (Read only property)

9.54.21 FindBarVisible as Boolean

Plugin Version: 18.5, Platform: macOS, Targets: Desktop only.

Function: Whether the container should display its find bar.

Notes: When this property is true and the findBarView property is set, then the find bar is displayed by the container. Otherwise, the find bar is not displayed.

The default value should be false.

(Read and Write property)

9.54.22 hasHorizontalRuler as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: True if the receiver maintains a horizontal ruler view, false if it doesn't.

Notes: (Read and Write property)

9.54.23 hasHorizontalScroller as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver keeps a horizontal scroller

Notes: (Read and Write property)

9.54.24 hasVerticalRuler as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Determines whether the scrollview keeps a vertical ruler object.

Notes: (Read and Write property)

9.54.25 `hasVerticalScroller` as `boolean`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: True if the scrollview displays a vertical scroller, false if it doesn't.

Notes: (Read and Write property)

9.54.26 `horizontalLineScroll` as `Double`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The amount by which the receiver scrolls itself horizontally when scrolling line by line to aFloat, expressed in the content view's coordinate system.

Notes: This amount is the amount used when the user clicks the scroll arrows on the horizontal scroll bar without holding down a modifier key. When displaying text in an NSScrollView, for example, you might set this amount to the height of a single line of text in the default font.

(Read and Write property)

9.54.27 `horizontalPageScroll` as `Double`

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The amount of the document view kept visible when scrolling horizontally page by page, expressed in the content view's coordinate system.

Notes: This amount is used when the user clicks the scroll arrows on the horizontal scroll bar while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page.

(Read and Write property)

9.54.28 `horizontalScrollElasticity` as `Integer`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The scroll view's horizontal elasticity mode.

Notes: A scroll view can scroll its contents past its bounds to achieve an elastic effect.

When set to `NSScrollElasticityAutomatic`, scrolling the horizontal axis beyond its document bounds only occurs if the document width is greater than the view width or, the vertical scroller is hidden and the hori-

zontal scroller is visible.

The default value is NSScrollElasticityAutomatic.

Available in Mac OS X v10.7 and later.
(Read and Write property)

9.54.29 horizontalScroller as NSScrollerMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The receiver's horizontal scroller, regardless of whether the receiver is currently displaying it, or nil if the receiver has none.

Notes: (Read and Write property)

9.54.30 lineScroll as Double

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The horizontal and vertical line scroll amounts to aFloat.

Notes: The line scroll is the amount by which the receiver scrolls itself when scrolling line by line, expressed in the content view's coordinate system. It's used when the user clicks the scroll arrows without holding down a modifier key. When displaying text in an NSScrollView, for example, you might set this value to the height of a single line of text in the default font.

As part of its implementation, this method sets VerticalLineScroll and HorizontalLineScroll.
(Read and Write property)

9.54.31 pageScroll as Double

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The vertical page scroll amount: the amount of the document view kept visible when scrolling vertically page by page, expressed in the content view's coordinate system.

Notes: This amount is used when the user clicks the scroll arrows on the vertical scroll bar while holding down the Option key. As part of its implementation, this method calls verticalPageScroll.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document

view is scrolled by one page.

Note that a scroll view can have two different page scroll amounts: `verticalPageScroll` and `horizontalPageScroll`. Use this method only if you can be sure they're both the same.
(Read and Write property)

9.54.32 `rulersVisible` as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: Whether rulers should be visible.

Notes: (Read and Write property)

9.54.33 `scrollerKnobStyle` as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The knob style of scroll views that use the overlay scroller style.

Notes: Applicable only to scroll views that use overlay scrollers.

Available in Mac OS X v10.7 and later.

(Read and Write property)

9.54.34 `scrollerStyle` as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The scroller style used by the scroll view.

Notes: This setting is automatically set at runtime, based on the user's preference setting and, if relevant, the set of connected pointing devices and their configured scroll capabilities, as determined by the `NSScroller` method `preferredScrollerStyle`.

Setting an scroll view's scroller style sets the style of both the horizontal and vertical scrollers. If the scroll view subsequently creates or is assigned a new horizontal or vertical scroller, they will be assigned the same scroller style that was assigned to the scroll view..

Available in Mac OS X v10.7 and later.

(Read and Write property)

9.54.35 scrollsDynamically as boolean

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: True if the scrollview redraws its document view while tracking the knob, false if it redraws only when the scroller knob is released.

Notes: NSScrollView scrolls dynamically by default.
(Read and Write property)

9.54.36 usesPredominantAxisScrolling as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Whether the scroll view uses a predominant scrolling axis for content.

Notes: Whether the scroll view supports a predominant scrolling direction. true if there is a predominant scrolling direction; otherwise false.

Some content is scrollable in both the horizontal and vertical axes, but is predominantly scrolled one axis at a time. Other content (such as a drawing canvas) should scroll freely in both axes.

Traditionally this is not an issue with scroll wheels since they can only scroll in one direction at a time. With scroll balls and touch surfaces, it becomes more difficult to determine the user's intention.

This property helps a scroll view determine the user's intention by specifying if there is a predominant scrolling axis for content.

The default value is true.

Available in Mac OS X v10.7 and later.

(Read and Write property)

9.54.37 verticalLineScroll as Double

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The amount by which the view scrolls itself vertically when scrolling line by line to aFloat, expressed in the content view's coordinate system.

Notes: This value is the amount used when the user clicks the scroll arrows on the vertical scroll bar without holding down a modifier key. When displaying text in an NSScrollView, for example, you might set this value to the height of a single line of text in the default font.

(Read and Write property)

9.54.38 verticalPageScroll as Double

Plugin Version: 7.8, Platform: macOS, Targets: Desktop only.

Function: The amount of the document view kept visible when scrolling vertically page by page to aFloat, expressed in the content view's coordinate system.

Notes: This amount is used when the user clicks the scroll arrows on the vertical scroll bar while holding down the Option key.

This amount expresses the context that remains when the receiver scrolls by one page, allowing the user to orient to the new display. It differs from the line scroll amount, which indicates how far the document view moves. The page scroll amount is the amount common to the content view before and after the document view is scrolled by one page. Thus, setting the page scroll amount to 0.0 implies that the entire visible portion of the document view is replaced when a page scroll occurs.

(Read and Write property)

9.54.39 verticalScrollElasticity as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The scroll view's vertical elasticity mode.

Notes: A scroll view can scroll its contents past its bounds to achieve an elastic effect.

When set to NSScrollElasticityAutomatic, scrolling the vertical axis beyond its document bounds occurs if any of the following are true: the vertical scroller is visible, the content height is greater than view height, or the horizontal scroller hidden.

Available in Mac OS X v10.7 and later.

(Read and Write property)

9.54.40 verticalScroller as NSScrollerMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The receiver's vertical scroller, regardless of whether the receiver is currently displaying it, or nil if the receiver has none.

Notes: (Read and Write property)

9.54.41 Constants

Elasticity behavior Constants

Constant	Value	Description
NSScrollElasticityAllowed	2	Allow content to be scrolled past its bounds on this axis in an elastic fashion. Available in Mac OS X v10.7 and later.
NSScrollElasticityAutomatic	0	Automatically determine whether to allow elasticity on this axis. Available in Mac OS X v10.7 and later.
NSScrollElasticityNone	1	Disallow scrolling beyond document bounds on this axis. Available in Mac OS X v10.7 and later.

Findbar position constants

Constant	Value	Description
NSScrollViewFindBarPositionAboveContent	1	The find bar is displayed above the scroll view content. Available in Mac OS X v10.7 and later.
NSScrollViewFindBarPositionAboveHorizontalRuler	0	The find bar is displayed above the horizontal ruler, if visible. Available in Mac OS X v10.7 and later.
NSScrollViewFindBarPositionBelowContent	2	The find bar is displayed below the scroll view content. Available in Mac OS X v10.7 and later.

9.55 class NSSliderMBS

9.55.1 class NSSliderMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: An NSSlider object displays a range of values for something in the application.

Notes: Sliders can be vertical or horizontal bars or circular dials. An indicator, or knob, notes the current setting. The user can move the knob in the slider's bar—or rotate the knob in a circular slider—to change the setting.

The NSSlider class uses the NSSliderCell class to implement its user interface.

Subclass of the NSControlMBS class.

Blog Entries

- [MonkeyBread Software Releases the MBS REALbasic plug-ins 8.4](#)

9.55.2 Methods

9.55.3 acceptsFirstMouse(event as NSEventMBS) as boolean

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Overridden by subclasses to return true if the receiver should be sent a mouseDown event for an initial mouse-down event, false if not.

Notes: The receiver can either return a value unconditionally or use the location of event *e* to determine whether or not it wants the event. The default implementation ignores the event and returns false.

Implement this event in a subclass to allow instances to respond to click-through. This allows the user to click on a view in an inactive window, activating the view with one click, instead of clicking first to make the window active and then clicking the view. Most view objects refuse a click-through attempt, so the event simply activates the window. Many control objects, however, such as instances of NSButton and NSSlider, do accept them, so the user can immediately manipulate the control without having to release the mouse button.

9.55.4 closestTickMarkValueToValue(value as Double) as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the value of the tick mark closest to the specified value.

9.55.5 Constructor

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new slider with size 100/100 and position 0/0

Example:

```
dim t as new NSSliderMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.55.6 Constructor(Handle as Integer) 615
- 9.55.7 Constructor(left as Double, top as Double, width as Double, height as Double) 615

9.55.6 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSSlider handle.

Example:

```
dim t as new NSSliderMBS(0, 0, 100, 100)
```

```
dim v as new NSSliderMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSSlider and the plugin retains this handle.

See also:

- 9.55.5 Constructor 615
- 9.55.7 Constructor(left as Double, top as Double, width as Double, height as Double) 615

9.55.7 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new slider with the given size and position.

Example:

```
dim x as new NSSliderMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.55.5 Constructor 615
- 9.55.6 Constructor(Handle as Integer) 615

9.55.8 `indexOfTickMarkAtPoint(p as NSPointMBS) as Integer`

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the tick mark closest to the location of the receiver represented by the given point.

Notes: Returns the index of the tick mark closest to the location specified by point. If point is not within the bounding rectangle (plus an extra pixel of space) of any tick mark, the method returns `NSNotFound`.

In its implementation of this method, the receiving `NSSlider` instance simply invokes the method of the same name of its `NSSliderCell` instance. This method invokes `rectOfTickMarkAtIndex:` for each tick mark on the slider until it finds a tick mark containing the point.

See also:

- 9.55.9 `indexOfTickMarkAtPoint(x as Double, y as Double) as Integer` 616

9.55.9 `indexOfTickMarkAtPoint(x as Double, y as Double) as Integer`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the tick mark closest to the location of the receiver represented by the given point.

Notes: The index of the tick mark closest to the location specified by point. If point is not within the bounding rectangle (plus an extra pixel of space) of any tick mark, the method returns `NSNotFound` (`&h7ffffff`).

See also:

- 9.55.8 `indexOfTickMarkAtPoint(p as NSPointMBS) as Integer` 616

9.55.10 `rectOfTickMarkAtIndex(index as Integer) as NSRectMBS`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the bounding rectangle of the tick mark at the given index.

Notes: If no tick mark is associated with index, the method raises `NSRangeException`.

9.55.11 `tickMarkValueAtIndex(index as Integer) as Double`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the receiver's value represented by the tick mark at the specified index.

9.55.12 Properties

9.55.13 allowsTickMarkValuesOnly as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the receiver fixes its values to those values represented by its tick marks.

Notes: True if the slider fixes its values to the values represented by its tick marks; otherwise, false.
(Read and Write property)

9.55.14 altIncrementValue as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The amount by which the receiver changes its value when the user Option-drag the slider knob.

Notes: The amount by which the value changes when the user drags the slider knob with the Option key held down. Unless you assign a value to AltIncrementValue, altIncrementValue returns -1.0, and the receiver behaves no differently with the Option key down than with it up.

(Read and Write property)

9.55.15 image as NSImageMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: This method has been deprecated. Returns nil.

Notes: The slider may scale and distort barImage to fit inside the bar.

The knob may cover part of the image. If you want the image to be visible all the time, you're better off placing it near the slider.

This method has been deprecated by Apple.

(Read and Write property)

9.55.16 isVertical as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns an integer indicating the orientation (horizontal or vertical) of the slider.

Notes: 1 if the receiver is vertical, 0 if it's horizontal, and -1 if the orientation can't be determined (for example, if the slider hasn't been displayed yet). A slider is defined as vertical if its height is greater than its width.

Settable with macOS 10.12.

(Read and Write property)

9.55.17 knobThickness as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The knob's thickness, in pixels.

Notes: The thickness of the slider knob. The thickness is defined to be the extent of the knob along the long dimension of the bar. In a vertical slider, then, a knob's thickness is its height; in a horizontal slider, a knob's thickness is its width.

(Read and Write property)

9.55.18 maxValue as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The maximum value the receiver can send to its target.

Notes: The slider's maximum value. A horizontal slider sends its maximum value when the knob is at the right end of the bar; a vertical slider sends it when the knob is at the top.

(Read and Write property)

9.55.19 minValue as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: the minimum value the receiver can send to its target.

Notes: The slider's minimum value. A vertical slider sends its minimum value when its knob is at the bottom; a horizontal slider, when its knob is all the way to the left.

(Read and Write property)

9.55.20 numberOfTickMarks as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The number of tick marks associated with the receiver.

Notes: The number of the slider's tick marks. The tick marks assigned to the minimum and maximum values

are included.
(Read and Write property)

9.55.21 `sliderType` as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The slider type.
Notes: Either `NSLinearSlider` or `NSCircularSlider`.
(Read and Write property)

9.55.22 `tickMarkPosition` as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: How the receiver's tick marks are aligned with it.
Notes: A constant indicating the position of the tick marks. Possible values are `NSTickMarkBelow`, `NSTickMarkAbove`, `NSTickMarkLeft`, and `NSTickMarkRight` (the last two are for vertical sliders). The default alignments are `NSTickMarkBelow` and `NSTickMarkLeft`.
(Read and Write property)

9.55.23 `title` as string

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The receiver's title.
Notes: The default title is the empty string.
(Read and Write property)

9.55.24 `titleCell` as `NSCellMBS`

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Sets the cell used to draw the receiver's title.
Notes: You only need to invoke this method if the default title cell, `NSTextFieldCell`, doesn't suit your needs—that is, you want to display the title in a manner that `NSTextFieldCell` doesn't permit. When you do choose to override the default, `titleCell` should be an instance of a subclass of `NSTextFieldCell`.

This method has been deprecated by Apple.
(Read and Write property)

9.55.25 titleColor as NSColorMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Sets the color used to draw the receiver's title.

Notes: This method has been deprecated by Apple.

(Read and Write property)

9.55.26 titleFont as NSFontMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Sets the font used to draw the receiver's title.

Notes: This method has been deprecated by Apple.

(Read and Write property)

9.55.27 trackFillColor as NSColorMBS

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The color of the filled portion of the slider track, in appearances that support it.

Notes: Requires macOS 10.12.2 or newer.

(Read and Write property)

9.55.28 Constants

Constants

Constant	Value	Description
NSCircularSlider	1	One of the slider type constants. A circular slider; that is, a dial.
NSLinearSlider	0	One of the slider type constants. A bar-shaped slider.
NSTickMarkAbove	1	One of the values for the tickMarkPosition property. Tick marks above (for horizontal sliders).
NSTickMarkBelow	0	One of the values for the tickMarkPosition property. Tick marks below (for horizontal sliders); the default for horizontal sliders.
NSTickMarkLeft	1	One of the values for the tickMarkPosition property. Tick marks to the left (for vertical sliders); the default. for vertical sliders
NSTickMarkRight	0	One of the values for the tickMarkPosition property. Tick marks to the right (for vertical sliders).

9.56 class NSStepperMBS

9.56.1 class NSStepperMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: A stepper consists of two small arrows that can increment and decrement a value that appears beside it, such as a date or time.

Notes: The illustration below shows a stepper to the right of a text field, which would show the stepper's value.

Subclass of the NSControlMBS class.

9.56.2 Methods

9.56.3 Constructor

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new stepper with size 100/100 and position 0/0

Example:

```
dim t as new NSStepperMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.56.4 Constructor(Handle as Integer) 621
- 9.56.5 Constructor(left as Double, top as Double, width as Double, height as Double) 622

9.56.4 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSStepper handle.

Example:

```
dim t as new NSStepperMBS(0, 0, 100, 100)
dim v as new NSStepperMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSSStepper and the plugin retains this handle.

See also:

- 9.56.3 Constructor 621
- 9.56.5 Constructor(left as Double, top as Double, width as Double, height as Double) 622

9.56.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a new stepper with the given size and position.

Example:

```
dim x as new NSSStepperMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.56.3 Constructor 621
- 9.56.4 Constructor(Handle as Integer) 621

9.56.6 Properties

9.56.7 autorepeat as boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating how the receiver responds to mouse events.

Notes: True if the first mouse down does one increment (or decrement) and, after a delay of 0.5 seconds, increments (or decrements) at a rate of ten times per second. False if the receiver does one increment (decrement) on a mouse up. The default is true.

(Read and Write computed property)

9.56.8 increment as Double

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The amount by which the receiver will change per increment (decrement).

Notes: (Read and Write computed property)

9.56.9 maxValue as Double

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The maximum value.

Notes: (Read and Write computed property)

9.56.10 minValue as Double

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The minimum value.

Notes: (Read and Write computed property)

9.56.11 valueWraps as boolean

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver wraps around the minimum and maximum values.

Notes: If true, then when incrementing or decrementing, the value wraps around to the minimum or maximum. If valueWraps is false, the value stays pinned at the minimum or maximum.

For example for an angle where 359 increases to 0.

(Read and Write computed property)

9.57 control NSSwitchControlMBS

9.57.1 control NSSwitchControlMBS

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The control to host a NSSwitch as control in Xojo.

Notes: Available on macOS 10.15 or newer.

Blog Entries

- [News from the MBS Xojo Plugins Version 22.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 22.1](#)
- [MBS Xojo Plugins, version 22.1pr3](#)
- [Adding NSSwitch to Xojo](#)
- [MBS Xojo Plugins, version 22.1pr2](#)

Xojo Developer Magazine

- [20.4, page 10: News](#)
- [20.3, page 10: News](#)

9.57.2 Properties

9.57.3 State as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The current state.

Notes: Set to true for toggle on and set to false for toggle off.
(Read and Write property)

9.57.4 View as NSSwitchMBS

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.
(Read only property)

9.57.5 Events

9.57.6 Action

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The action event.

9.57.7 BoundsChanged

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.57.8 Close

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.57.9 ConstructContextMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.57.10 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: Called when a menuitem is choosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.57.11 `didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)`

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the `willShowContextualMenu` event.

9.57.12 `EnableMenuItems`

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named `MenuBarSelected`.

9.57.13 `FrameChanged`

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.57.14 `GotFocus`

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named `FocusReceived`.

Notes:

This only fires if the control itself got focus and not a sub control.

9.57.15 `LostFocus`

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.57.16MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ãs region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.57.17 MouseDrag(x as Integer, y as Integer)

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.57.18 MouseUp(x As Integer, y As Integer)

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.57.19 Open

Plugin Version: 22.1, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.
In Xojo version 2021r3 and newer this event is named `Opening`.

9.57.20 ScaleFactorChanged(NewFactor as double)

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.57.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.58 class NSSwitchMBS

9.58.1 class NSSwitchMBS

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The switch control class for Xojo.

Notes: The NSSwitch class provides a simple interface for displaying and toggling a Boolean state, such as on/off. A switch toggles its state and sends its action when clicked, activated through the keyboard, or tapped in the Touch Bar. NSSwitch also allows dragging between states, and if continuous is true, the switch sends its action for each change in position during the drag.

Use a switch to toggle significant preferences, or preferences that provide access to other controls. Avoid creating lists or tables of switches; instead, for general-purpose toggles, use an instance of NSButtonMBS to display a checkbox.

NSSwitch doesn't use an instance of NSCellMBS to provide its functionality. The cellClass class property and cell instance property both return nil, and they ignore attempts to set a non-nil value.

Available on macOS 10.15 or newer.

Subclass of the NSControlMBS class.

Blog Entries

- [News from the MBS Xojo Plugins Version 22.1](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 22.1](#)
- [Adding NSSwitch to Xojo](#)
- [MBS Xojo Plugins, version 22.1pr2](#)

Xojo Developer Magazine

- [20.4, page 10: News](#)
- [20.3, page 10: News](#)

9.58.2 Methods

9.58.3 Constructor

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with size 100/100 and position 0/0

Example:

`dim t as new NSSwitchMBS`

Notes: On success the handle property is not zero.
See also:

- 9.58.4 Constructor(Handle as Integer) 630
- 9.58.5 Constructor(left as double, top as double, width as double, height as double) 630

9.58.4 Constructor(Handle as Integer)

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with size 100/100 and position 0/0

Example:

`dim t as new NSSwitchMBS`

Notes: On success the handle property is not zero.
See also:

- 9.58.3 Constructor 629
- 9.58.5 Constructor(left as double, top as double, width as double, height as double) 630

9.58.5 Constructor(left as double, top as double, width as double, height as double)

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new button with the given size and position.

Example:

`dim x as new NSSwitchMBS(0, 0, 100, 100)`

Notes: On success the handle property is not zero.
See also:

- 9.58.3 Constructor 629
- 9.58.4 Constructor(Handle as Integer) 630

9.58.6 Properties

9.58.7 state as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: Desktop only.

Function: The current state.

Notes: Set to true for toggle on and set to false for toggle off.
(Read and Write property)

9.59 class NSTableColumnMBS

9.59.1 class NSTableColumnMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: An NSTableColumn stores the display characteristics and attribute identifier for a column in an NSTableView.

Notes: The NSTableColumn determines the width and width limits, resizable, and editability of its column in the NSTableView. It also stores two NSCell objects: the header cell, which is used to draw the column header, and the data cell, used to draw the values for each row. You can control the display of the column by setting the subclasses of NSCell used and by setting the font and other display characteristics for these NSCells. For example, you can use the default NSTextFieldCell for displaying string values or substitute an NSImageCell to display pictures.

Blog Entries

- [MBS Real Studio Plugins, version 12.0pr5](#)

Xojo Developer Magazine

- [14.4, pages 27 to 28: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.59.2 Methods

9.59.3 Constructor(identifier as string)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Initializes a NSTableColumn with identifier as its identifier and with an NSTextFieldCell as its data cell.

9.59.4 dataCellForRow(row as Integer) as NSCellMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the NSCell object used by the NSTableView to draw values for the receiver.

Notes: NSTableView always calls this method. By default, this method just calls dataCell. Subclassers can override if they need to potentially use different cells for different rows. Subclasses should expect this method to be invoked with row equal to -1 in cases where no actual row is involved but the table view needs to get some generic cell info.

9.59.5 sizeToFit

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Resizes the receiver to fit the width of its header cell.

Notes: If the maximum width is less than the width of the header, the maximum is increased to the header's width. Similarly, if the minimum width is greater than the width of the header, the minimum is reduced to the header's width. Marks the NSTableView as needing display if the width actually changes.

9.59.6 Properties

9.59.7 dataCell as NSCellMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The NSCell object used by the NSTableView to draw values for the receiver.

Notes: You can use this property to control the font, alignment, and other text attributes for an NSTableColumn. You can also assign a cell to display things other than text—for example, an NSImageCell to display images.

(Read and Write property)

9.59.8 Editable as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Controls whether the user can edit cells in the receiver by double-clicking them.

Notes: If value is true a double click initiates editing; if flag is false it merely sends the double-click event to the NSTableView. You can initiate editing programmatically regardless of this setting with NSTableView's editColumn() method.

(Read and Write property)

9.59.9 headerCell as NSTableHeaderCellMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The NSCell used to draw the receiver's header to aCell.

Notes: (Read and Write property)

9.59.10 headerToolTip as string

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The tooltip string that is displayed when the cursor pauses over the header cell of the receiver.

Notes: (Read and Write property)

9.59.11 Hidden as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Whether the column is hidden.

Notes: (Read and Write property)

9.59.12 identifier as string

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The object used by the data source to identify the attribute corresponding to the receiver.

Notes: (Read and Write property)

9.59.13 maxWidth as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The maximum width.

Notes: Setting this value will also adjusting the current width if it's greater than this value.
(Read and Write property)

9.59.14 minWidth as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The minimum width.

Notes: Setting this value will also adjusting the current width if it's less than this value.
(Read and Write property)

9.59.15 Resizable as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Whether this column can be resized.

Notes: (Read and Write property)

9.59.16 resizingMask as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Defines the resizing mode for this column.

Notes: Use kNoResizing, kAutoresizingMask and kUserResizingMask constants.
(Read and Write property)

9.59.17 sortDescriptorPrototype as NSSortDescriptorMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The table column's sort descriptor prototype.

Notes: A table column is considered sortable if it has a sort descriptor that specifies the sorting direction, a key to sort by, and a selector that defines how to sort.
(Read and Write property)

9.59.18 tableView as NSTableViewMBS

Plugin Version: 12.0, Platform: macOS, Targets: Desktop only.

Function: The owner tableview.

Notes: (Read and Write property)

9.59.19 title as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The title of the table column's header.

Notes: (Read and Write property)

9.59.20 width as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The width of the column.

Notes: (Read and Write property)

9.59.21 Constants

Constants

Constant	Value	Description
<code>NSTableColumnAutore-sizingMask</code>	1	One of the constants specify the resizing modes available for the table column. Allows the table column to resize automatically in response to resizing the tableview. Enabling this option is the same as enabling the "Live Resizable" option in Interface Builder. The resizing behavior for the table view is set using the <code>NSTableView</code> method <code>setColumnAutore-sizingStyle:</code> . Available in Mac OS X v10.4 and later.
<code>NSTableColumnNoResizing</code>	0	One of the constants specify the resizing modes available for the table column. Prevents the table column from resizing. Available in Mac OS X v10.4 and later.
<code>NSTableColumnUserResizingMask</code>	2	One of the constants specify the resizing modes available for the table column. Allows the table column to be resized explicitly by the user. Enabling this option is the same as enabling the "User Resizable" option in Interface Builder. Available in Mac OS X v10.4 and later.

9.60 control NSTableControlMBS

9.60.1 control NSTableControlMBS

Plugin Version: 15.2, Platform: macOS, Targets: Desktop only.

Function: The control for a NSTableView.

Notes: Please use NSOutlineControlMBS for hierarchical lists and NSTableControlMBS for normal lists.

Blog Entries

- [MBS Xojo Plugins, version 22.5pr3](#)
- [Xojo 2022r3 released](#)
- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 17.1](#)
- [MBS Xojo Plugins, version 17.1pr5](#)
- [MBS Xojo Plugins, version 17.1pr3](#)
- [MBS Xojo Plugins, version 17.1pr2](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr2](#)

Videos

- [Presentation from London conference about MBS Plugins.](#)

Xojo Developer Magazine

- [15.3, page 10: News](#)
- [14.4, page 22: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.60.2 Properties

9.60.3 AcceptTabs as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Whether the control should accept tab keys.

Notes: If true, the plugin will not forward the tab keydown/keyup events to Xojo, because Xojo would do switch to next control.

(Read and Write property)

9.60.4 allowsColumnReordering as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to rearrange columns by dragging their headers.

Notes: The default value of this property is true, which allows the user to rearrange the table view's columns. You can rearrange columns programmatically regardless of this setting.
(Read and Write property)

9.60.5 allowsColumnResizing as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to resize columns by dragging between their headers.

Notes: The default of this property is true, which allows the user to resize the table view's columns. You can resize columns programmatically regardless of this setting.
(Read and Write property)

9.60.6 allowsColumnSelection as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select columns by clicking their headers.

Notes: The default is false, which prevents the user from selecting columns (if you create the table view in Interface Builder, the default value is true). You can select columns programmatically regardless of this setting.
(Read and Write property)

9.60.7 allowsEmptySelection as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select zero columns or rows.

Notes: The default is true, which allows the user to select zero columns or rows.
(Read and Write property)

9.60.8 allowsMultipleSelection as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view allows the user to select more than one column or row at a time.

Notes: The default is false, which allows the user to select only one column or row at a time. You can select multiple columns or rows programmatically regardless of this setting.

(Read and Write property)

9.60.9 autohidesScrollers as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view automatically hides its scroll bars when they are not needed.

Notes: The horizontal and vertical scroll bars are hidden independently of each other. When the value of this property is YES and the content of the scroll view doesn't extend beyond the size of the clip view on a given axis, the scroller on that axis is removed to leave more room for the content.

(Read and Write property)

9.60.10 disableCellEvents as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Whether to disable cell based events.

Notes: The table view can work with cell modes and use NSCell to disable cells.

Or since OS X 10.7 it can work with NSView to display cells or rows.

This property lets you explicitly disable cells and use only views.

(Read and Write property)

9.60.11 disableViewEvents as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Whether to disable view based events.

Notes: The table view can work with cell modes and use NSCell to disable cells.

Or since OS X 10.7 it can work with NSView to display cells or rows.

This property lets you explicitly disable views and use only cells.

(Read and Write property)

9.60.12 hasHorizontalScroller as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a horizontal scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a horizontal scroller as needed. The default value of this property is false.

(Read and Write property)

9.60.13 hasVerticalScroller as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean that indicates whether the scroll view has a vertical scroller.

Notes: When the value of this property is true, the scroll view allocates and displays a vertical scroller as needed. The default value of this property is false.

(Read and Write property)

9.60.14 ScrollView as NSScrollViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The scroll view used in this control.

Notes: (Read only property)

9.60.15 View as NSTableViewMBS

Plugin Version: 15.2, Platform: macOS, Targets: Desktop only.

Function: The table view used in this control.

Notes: (Read only property)

See also:

- 9.60.82 view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS

659

9.60.16 Events**9.60.17 acceptDrop(info as NSDraggingInfoMBS, row as Integer, dropOperation as Integer) as boolean**

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called by TableView when the mouse button is released over a table view that previously decided to allow a drop.

Notes: info: An object that contains more information about this dragging operation.

row: The index of the proposed target row.

operation: The type of dragging operation.

Returns true if the drop operation was successful, otherwise false.

The data source should incorporate the data from the dragging pasteboard in the implementation of this method. You can use the draggingPasteboard method to get the data for the drop operation from info.

To accept a drop on the second row, row would be 2 and operation would be NSTableViewDropOn. To accept a drop below the last row, row would be TableView.numberOfRows and operation would be NSTableViewDropAbove.

9.60.18 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.60.19 Close

Plugin Version: 15.2, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.60.20 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: This event informs the delegate that a column was moved by user action in the table view.

9.60.21 ColumnDidResize(notification as NSNotificationMBS, tableColumn as NSTableColumnMBS, OldWidth as Double)

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: This event informs you that a column was resized in the table view.

9.60.22 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.60.23 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuItem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.60.24 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Optional return a different cells for each row.

Notes: A different data cell can be returned for any particular tableColumn and row, or a cell that will be used for the entire row (a full width cell). The returned cell should properly implement copyWithZone:, since the cell may be copied by NSTableView. If the tableColumn is non-nil, and nil is returned, then the table will use the default cell from tableColumn.dataCellForRow(Row).

When each row is being drawn, this method will first be called with a nil tableColumn. At this time, you can return a cell that will be used to draw the entire row, acting like a group. If you do return a cell for the 'nil' tableColumn, be prepared to have the other corresponding datasource and delegate methods to be called with a 'nil' tableColumn value. If don't return a cell, the method will be called once for each tableColumn in the tableView, as usual.

9.60.25 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Tells the delegate that a row view was added at the specified row.

Notes: rowView: The row view.

row: The index of the row.

At this point, the delegate can add extra views, or modify the properties of rowView.

This method is only valid for NSView-based table views.

9.60.26 didClickTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Called if a table column was clicked on.

9.60.27 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.60.28 didDragTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Sent at the time the mouse button goes up in tableView and tableColumn has been dragged during the time the mouse button was down.

Notes: tableColumn: The table column.

The behavior of this method on Mac OS X v10.5 is different from prior versions. On Mac OS X v 10.5 the dragged column is sent to the subclass. In earlier versions the table column that is currently located at the dragged column's original index is sent.

9.60.29 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Tells the delegate that a row view was removed from the table at the specified row.

Notes: rowView: The row view.

row: The index of the row.

If row equals -1, the row is being deleted from the table and is no longer a valid row; otherwise row is a valid row that is being removed by being moved off screen.

This method is only valid for NSView-based table views.

9.60.30 didTile

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The tableview did tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.60.31 DoubleClick

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse made a double click.

9.60.32 draggingSessionEnded(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to determine when a dragging session has ended.

Notes: session: The dragging session.

screenPoint: The ending drag location in screen coordinates.

operation: The drag operation. See NSDragOperation for supported values.

This delegate method can be used to determine when the dragging source operation ended at a specific location, such as the trash, by checking for an operation of NSDragOperationDelete.

9.60.33 draggingSessionWillBegin(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, rowIndexes as NSIndexSetMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to determine when a dragging session will begin.

Notes: session: The dragging session.

screenPoint: The initial drag location in screen coordinates.

`rowIndexes`: The indexes of the rows to be dragged, excluding rows that were not dragged due to `pasteboardItemForRow` returning `nil`.

Implement this method to know when the dragging session is about to begin and to potentially modify the dragging session.

The dragged item order will directly match the pasteboard writer array used to begin the dragging session with the `NSView` method `beginDraggingSessionWithItems`. Hence, the order is deterministic, and can be used in `acceptDrop` when enumerating the `NSDraggingInfo` pasteboard classes.

9.60.34 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named `MenuBarSelected`.

9.60.35 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.60.36 GotFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named `FocusReceived`.

Notes:

This only fires if the control itself got focus and not a sub control.

9.60.37 heightOfRow(row as Int64) as Double

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Implement this event to support a table with varying row heights.

Notes: The height returned by this method should not include intercell spacing and must be greater than zero. Performance Considerations: For large tables in particular, you should make sure that this method is efficient. NSTableView may cache the values this method returns, but this should NOT be depended on, as all values may not be cached. To signal a row height change, call `noteHeightOfRowsWithIndexesChanged`. For a given row, the same row height should always be returned until `noteHeightOfRowsWithIndexesChanged` is called, otherwise unpredictable results will happen. NSTableView automatically invalidates its entire row height cache in `reloadData`, and `noteNumberOfRowsChanged`.

9.60.38 `isGroupRow(row as Int64) as boolean`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to indicate that a specified row is a group row.

Notes: row: The row index.

Return true if the specified row should have the group row style drawn, false otherwise.

If the cell in row is an `NSTextFieldCell` and contains only a string, the group row style attributes will automatically be applied to the cell.

Group rows in view-based table views can be made to visually ‚float‘ by setting the tableview method `setFloatsGroupRows` to true.

Note: When configured as a source list style table view, rows identified as group rows draw with a specific style unique for source lists.

Available in Mac OS X v10.5 and later.

9.60.39 `LeftMouseDown(e as NSEventMBS) as boolean`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.40 `LeftMouseDragged(e as NSEventMBS) as boolean`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the left button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.41 LeftMouseUp(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the left mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.42 LostFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.60.43MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.60.44 `mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Sent to the subclass whenever the mouse button is clicked in the table view's header column.

Notes: tableColumn: The table column.

9.60.45 `MouseDownDrag(x as Integer, y as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

9.60.46 `MouseDownUp(x as Integer, y as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.60.47 `namesOfPromisedFilesDroppedAtDestination(dropDestination as folderItem, DraggedRowsWithIndexes as NSIndexSetMBS) as string()`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns an array of filenames that represent the indexSet rows for a drag to dropDestination.

Notes: dropDestination: The drop location where the files are created.

indexSet: The indexes of the items being dragged.

Returns an array of filenames (not full paths) for the created files that the receiver promises to create.

This method is called when a destination has accepted a promise drag.

For more information on file promise dragging, see documentation on the NSDraggingSource protocol and namesOfPromisedFilesDroppedAtDestination:.

9.60.48 nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to modify how type selection works.

Notes: startRow: The starting row of the search range.

endRow: The ending row of the search range.

searchString: A string containing the typed selection.

Return the first row in the range of startRow through endRow (excluding endRow itself) that matches selectionString. Return -1 if no match is found.

It is possible for endRow to be less than startRow if the search will wrap.
Available in Mac OS X v10.5 and later.

9.60.49 numberOfRowsInTableView as Integer

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when the table view needs to know the number of rows.

Notes: numberOfRowsInTableView is called very frequently, so it must be efficient.

9.60.50 objectValue(column as NSTableColumnMBS, row as Integer) as Variant

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a value is required for a given cell.

Notes: Please implement your own arrays to store values.

9.60.51 Open

Plugin Version: 15.2, Platform: macOS, Targets: .

Function:

The control is about to was created and you can initialize it.
In Xojo version 2021r3 and newer this event is named Opening.

9.60.52 OtherMouseDown(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has pressed a mouse button other than the left or right one.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.53 OtherMouseDragged(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with a button other than the left or right button pressed.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.54 OtherMouseUp(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released a mouse button other than the left or right button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.55 pasteboardItemForRow(row as Integer) as NSPasteboardItemMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called to allow the table to support multiple item dragging.

Notes: row: The row.

Returns an instance of NSPasteboardItem. Returning nil excludes the row from being dragged.

This method is required for multi-image dragging.

If this method is implemented, then writeRowsWithIndexes will not be called.

This is pasteboardWriterForRow method.

9.60.56 RightMouseDown(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the view that the user has pressed the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.57 RightMouseDragged(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has moved the mouse with the right button pressed .

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.58 RightMouseUp(e as NSEventMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Informs the receiver that the user has released the right mouse button.

Notes: This event is called before the normal event processing from Xojo happens. So return true to hide event from Xojo runtime.

9.60.59 rowActionsForRow(row as Integer, edge as Integer) as NSTableViewRowActionMBS()

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Asks the control to provide an array of row actions to be attached to the specified edge of a table row and displayed when the user swipes horizontally across the row.

Notes: row: The index of the target row.

edge: The edge (NSTableViewRowActionEdgeLeading or NSTableViewRowActionEdgeTrailing) for which row actions are requested. This is based on the direction in which the user swiped on the row. Swiping to the right results in an edge value of leading. Swiping to the left results in an edge value of trailing.

Returns an array of row actions (of class NSTableViewRowActionMBS) to be enabled on the specified edge of the table row.

Implement this method if your table row supports actions that are displayed when the user swipes horizontally across the row. For example, your table view could use this method to implement a swipe left to delete function in your table rows. When called, this method receives the table view, the index of the row

the user swiped, and an edge of type `NSTableRowActionEdge`. The method should return an array of any row actions of class `NSTableViewRowAction` that are supported for the specified edge. If no row actions are available, an empty array should be returned.

If this method isn't implemented, then the table row displays no actions when the user swipes horizontally away from the specified edge.

9.60.60 `rowViewForRow(row as Integer)` as `NSTableRowViewMBS`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Asks the delegate for a view to display the specified row.

Notes: row: The row index.

Return an instance or subclass of `NSTableRowView`. If nil is returned, an `NSTableRowView` instance will be created and used.

You can implement this event to return a custom `NSTableRowView` for row.

The reuse queue can be used in the same way as documented in `tableView:view:row:`. The returned view will have attributes properly set to it before it's added to the `tableView`.

This method is only valid for `NSView`-based table views.

9.60.61 `ScaleFactorChanged(NewFactor as Double)`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.60.62 `SelectionDidChange(notification as NSNotificationMBS)`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: This event informs you that the table view's selection has changed.

9.60.63 `selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS)` as `NSIndexSetMBS`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to modify the proposed selection.

Notes: proposedSelectionIndexes: An index set containing the indexes of the proposed selection.

Return an `NSIndexSet` instance containing the indexes of the new selection. Return `proposedSelectionIndexes` if the proposed selection is acceptable, or the value of the table view's existing selection to avoid changing the selection.

This method may be called multiple times with one new index added to the existing selection to find out if a particular index can be selected when the user is extending the selection with the keyboard or mouse.

Implementation of this method is optional. If implemented, this method will be called instead of `shouldSelectRow`.

Available in Mac OS X v10.5 and later.

9.60.64 SelectionIsChanging(notification as NSNotificationMBS)

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: This event informs you that the table view's selection is in the process of changing (typically because the user is dragging the mouse across a number of rows).

9.60.65 selectionShouldChangeInTableView as boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Returns whether the selection should change.

Notes: Return true to allow the table view to change its selection (typically a row being edited), false to deny selection change.

The user can select and edit different cells within the same row, but can't select another row unless the delegate approves. The subclass can implement this method for complex validation of edited rows based on the values of any of their cells.

9.60.66 setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a cell value is saved to the datasource.

9.60.67 shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Returns whether the cell at the specified row and column can be edited.

Notes: TableColumn: The table column.

rowIndex: The row index.

Return true to allow editing the cell, false to deny editing.

The subclass can implement this method to disallow editing of specific cells.

Note: This method is only valid for cell-based table views.

9.60.68 shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64) as boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Sent to the subclass to allow or prohibit the specified column to be dragged to a new location.

Notes: columnIndex: The index of the column being dragged.

newColumnIndex: The proposed target index of the column.

Return true if the column reordering should be allowed, otherwise false.

When a column is initially dragged by the user, the delegate is first called with a newColumnIndex value of -1. Returning false will disallow that column from being reordered at all. Returning true allows it to be reordered, and the delegate will be called again when the column reaches a new location.

The actual NSTableColumn instance can be retrieved from the tableColumns array.

If this method is not implemented, all columns are considered reorderable.

Available in Mac OS X v10.6 and later.

9.60.69 shouldSelectRow(row as Int64) as boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Returns whether the table view should allow selection of the specified row.

Notes: rowIndex: The row index.

Return true to permit selection of the row, false to deny selection.

The delegate can implement this method to disallow selection of particular rows. For better performance and finer-grain control over the selection, use `selectionIndexesForProposedSelection`.

9.60.70 `shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Returns whether the specified table column can be selected.

Notes: `TableColumn`: The table column.

Return true to permit selection, otherwise false.

The subclass can implement this event to disallow selection of particular columns.

9.60.71 `shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control cell expansion for a specific row and column.

Notes: `TableColumn`: The table column.

`row`: The row index.

Return true if the tooltip cell should expand, false otherwise.

Cell expansion can occur when the mouse hovers over the specified cell and the cell contents are unable to be fully displayed within the cell. If this method returns true, the full cell contents will be shown in a special floating tool tip view, otherwise the content is truncated.

Note: This method is only valid for cell-based table views.

Available in Mac OS X v10.5 and later.

9.60.72 `shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control the tracking behavior for a specific cell.

Notes: cell: The cell to track.

tableColumn: The table column.

row: A row in tableView.

Returns true if the cell should track, false otherwise.

Normally, only selectable or selected cells can be tracked. If you implement this method, cells which are not selectable or selected can be tracked, and vice-versa.

For example, this allows you to have an NSButtonCell in a table which does not change the selection, but can still be clicked on and tracked.

Note: This method is only valid for cell-based table views.

Available in Mac OS X v10.5 and later.

9.60.73 `shouldTypeSelectForEvent(e as NSEventMBS, searchString as string)` as Boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control type select for a specific event.

Notes: event: The event.

searchString: The search string or nil if no type select has began.

Return true to allow type select for event, false otherwise.

Typically, this is called from the table view keyDown implementation and the event will be a key event.
Available in Mac OS X v10.5 and later.

9.60.74 `sizeToFitWidthOfColumn(column as Int64)` as Double

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to provide custom sizing behavior when a column's resize divider is double clicked.

Notes: column: The index of the column.

Returns the width of the specified column.

By default, NSTableView iterates every row in the table, accesses a cell via preparedCellAtColumn, and requests the cellSize to find the appropriate largest width to use.

For accurate results and performance, it is recommended that this method is implemented when using large tables. By default, large tables use a monte carlo simulation instead of iterating every row.

Available in Mac OS X v10.6 and later.

9.60.75 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called by TableView to indicate that sorting may need to be done.

Notes: The data source typically sorts and reloads the data, and adjusts the selections accordingly. If you need to know the current sort descriptors and the data source doesn't manage them itself, you can get the current sort descriptors by calling TableView.sortDescriptors function.

9.60.76 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow.

9.60.77 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow.

9.60.78 toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Returns a string that is displayed as a tooltip for the specified cell in the column and row.

Notes: Cell: The cell.

r: The proposed active area of the tooltip. You can modify rect to provide an alternative active area.

TableColumn: The table column.

row: The row index.

mouseLocation: The mouse location.

Return a string containing the tooltip. Return empty string if no tooltip is desired.

By default, rect is computed as cell.drawingRectForBounds(cellFrame).

Available in Mac OS X v10.4 and later.

9.60.79 typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to provide an alternate text value used for type selection for a specified row and column.

Notes: tableColumn: The table column.

row: The row index.

Returns a string that is used in type select comparison for row and tableColumn. Return "" if the row or tableColumn should not be searched.

Implement this method to change the string value that is searched for based on what is displayed. By default, all cells with text in them are searched.

If this event is not implemented the string value is the cell string value.

Implementation of this event is optional.

Available in Mac OS X v10.5 and later.

9.60.80 updateDraggingItemsForDrag(draggingInfo as NSDraggingInfoMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Implement this method to allow the table to update dragging items as they are dragged over a view.

Notes: The dragging information.

Required for multi-image dragging. Typically this will involve invoking `enumerateDraggingItemsWithOptions` on the `draggingInfo` parameter value and setting the `draggingItem` object, to an `imageComponentsProvider` to a proper image based on the content.

For view-based table views, you can use the `NSTableCellView` method `draggingImageComponents`. For cell-based tables, use the `NSCell` method `draggingImageComponentsWithFrame`.

9.60.81 `validateDrop(info as NSDraggingInfoMBS, proposedRow as Integer, dropOperation as Integer) as Integer`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Used by a `tableView` to determine a valid drop target.

Notes: `info`: An object that contains more information about this dragging operation.

`row`: The index of the proposed target row.

`operation`: The type of dragging operation proposed.

Returns the dragging operation the data source will perform.

The data source may retarget a drop by calling `setDropRow` and returning something other than `NSDraggingOperationNone`. A data source might retarget for various reasons, such as to provide better visual feedback when inserting into a sorted position.

To propose a drop on the second row, `row` would be 2 and `operation` would be `NSTableViewDropOn`. To propose a drop below the last row, `row` would be `tableView.numberOfRows` and `operation` would be `NSTableViewDropAbove`.

9.60.82 `view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Asks the delegate for a view to display the specified row and column.

Notes: see also

<https://developer.apple.com/reference/appkit/nstableviewdelegate/1527449-tableview?language=objc>

See also:

- 9.60.15 `View as NSTableViewMBS`

640

9.60.83 `willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)`

Plugin Version: 15.2, Platform: macOS, Targets: .

Function: Informs you that the tableview will display the specified cell at the row in the column.

Notes: `Cell`: The cell to be displayed.

TableColumn: The table column.

row: The row index.

The event can modify the display attributes of cell to alter the appearance of the cell.

Because cell is reused for every row in tableColumn, the event must set the display attributes both when drawing special cells and when drawing normal cells.

Note: The implementation of this method must not draw portions of the cell. It should only alter the state of the passed in cell.

9.60.84 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.60.85 willTile

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The tableview will tile.

Notes: The internal tile function properly sizes the table view and its header view and marks it as needing display.

9.60.86 writeRowsWithIndexes(rowIndexes as NSIndexSetMBS, pboard as NSPasteboardMBS) as boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether a drag operation is allowed.

Notes: rowIndexes: An index set of row numbers that will be participating in the drag.

pboard: The pasteboard to which to write the drag data.

Returns true if the drag operation is allowed, false otherwise.

Called by TableView after it has been determined that a drag should begin, but before the drag has been started.

To refuse the drag, return false. To start a drag, return true and place the drag data onto pboard (data, owner, and so on). The drag image and other drag-related information will be set up and provided by the table view once this call returns with true.

9.61 class NSTableDataSourceMBS

9.61.1 class NSTableDataSourceMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The class for a data source of a NSTableView.

Notes: A listbox stores the data itself, but the NSTableView is just a view.

so you need to handle the storage of the table view yourself by subclassing this class and filling the events.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 17.1pr1](#)

Xojo Developer Magazine

- [14.4, page 27: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.61.2 Events

9.61.3 Close

Plugin Version: 8.4, Platform: macOS, Targets: .

Function: The event is called when the datasource is destroyed.

9.61.4 numberOfRowsInTableView as Integer

Plugin Version: 8.4, Platform: macOS, Targets: .

Function: Called when the table view needs to know the number of rows.

Notes: numberOfRowsInTableView is called very frequently, so it must be efficient.

9.61.5 objectValue(column as NSTableColumnMBS, row as Integer) as Variant

Plugin Version: 8.4, Platform: macOS, Targets: .

Function: Called when a value is required for a given cell.

Notes: Please implement your own arrays to store values.

9.61.6 setObjectValue(value as Variant, column as NSTableColumnMBS, row as Integer)

Plugin Version: 8.4, Platform: macOS, Targets: .

Function: Called when a cell value is saved to the datasource.

9.61.7 sortDescriptorsDidChange(oldDescriptors() as NSSortDescriptorMBS)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called by TableView to indicate that sorting may need to be done.

Notes: The data source typically sorts and reloads the data, and adjusts the selections accordingly. If you need to know the current sort descriptors and the data source doesn't manage them itself, you can get the current sort descriptors by calling TableView.sortDescriptors function.

9.62 class NSTableHeaderCellMBS

9.62.1 class NSTableHeaderCellMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: An NSTableHeaderCell is used by an NSTableView to draw its column headers.

Notes: See the NSTableViewMBS class specification for more information on how it's used.

Subclasses of NSTableHeaderCell can override drawInteriorWithFrame, editWithFrame, and highlight:withFrame to change the way headers appear. See the NSCell class specification, and the following description, for information on these methods. (This works in Cocoa, but does not yet work in the plugin. If you need, send in a feature request.)

Subclass of the NSTextFieldCellMBS class.

Blog Entries

- [Nearly 2000 new Functions in the 9.6 prerelease of MBS](#)

Xojo Developer Magazine

- [7.6, page 8: News](#)
- [14.4, page 29: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.62.2 Methods

9.62.3 drawSortIndicatorWithFrame(cellFrame as NSRectMBS, inView as NSViewMBS, ascending as boolean, priority as Integer)

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Draws a sorting indicator given a cellFrame contained inside controlView.

Notes: If priority is 0, this is the primary sort indicator. If ascending is true, a "∧" indicator will be drawn.

Override this method to customize the sorting user interface.

Available in Mac OS X v10.3 and later.

9.62.4 sortIndicatorRectForBounds(r as NSRectMBS) as NSRectMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the location to display the sorting indicator given rectangle.

Notes: Available in Mac OS X v10.3 and later.

9.63 class NSTableViewMBS

9.63.1 class NSTableViewMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: An NSTableView is used by an NSTableView to draw headers over its columns and to handle mouse events in those headers.

Notes: NSTableView uses NSTableViewCell to implement its user interface.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard. Subclass of the NSViewMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 12.0pr5](#)
- [Nearly 2000 new Functions in the 9.6 prerelease of MBS](#)

Xojo Developer Magazine

- [7.6, page 8: News](#)

9.63.2 Methods

9.63.3 columnAtPoint(point as NSPointMBS) as Integer

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the column whose header lies under aPoint in the receiver, or -1 if no such column is found.

Notes: point: is expressed in the receiver's coordinate system.

9.63.4 Constructor

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates a new table header view with size 100/100 and position 0/0

Example:

```
dim t as new NSTableViewMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.63.5 Constructor(Handle as Integer) 666
- 9.63.6 Constructor(left as Double, top as Double, width as Double, height as Double) 666

9.63.5 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSTableView handle.

Example:

```
dim t as new NSTableViewMBS(0, 0, 100, 100)
dim v as new NSTableViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSTableView and the plugin retains this handle.
See also:

- 9.63.4 Constructor 665
- 9.63.6 Constructor(left as Double, top as Double, width as Double, height as Double) 666

9.63.6 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates a new table header view with the given size and position.

Example:

```
dim x as new NSTableViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.
See also:

- 9.63.4 Constructor 665
- 9.63.5 Constructor(Handle as Integer) 666

9.63.7 draggedColumn as Integer

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: If the user is dragging a column in the receiver, returns the index of that column.

Notes: Otherwise returns -1.

9.63.8 draggedDistance as Double

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: If the user is dragging a column in the receiver, returns the column's horizontal distance from its original position.

Notes: Otherwise the return value is meaningless.

9.63.9 headerRectOfColumn(Column as Integer) as NSRectMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle containing the header tile for the column at columnIndex.

Notes: Raises an NSInternalInconsistencyException if columnIndex is out of bounds.

9.63.10 resizedColumn as Integer

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: If the user is resizing a column in the receiver, returns the index of that column.

Notes: Otherwise returns -1.

9.63.11 Properties

9.63.12 tableView as NSTableViewMBS

Plugin Version: 12.0, Platform: macOS, Targets: Desktop only.

Function: The owner tableview.

Notes: (Read and Write computed property)

9.64 class NSTableViewMBS

9.64.1 class NSTableViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The NSTableView class is the view shown for a row in an NSTableView.

Notes: It is responsible for displaying attributes associated with the row, including the selection highlight, and group row look.

Subclass of the NSViewMBS class.

Blog Entries

- [MBS Xojo Plugins, version 17.1pr2](#)

9.64.2 Methods

9.64.3 Constructor

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The constructor.

9.64.4 Properties

9.64.5 backgroundColor as NSColorMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The background color of the row.

Notes: The property defaults to the table view's backgroundColor, unless usesAlternatingRowBackgroundColors is set to true. In that case, the colors alternate, and are automatically updated as required by insertions and deletions.

The value of the background color can be customized in the NSTableViewMBS.didAddRowView event. The property is animatable.

(Read and Write property)

9.64.6 emphasized as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Determines whether the row will draw with the alternate or secondary color (unless overridden).

Notes: When emphasized is true, the view will draw with the alternateSelectedControlColor defined by NSColor. When false it will use the secondarySelectedControlColor defined by NSColor.

(Read and Write property)

9.64.7 Floating as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Specifies whether the row is drawn using the floating style.

Notes: Floating is a temporary attribute that is set when a particular group row is actually floating above other rows. The state may change dynamically based on the position of the group row. Drawing may be different for rows that are currently 'floating'.

(Read and Write property)

9.64.8 groupRowStyle as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Specifies whether this row view is a group row.

Notes: When true this row is a group row and will draw appropriately.

(Read and Write property)

9.64.9 NextRowSelected as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Whether next row is selected.

Notes: (Read and Write property)

9.64.10 PreviousRowSelected as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Whether the previous row is selected.

Notes: (Read and Write property)

9.64.11 selected as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Determines whether the row is selected.

Notes: True if selected, otherwise false.

(Read and Write property)

9.64.12 selectionHighlightStyle as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Specifies the selection highlight style.

Notes: The possible values are specified in NSTableViewSelectionHighlightStyle in NSTableView.

(Read and Write property)

9.65 class NSTableViewMBS

9.65.1 class NSTableViewMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: An NSTableView object displays record-oriented data in a table and allows the user to edit values and resize and rearrange columns.

Notes: You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard. Subclass of the NSViewMBS class.

Blog Entries

- [MBS Xojo Plugins, version 20.6pr1](#)
- [MBS Xojo Plugins, version 17.1pr2](#)
- [MBS Xojo / Real Studio Plugins, version 15.5pr3](#)
- [MBS Xojo / Real Studio Plugins, version 15.2pr2](#)
- [MBS Xojo / Real Studio Plugins, version 14.2pr9](#)
- [MBS Xojo / Real Studio Plugins, version 13.4pr7](#)
- [MBS Real Studio Plugins, version 13.0pr7](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MonkeyBread Software Releases the MBS REALbasic plug-ins 9.2](#)
- [MonkeyBread Software Releases the MBS REALbasic plug-ins 8.4](#)

Xojo Developer Magazine

- [6.6, page 8: News](#)
- [14.4, page 25: NSTabula Rasa, What to do when your new sports car arrives in parts by Ulrich Bogun](#)

9.65.2 Methods

9.65.3 addTableColumn(column as NSTableColumnMBS)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Adds a given column as the last column of the receiver.

9.65.4 beginUpdates

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Begins a group of updates for the table view.

Notes: For `NSView`-based table views, multiple row changes—that is, insertions, deletions, and moves—are animated simultaneously by surrounding calls to those method calls with `beginUpdates` and `endUpdates`. These methods are nestable.

The selected rows are maintained during the series of insertions, deletions, moves, and scrolling. If a selected row is deleted, a selection changed notification occurs after `removeRowsAtIndexes` is called.

It is not necessary to call `beginUpdates` and `endUpdates` if only one insertion, deletion, or move is occurring and the table view is an `NSView`-based table view. When using an `NSCell`-based table view, you must surround any insertion, deletion, or move in an update block for animations to occur.

The main reason for doing a batch update of changes to a table view is to avoid having the table animate unnecessarily.

Note that these methods should be called to reflect changes in your model; they do not make any underlying model changes.

For `NSCell`-based table views, it is required to call `beginUpdates` if you want to animate the `insertRowsAtIndexes`, `removeRowsAtIndexes`, and `moveRowAtIndexPath`.

9.65.5 canDragRowsWithIndexes(rowIndexes as NSIndexSetMBS, mouseDownPoint as NSPointMBS) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value indicating whether the table view allows dragging the rows at with the drag initiated at the specified point.

Notes: `rowIndexes`: The row indexes to drag.

`mouseDownPoint`: The location where the drag was initiated.

Returns no to disallow the drag.

9.65.6 columnAtPoint(p as NSPointMBS) as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the column a given point lies in.

Notes: Returns the index of the column `aPoint` lies in, or `-1` if `aPoint` lies outside the receiver's bounds.

See also:

- 9.65.7 `columnAtPoint(x as Double, y as Double) as Integer`

9.65.7 columnAtPoint(x as Double, y as Double) as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the column a given point lies in.

Notes: Returns the index of the column aPoint lies in, or -1 if aPoint lies outside the receiver's bounds.

See also:

- 9.65.6 columnAtPoint(p as NSPointMBS) as Integer

672

9.65.8 columnForView(view as NSViewMBS) as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the column index for the specified view.

Notes: view: The view for which to retrieve the column.

Returns the index of the column containing view in the tableColumns array. This method returns -1 if the view is not in the table view. This method may also return -1 if the row containing the view is being animated away, such as during the deletion of a row.

This method is typically called in the action method of an NSButton (or NSControl) to find out what row (and column) the action should be performed on.

The implementation is $O(n)$ where n is the number of visible rows, so this method should generally not be called within a loop.

9.65.9 columnIndexesInRect(rect as NSRectMBS) as NSIndexSetMBS

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the indexes of the receiver's columns that intersect the specified rectangle.

Notes: Available in Mac OS X v10.5 and later.

9.65.10 columnWithIdentifier(identifier as string) as Integer

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the first column in the receiver whose identifier is equal to a given identifier.

Notes: Returns the index of the first column in the receiver whose identifier is equal to anObject (when compared using isEqual:) or -1 if no columns are found with the specified identifier.

9.65.11 Constructor

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new table view with size 100/100 and position 0/0

Example:

```
dim t as new NSTableViewMBS
```

Notes: On success the handle property is not zero.

See also:

- 9.65.12 Constructor(Handle as Integer) 674
- 9.65.13 Constructor(left as Double, top as Double, width as Double, height as Double) 674

9.65.12 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given NSTableView handle.

Example:

```
dim t as new NSTableViewMBS(0, 0, 100, 100)
```

```
dim v as new NSTableViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a NSTableView and the plugin retains this handle.

See also:

- 9.65.11 Constructor 674
- 9.65.13 Constructor(left as Double, top as Double, width as Double, height as Double) 674

9.65.13 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Creates a new table view with the given size and position.

Example:

```
dim x as new NSTableViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 9.65.11 Constructor 674
- 9.65.12 Constructor(Handle as Integer) 674

9.65.14 deselectAll

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Deselects all selected rows or columns if empty selection is allowed; otherwise does nothing.

9.65.15 deselectColumn(column as Integer)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Deselects the column at a given index if it's selected.

Notes: Column is from 0 to numberOfColumns-1.

9.65.16 deselectRow(row as Integer)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Deselects the row at a given index if it's selected.

Notes: Deselects the row at rowIndex if it's selected, regardless of whether empty selection is allowed.

9.65.17 Destructor

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: The destructor.

Notes: Xojo calls this destructor automatically.

9.65.18 dragImageForRowsWithIndexes(dragRows as NSIndexSetMBS, tableColumns() as NSTableColumnMBS, theEvent as NSEventMBS, byref dragImageOffset as NSPointMBS) as NSImageMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Computes and returns an image to use for dragging.

Notes: `dragRows`: An index set containing the row indexes that should be in the image.

`tableColumns`: An array of table columns that should be in the image.

`dragEvent`: The event that initiated the drag.

`dragImageOffset`: An in/out parameter specifying the offset of the cursor in the image, the default value is `NSZeroPoint`. Returning `NSZeroPoint` causes the cursor to be centered.

Returns an `NSImage` containing a custom image for the specified rows and columns participating in the drag.

9.65.19 `edit(column as Integer, row as Integer, selectit as boolean)`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Edits the cell at `columnIndex` and `rowIndex`, selecting its entire contents if `flag` is true.

Notes: This method is invoked automatically in response to user actions; you should rarely need to invoke it directly.

This method scrolls the receiver so that the cell is visible, sets up the field editor, and sends `selectWithFrame()` and `editWithFrame()` to the field editor's `NSCell` object with the `NSTableView` as the text delegate.

The row at `rowIndex` must be selected prior to calling `editColumn:row:withEvent:select:`, or an exception will be raised.

9.65.20 `endUpdates`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Ends the group of updates for the table view.

Notes: Ends the group of updates for the table view. This method, like `beginUpdates`, is nestable. See `beginUpdates` for details.

9.65.21 `frameOfCellAtColumnRow(column as Integer, row as Integer) as NSRectMBS`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a rectangle locating the cell that lies at the intersection of the specified column and row.

Notes: `column`: The index in the `tableColumns` array of the column containing the cell whose rectangle you want.

`row`: The index of the row containing the cell whose rectangle you want.

Returns a rectangle locating the cell that lies at the intersection of `columnIndex` and `rowIndex`. This method returns `NSRectMBS.Zero` if `columnIndex` or `rowIndex` is greater than the number of columns or rows in the table view.

You can use this method to update a single cell more efficiently than sending the table view a `reloadData` message using `reloadData` function.

The result of this method is used in a `drawWithFrame:inView:` message to the table column's data cell. You can subclass and override this method to customize the frame of a particular cell. However, never return a frame larger than the default implementation returns.

The default frame is computed to have a height equal to the `rectOfRow:` for `rowIndex`, minus the half `intercellSpacing` height on the top and half on the bottom. The width of frame is equal to the width of the table column minus half the `intercellSpacing` width on the left, and half on the right.

9.65.22 `hiddenRowIndexes` as `NSIndexSetMBS`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The indexes of all hidden table rows.

Notes: The value of this property is an index set containing the indexes of any hidden table rows. Table rows may be hidden by invoking the `hideRowsAtIndexes` method. Some drag-and-drop operations also result in hidden rows.

9.65.23 `hideRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Hides the specified table rows.

Notes: `indexes`: An index set containing indexes of the rows to be hidden.

`rowAnimation`: An animation effect to be applied when the rows are hidden.

Use this method when you no longer want the data to be visible to the user, but you don't want to permanently remove the data. Hidden table rows have a height of zero and cannot be selected by the user. However, if a selected table row is hidden, it will remain selected.

Hiding a table row causes the `didRemoveRowView` delegate method to be invoked.

9.65.24 `insertRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Inserts the rows using the specified animation.

Notes: indexes: The final positions of the new rows to be inserted.

animationOptions: The animation displayed during the insert. See `NSTableViewAnimationOptions` for the possible values that can be combined using the bitwise OR operator.

The `numberOfRows` in the table view is automatically increased by the count of indexes.

Calling this method multiple times within the same `beginUpdates` and `endUpdates` block is allowed, and changes are processed incrementally.

NSCell-based table views must first call `beginUpdates` before calling this method.

9.65.25 `isColumnSelected(column as Integer)` as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether the column at a given index is selected.

Notes: column is from 0 to to `numberOfColumns`.

9.65.26 `isRowSelected(row as Integer)` as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether the row at a given index is selected.

Notes: row is from 0 to to `numberOfRows`.

9.65.27 `moveColumn(column as Integer, toIndex as Integer)`

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: Moves the column and heading at a given index to a new given index.

Notes: columnIndex: The current index of the column to move.

newIndex: The new index for the moved column.

This method raises the `ColumnDidMove` event.

9.65.28 `moveRowAtIndex(oldIndex as Integer, newIndex as Integer)`

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Moves the specified row to the new row location using animation.

Notes: oldIndex: Initial row index.
newIndex: New row index.

This is similar to removing a row at oldIndex and inserting it at newIndex, except the same view is used and simply has its position updated to the new location. Changes happen incrementally as they are sent to the table, so as soon as this method is called the row can be considered moved. However the underlying view is not moved until endUpdates has been called. This method can be called multiple times within the same beginUpdates and endUpdates block.

NSCell-based table views must first call beginUpdates before calling this method.

9.65.29 noteHeightOfRowsWithIndexesChanged(indexSet as NSIndexSetMBS)

Plugin Version: 16.0, Platform: macOS, Targets: Desktop only.

Function: Informs the table view that the rows specified in indexSet have changed height.

Notes: indexSet: Index set of rows that have changed their height.

If you implement heightForRow event this method immediately retiles the table view using the row heights the event provides.

For NSView-based tables, this method will animate. To turn off the animation, create an NSAnimationContext grouping and set the duration to 0. Then call this method and end the grouping.

For NSCell-based tables, this method normally doesn't animate. However, it will animate if you call it inside a beginUpdates/endUpdates block.

9.65.30 noteNumberOfRowsChanged

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Informs the receiver that the number of records in its data source has changed.

Notes: This method allows the receiver to update the scrollers in its scroll view without actually reloading data into the receiver. It's useful for a data source that continually receives data in the background over a period of time, in which case the table view can remain responsive to the user while the data is received.

9.65.31 rectOfColumn(column as Integer) as NSRectMBS

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle containing the column at a given index.

Notes: The rectangle containing the column at columnIndex. Returns NSRectMBS.Zero (an empty rectangle) if columnIndex lies outside the range of valid column indices for the receiver.

You can use this method to update a single column more efficiently than sending the table view a reloadData message.

```
aTableView.setNeedsDisplayInRect(aTableView.rectOfColumn(column))
```

9.65.32 rectOfRow(row as Integer) as NSRectMBS

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the rectangle containing the row at a given index.

Notes: Returns the rectangle containing the row at rowIndex. Returns NSRectMBS.Zero (an empty rectangle) if rowIndex lies outside the range of valid row indices for the receiver.

9.65.33 reloadData

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Marks the receiver as needing redisplay, so it will reload the data for visible cells and draw the new values.

Notes: This method forces redraw of all the visible cells in the receiver. If you want to update the value in a single cell, column, or row, it is more efficient to use frameOfCellAtColumn(), rectOfColumn(), or rectOfRow() in conjunction with setNeedsDisplayInRect(). If you just want to update the scroller, use numberOfRowsChanged; if the height of a set of rows changes, use noteHeightOfRowsWithIndexesChanged().

See also:

- 9.65.34 reloadData(rowIndexes as NSIndexSetMBS, columnIndexes as NSIndexSetMBS) 680

9.65.34 reloadData(rowIndexes as NSIndexSetMBS, columnIndexes as NSIndexSetMBS)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Reloads the data for only the specified rows and columns.

Notes: rowIndexes: The indexes of the rows to update.

columnIndexes: The indexes of the columns to update.

For cells that are visible, the appropriate dataSource and delegate methods are called and the cells are redrawn.

For tables that support variable row heights, the row height is not re-queried from the delegate; it is your responsibility to invoke noteHeightOfRowsWithIndexesChanged if a row height change is required.

For NSView-based table views, this method drops the view-cells in the table row, but not the NSTableRowView instances.

See also:

- 9.65.33 reloadData 680

9.65.35 removeRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Removes the rows using the specified animation.

Notes: indexes: An index set containing the rows to remove.

animationOptions: The animation displayed during the insert. See NSTableViewAnimationOptions for the possible values that can be combined using the bitwise OR operator.

This method deletes from the table the rows represented at indexes and automatically decreases numberOfRows by the count of indexes.

The row indexes should be with respect to the current state displayed in the table view, and not the final state, because the specified rows do not exist in the final state.

Calling this method multiple times within the same beginUpdates and endUpdates block is allowed, and changes are processed incrementally.

Changes are processed incrementally as the insertRowsAtIndexes, removeRowsAtIndexes, and the moveRowAtIndex methods are called. It is acceptable to delete row 0 multiple times, as long as there is still a row available.

NSCell-based table views must first call beginUpdates before calling this method.

9.65.36 removeTableColumn(column as NSTableColumnMBS)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Removes a given column from the receiver.

9.65.37 rowAtPoint(p as NSPointMBS) as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the row a given point lies in.

Notes: Returns the index of the row aPoint lies in, or -1 if aPoint lies outside the receiver's bounds.

See also:

- 9.65.38 rowAtPoint(x as Double, y as Double) as Integer 682

9.65.38 rowAtPoint(x as Double, y as Double) as Integer

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the row a given point lies in.

Notes: Returns the index of the row aPoint lies in, or -1 if aPoint lies outside the receiver's bounds.

See also:

- 9.65.37 rowAtPoint(p as NSPointMBS) as Integer 681

9.65.39 rowForView(view as NSViewMBS) as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the row for the specified view.

Notes: view: The view for which to retrieve the row.

Returns the index of the row containing to view. This method returns -1 if the view is not in the table view. This method may also return -1 if the row containing the view is being animated away, such as during the deletion of a row.

This method is typically called in the action method for an NSButton (or NSControl) to find out what row (and column) the action should be performed on.

The implementation is $O(n)$ where n is the number of visible rows, so this method should generally not be called within a loop.

9.65.40 rowsInRect(rect as NSRectMBS) as NSRangeMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a range of indexes for the rows that lie wholly or partially within the vertical boundaries of the specified rectangle.

Notes: Rect: A rectangle in the coordinate system of the table view.

Returns a range of indexes for the table view,Ãs rows that lie wholly or partially within the horizontal boundaries of aRect. If the width or height of aRect is 0, this method returns an NSRange whose length is

0.

The location of the range is the index of the first row in the rectangle, and the length is the number of rows that lie in the rectangle.

9.65.41 rowViewAtRow(row as Integer, makeIfNecessary as Boolean) as NSViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a row view at the specified index, creating one if necessary.

Notes: row: The row index.

makeIfNecessary: True if a view is required, NO if you want to update properties on a view, if one is available.

Returns an instance, or subclass, of NSTableViewRowView. Returning nil is also valid if makeIfNecessary is false and the view did not exist.

This method first attempts to return a currently displayed view in the visible area. If there is no visible view, and makeIfNecessary is true, a prepared temporary view is returned. If makeIfNecessary is false, and the view is not visible, nil is returned.

In general, makeIfNecessary should be true if you require a resulting view, and false if you want to update properties on a view only if it is available (generally this means it is visible).

An exception is thrown if row falls outside of the number of rows in the table (numberOfRows). The returned result should generally not be held onto for longer than the current run loop cycle. It's better to call rowViewAtRow whenever a view is required.

9.65.42 scrollColumnToVisible(column as Integer)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Scrolls the receiver and header view horizontally in an enclosing NSClipView so the column specified by columnIndex is visible.

9.65.43 scrollRowToVisible(row as Integer)

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Scrolls the receiver vertically in an enclosing NSClipView so the row specified by rowIndex is visible.

9.65.44 ScrollToLine(Line as Integer, Animated as Boolean)

Plugin Version: 16.0, Platform: macOS, Targets: Desktop only.

Function: Scrolls to line.

Notes: If animated is true, the scroll is animated.

The line is centered in the middle of the viewable area if possible.

9.65.45 selectAll

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Selects all rows or all columns, according to whether rows or columns were most recently selected.

Notes: If the table allows multiple selection, this action method selects all rows or all columns, according to whether rows or columns were most recently selected. If nothing has been recently selected, this method selects all rows. If this table doesn't allow multiple selection, this method does nothing.

9.65.46 selectColumnIndexes(indexes as NSIndexSetMBS, extend as boolean)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Sets the column selection using indexes.

Notes: If the extend flag is false the selected columns are specified by indexes. If extend is true, the columns indicated by indexes are added to the collection of already selected columns, providing multiple selection.

9.65.47 selectedColumnIndexes as NSIndexSetMBS

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns an index set containing the indexes of the selected columns.

Notes: Returns an index set containing the indexes of the selected columns.

9.65.48 selectedRowIndex as NSIndexSetMBS

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Returns an index set containing the indexes of the selected rows.

Notes: Available in Mac OS X v10.3 and later.

9.65.49 selectRowIndexes(indexes as NSIndexSetMBS, extend as boolean)

Plugin Version: 9.1, Platform: macOS, Targets: Desktop only.

Function: Sets the row selection using indexes.

Notes: If the extend flag is false the selected rows are specified by indexes. If extend is true, the rows indicated by indexes are added to the collection of already selected rows, providing multiple selection.

Available in Mac OS X v10.3 and later.

9.65.50 setDraggingSourceOperationMask(mask as Integer, isLocal as Boolean)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Sets the default operation mask returned by draggingSourceOperationMaskForLocal to mask.

Notes: mask: The drag operation mask. See NSDragOperation for the supported values.

isLocal: True if the destination is the same application, otherwise false. In either case the specified mask value is archived and used.

9.65.51 setDropRow(row as Integer, dropOperation as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Retargets the proposed drop operation.

Notes: row: The target row index.

dropOperation: The drop operation. Supported values are specified by NSTableViewDropOperation.

For example, to specify a drop on the second row, specify row as 1, and operation as NSTableViewDropOn. To specify a drop below the last row, specify row as [self numberOfRows] and operation as NSTableViewDropAbove.

Passing a value of -1 for row and NSTableViewDropOn as the operation causes the entire table view to be highlighted rather than a specific row. This is useful if the data displayed by the table view does not allow the user to drop items at a specific row location.

9.65.52 setSortDescriptor(sortDescriptor as NSSortDescriptorMBS)

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: Sets the receiver's sort descriptors.

Notes: A table column is considered sortable if it has a sort descriptor that specifies the sorting direction, a key to sort by, and a selector defining how to sort.

The array of sort descriptors is archived. Sort descriptors persist along with other column information if an autosave name is set.

Calling `setSortDescriptors` may have the side effect of invoking the data source method `tableViewSortDescriptorsDidChange`.

9.65.53 `setSortDescriptors(sortDescriptors())` as `NSSortDescriptorMBS`

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: Sets the receiver's sort descriptors.

Notes: A table column is considered sortable if it has a sort descriptor that specifies the sorting direction, a key to sort by, and a selector defining how to sort.

The array of sort descriptors is archived. Sort descriptors persist along with other column information if an autosave name is set.

Calling `setSortDescriptors` may have the side effect of invoking the data source method `tableViewSortDescriptorsDidChange`.

9.65.54 `sizeLastColumnToFit`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Resizes the last column if there's room so the receiver fits exactly within its enclosing clip view.

9.65.55 `sizeToFit`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Changes the width of columns in the receiver so all columns are visible.

Notes: All columns are resized to the same size, up to a column's maximum size. This method then invokes `tile`.

9.65.56 `sortDescriptors` as `NSSortDescriptorMBS()`

Plugin Version: 14.2, Platform: macOS, Targets: Desktop only.

Function: Returns the receiver's sort descriptors.

9.65.57 tableColumns as NSTableColumnMBS()

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns an array containing the the NSTableColumn objects in the receiver.

Notes: The array returned by tableColumns contains all receiver's columns, including those that are hidden.

9.65.58 tableColumnWithIdentifier(identifier as string) as NSTableColumnMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: Returns the NSTableColumn object for the first column whose identifier is equal to a given object.

Notes: Returns the NSTableColumn object for the first column whose identifier is equal to anObject, as compared using isEqual:, or nil if no columns are found with the specified identifier.

9.65.59 tile

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Properly sizes the receiver and its header view and marks it as needing display.

Notes: Also resets cursor rectangles for the header view and line scroll amounts for the NSScrollView object.

9.65.60 unhideRowsAtIndexes(indexes as NSIndexSetMBS, animationOptions as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Unhides the specified table rows.

Notes: indexes: An index set containing indexes of the hidden rows to be shown again.

rowAnimation: An animation effect to be applied when the rows are hidden.

Unhiding a table row causes the didAddRowView event to be invoked.

9.65.61 viewAtColumn(column as Integer, row as Integer, makeIfNecessary as Boolean) as NSViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Returns a view at the specified row and column indexes, creating one if necessary.

Notes: column: The index of the column in the tableColumns array.

row: The row index.

makeIfNecessary: true if a view is required, false if you want to update properties on a view, if one is available.

Returns an instance of `NSView`.

This method first attempts to return an available view, which is generally in the visible area. If there is no available view, and `makeIfNecessary` is true, a prepared temporary view is returned. If `makeIfNecessary` is false, and the view is not available, nil will be returned.

In general, `makeIfNecessary` should be true if you require a resulting view, and false if you only want to update properties on a view only if it is available (generally this means it is visible).

An exception will be thrown if `row` is not within the `numberOfRows`. The returned result should generally not be held onto for longer than the current run loop cycle. Instead they should re-query the table view for the row view.

9.65.62 Properties

9.65.63 `allowsColumnReordering` as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Controls whether the user can drag column headers to reorder columns.

Notes: The default is true. You can rearrange columns programmatically regardless of this setting. (Read and Write property)

9.65.64 `allowsColumnResizing` as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the receiver allows the user to resize columns by dragging between their headers.

Notes: (Read and Write property)

9.65.65 `allowsColumnSelection` as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Controls whether the user can select an entire column by clicking its header.

Notes: The default is false. You can select columns programmatically regardless of this setting. (Read and Write property)

9.65.66 allowsEmptySelection as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Controls whether the receiver allows zero rows or columns to be selected.

Notes: Unlike with the other settings that affect selection behavior, you cannot set an empty selection programmatically if empty selection is disallowed.

(Read and Write property)

9.65.67 allowsMultipleSelection as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Controls whether the user can select more than one row or column at a time.

Notes: The default is false. You can select multiple columns or rows programmatically regardless of this setting.

(Read and Write property)

9.65.68 allowsTypeSelect as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver allows the user to type characters to select rows.

Notes: Available in Mac OS X v10.5 and later.

(Read and Write property)

9.65.69 autosaveName as string

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The name under which table information is automatically saved.

Notes: The table information is saved separately for each user and for each application that user uses.

Note that even when a table view has an autosave name, it may not be saving table information automatically.

(Read and Write property)

9.65.70 autosaveTableColumns as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Whether the order and width of this table view's columns are automatically saved.

Notes: If flag is different from the current value, this method also reads in the saved information and sets the table options to match.

The table information is saved separately for each user and for each application that user uses. Note that if `autosaveName` returns `nil`, this setting is ignored and table information isn't saved.
(Read and Write property)

9.65.71 `backgroundColor` as `NSColorMBS`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The color used to draw the background of the receiver.

Notes: The default background color is light gray.

(Read and Write property)

9.65.72 `clickedColumn` as `Integer`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the column the user clicked to trigger an action message.

Notes: The index of the column the user clicked to trigger an action message. Returns `-1` if the user clicked in an area of the table view not occupied by columns.

Index is zero based.

(Read only property)

9.65.73 `clickedRow` as `Integer`

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the row the user clicked to trigger an action message.

Notes: The index of the row the user clicked to trigger an action message. Returns `-1` if the user clicked in an area of the table view not occupied by table rows.

Index is zero based.

(Read only property)

9.65.74 columnAutoresizingStyle as Integer

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: The table's column autoresizing style.

Notes: Use the following constants:

```
NSTableViewMBS.kFirstColumnOnlyAutoresizingStyle  
NSTableViewMBS.kUniformColumnAutoresizingStyle  
NSTableViewMBS.kSequentialColumnAutoresizingStyle  
NSTableViewMBS.kReverseSequentialColumnAutoresizingStyle  
NSTableViewMBS.kNoColumnAutoresizing  
NSTableViewMBS.kLastColumnOnlyAutoresizingStyle
```

Available in Mac OS X v10.4 and later.

(Read and Write property)

9.65.75 cornerView as NSViewMBS

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: The view used to draw the area to the right of the column headers and above the vertical scroller of the enclosing scroll view.

Notes: This is by default a simple view that merely fills in its frame, but you can replace it with a custom view using this property.

(Read and Write property)

9.65.76 dataSource as NSTableDataSourceMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The object that provides the data displayed in the table view.

Notes: (Read and Write property)

9.65.77 draggingDestinationFeedbackStyle as Integer

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The feedback style displayed when the user drags over the table view.

Notes: Available in Mac OS X v10.6 and later.

(Read and Write property)

9.65.78 editedColumn as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the column being edited.

Notes: First column has index zero.

If sent during editColumn(), the index of the row being edited; otherwise -1.

(Read only property)

9.65.79 editedRow as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the row being edited.

Notes: The first row index is zero.

If sent during editColumn(), the index of the row being edited; otherwise -1.

(Read only property)

9.65.80 effectiveRowSizeStyle as Integer

Plugin Version: 13.4, Platform: macOS, Targets: Desktop only.

Function: Returns the effective row size style for the table.

Notes: If the rowSizeStyle is NSTableViewRowSizeStyleDefault, then this method returns the default size for this table.

The default size is currently set in the System Preferences by the users.

Available in OS X v10.7 and later.

(Read only property)

9.65.81 effectiveStyle as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The effective style when style is NSTableViewStyleAutomatic.

Notes: Otherwise, it returns the same value as style.

(Read only property)

9.65.82 floatsGroupRows as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table view draws grouped rows as if they are floating.

Notes: Group rows are rows for which the table view delegate's `isGroupRow` method returns true. These rows can be displayed as if they are floating in a view-based table view.

The default value of this property is true.

(Read and Write property)

9.65.83 focusedColumn as Integer

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The currently focused column.

Notes: Returns the index of the column, or -1 if there is no focused column

The focus interaction will always be on the selectedRow of the table. If the selectedRow is a full width cell, then focusedColumn will return 1 when focused..

Available in Mac OS X v10.6 and later.

(Read and Write property)

9.65.84 gridColor as NSColorMBS

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The color used to draw grid lines.

Notes: The default color is gray.

(Read and Write property)

9.65.85 gridStyleMask as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The receiver's grid style mask.

Notes: use the constants;

<code>kGridNone</code>	= 0,
<code>kSolidVerticalGridLineMask</code>	= 1
<code>kSolidHorizontalGridLineMask</code>	= 2

(Read and Write property)

9.65.86 headerView as NSTableHeaderViewMBS

Plugin Version: 9.6, Platform: macOS, Targets: Desktop only.

Function: The NSTableHeaderView object used to draw headers over columns.

Notes: The NSTableHeaderView object used to draw headers over columns, or nil if the receiver has no header view

(Read and Write property)

9.65.87 highlightedtableColumn as NSTableColumnMBS

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: The table column highlighted in the receiver.

Notes: A highlightable column header can be used in conjunction with row selection to highlight a particular column of the table. An example of this is how the Mail application indicates the currently sorted column.

(Read and Write property)

9.65.88 intercellSpacing as NSSizeMBS

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: The horizontal and vertical spacing between cells.

Notes: The default spacing is (3.0, 2.0).

(Read and Write property)

9.65.89 numberOfColumns as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of columns in the receiver.

Notes: The value returned includes table columns that are currently hidden.

(Read only property)

9.65.90 numberOfRows as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of rows in the receiver.

Notes: Typically you should not ask the table view how many rows it has; instead you should interrogate the table view's data source.

(Read only property)

9.65.91 numberOfSelectedColumns as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of selected columns.

Notes: (Read only property)

9.65.92 numberOfSelectedRows as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of selected rows.

Notes: (Read only property)

9.65.93 rowActionsVisible as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether a table row's actions are visible.

Notes: This property contains a Boolean value indicating whether a table row's actions are visible or not—the user has swiped the row to reveal the row actions. Set the value of this property to false to hide any visible row actions. Setting the value of this property to true is not supported, and will result in an exception.

(Read and Write property)

9.65.94 rowHeight as Double

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the height of each row in the receiver.

Notes: The default row height is 16.0.

(Read and Write property)

9.65.95 rowSizeStyle as Integer

Plugin Version: 13.4, Platform: macOS, Targets: Desktop only.

Function: The row size used by the tableview: small, medium, large, or on a custom row by row basis.

Notes: The row size style can be modified on a row by row basis by invoking the event `heightOfRow`, if implemented.

The `rowSizeStyle` defaults to `NSTableViewRowSizeStyleCustom`. `NSTableViewRowSizeStyleCustom` indicates to use the `rowHeight` of the table, instead of the pre-determined system values.

Generally, `rowSizeStyle` should always be `NSTableViewRowSizeStyleCustom` except for "source lists". To implement variable row heights, set the value to `NSTableViewRowSizeStyleCustom` and implement `tableView:heightOfRow:` in the delegate.

Available in OS X v10.7 and later.
(Read and Write property)

9.65.96 `selectedColumn` as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the last column selected or added to the selection.

Notes: Returns the index of the last column selected or added to the selection, or `-1` if no column is selected.
(Read only property)

9.65.97 `selectedRow` as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Returns the index of the last row selected or added to the selection.

Notes: Returns the index of the last row selected or added to the selection, or `-1` if no row is selected.
(Read only property)

9.65.98 `selectionHighlightStyle` as Integer

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: The selection highlight style used by the receiver to indicate row and column selection.

Notes: Available in Mac OS X v10.5 and later.
(Read and Write property)

9.65.99 style as Integer

Plugin Version: 21.0, Platform: macOS, Targets: Desktop only.

Function: The table view style.

Example:

```
dim tableView as NSTableViewMBS // your table view
```

```
tableView.style = NSTableViewMBS.NSTableViewStyleFullWidth
```

Notes: Defaults to NSTableViewStyleAutomatic.

For macOS 11.0 or newer.

Available styles:

NSTableViewStyleFullWidth	The table view style resolves to a full-width style.
NSTableViewStyleInset	The table view style resolves to an inset style.
NSTableViewStyleSourceList	The table view style resolves to a source-list style.
NSTableViewStylePlain	The table view style resolves to a plain style.

(Read and Write property)

9.65.100 usesAlternatingRowBackgroundColors as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the receiver uses the standard alternating row colors for its background.

Notes: Available in Mac OS X v10.3 and later.

(Read and Write property)

9.65.101 usesStaticContents as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: A Boolean value indicating whether the table uses static data.

Notes: A static table does not rely on a data source to provide the number of rows. A static table view's contents are set at design time and can be changed programmatically as needed. Typically, you do not change the contents of a static table view after setting them.

In Xcode, any rows you add to a static table are saved in the corresponding nib or storyboard file and

loaded with the rest of the table at runtime. You can add table rows programmatically to a static table view using the `insertRowsAtIndexes` method. When adding rows programmatically, your table view delegate must implement the `view` method to provide the corresponding view for any new rows. You can also remove rows at any time using the `removeRowsAtIndexes` method.

A table with static contents must be an `NSView`-based table view.

Available in macOS 10.10 or later.
(Read and Write property)

9.65.102 `verticalMotionCanBeginDrag` as boolean

Plugin Version: 8.4, Platform: macOS, Targets: Desktop only.

Function: Whether vertical motion is treated as a drag or selection change to flag.

Notes: If flag is false then vertical motion will not start a drag. The default is true.

Note that horizontal motion is always a valid motion to begin a drag. Most often, you would want to disable vertical dragging when it's expected that horizontal dragging is the natural motion.

(Read and Write property)

9.65.103 `indicatorImageInTableColumn(column as NSTableColumnMBS)` as `NSImageMBS`

Plugin Version: 9.2, Platform: macOS, Targets: Desktop only.

Function: The indicator image of a given table column.

Notes: An indicator image is an arbitrary (small) image that is rendered on the right side of the column header. An example of its use is in Mail to indicate the sorting direction of the currently sorted column in a mailbox.

(Read and Write computed property)

9.65.104 Events**9.65.105 ColumnDidMove(notification as NSNotificationMBS, oldColumn as Integer, newColumn as Integer)**

Plugin Version: 9.2, Platform: macOS, Targets: .

Function: This event informs the delegate that a column was moved by user action in the table view.

9.65.106 ColumnDidResize(notification as NSNotificationMBS, column as NSTableColumnMBS, index as Integer)

Plugin Version: 9.2, Platform: macOS, Targets: .

Function: This event informs you that a column was resized in the table view.

9.65.107 dataCell(tableColumn as NSTableColumnMBS, row as Int64) as NSCellMBS

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Optional return a different cells for each row.

Notes: A different data cell can be returned for any particular tableColumn and row, or a cell that will be used for the entire row (a full width cell). The returned cell should properly implement copyWithZone:, since the cell may be copied by NSTableView. If the tableColumn is non-nil, and nil is returned, then the table will use the default cell from tableColumn.dataCellForRow(Row).

When each row is being drawn, this method will first be called with a nil tableColumn. At this time, you can return a cell that will be used to draw the entire row, acting like a group. If you do return a cell for the 'nil' tableColumn, be prepared to have the other corresponding datasource and delegate methods to be called with a 'nil' tableColumn value. If don't return a cell, the method will be called once for each tableColumn in the tableView, as usual.

9.65.108 didAddRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Tells the delegate that a row view was added at the specified row.

Notes: rowView: The row view.

row: The index of the row.

At this point, the delegate can add extra views, or modify the properties of rowView.

This method is only valid for NSView-based table views.

9.65.109 didClickTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Called if a table column was clicked on.

9.65.110 didDragTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Sent at the time the mouse button goes up in tableView and tableColumn has been dragged during the time the mouse button was down.

Notes: tableColumn: The table column.

The behavior of this method on Mac OS X v10.5 is different from prior versions. On Mac OS X v 10.5 the dragged column is sent to the subclass. In earlier versions the table column that is currently located at the dragged column's original index is sent.

9.65.111 didRemoveRowView(rowView as NSTableRowViewMBS, row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Tells the delegate that a row view was removed from the table at the specified row.

Notes: rowView: The row view.

row: The index of the row.

If row equals -1, the row is being deleted from the table and is no longer a valid row; otherwise row is a valid row that is being removed by being moved off screen.

This method is only valid for NSView-based table views.

9.65.112 DoubleClick

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: A double click was recognized.

9.65.113 heightOfRow(row as Int64) as Double

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Implement this event to support a table with varying row heights.

Notes: The height returned by this method should not include intercell spacing and must be greater than zero. Performance Considerations: For large tables in particular, you should make sure that this method is efficient. NSTableView may cache the values this method returns, but this should NOT be depended on, as all values may not be cached. To signal a row height change, call noteHeightOfRowsWithIndexesChanged. For a given row, the same row height should always be returned until noteHeightOfRowsWithIndexesChanged is called, otherwise unpredictable results will happen. NSTableView automatically invalidates its entire row height cache in reloadData, and noteNumberOfRowsChanged.

9.65.114 isGroupRow(row as Int64) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to indicate that a specified row is a group row.

Notes: row: The row index.

Return true if the specified row should have the group row style drawn, false otherwise.

If the cell in row is an NSTextFieldCell and contains only a string, the group row style attributes will automatically be applied to the cell.

Group rows in view-based table views can be made to visually ‚float‘ by setting the tableview method setFloatsGroupRows to true.

Note: When configured as a source list style table view, rows identified as group rows draw with a specific style unique for source lists.

Available in Mac OS X v10.5 and later.

9.65.115 mouseDownInHeaderOfTableColumn(tableColumn as NSTableColumnMBS)

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Sent to the subclass whenever the mouse button is clicked in the table view’s header column.

Notes: tableColumn: The table column.

9.65.116 nextTypeSelectMatchFromRow(startRow as Int64, endRow as Int64, searchString as string) as Int64

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to modify how type selection works.

Notes: startRow: The starting row of the search range.

endRow: The ending row of the search range.

searchString: A string containing the typed selection.

Return the first row in the range of startRow through endRow (excluding endRow itself) that matches selectionString. Return -1 if no match is found.

It is possible for endRow to be less than startRow if the search will wrap.

Available in Mac OS X v10.5 and later.

9.65.117 rowViewForRow(row as Integer) as NSTableViewMBS

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Asks the delegate for a view to display the specified row.

Notes: row: The row index.

Return an instance or subclass of NSTableView. If nil is returned, an NSTableView instance will be created and used.

You can implement this event to return a custom NSTableView for row.

The reuse queue can be used in the same way as documented in tableView:view:row:. The returned view will have attributes properly set to it before it's added to the tableView.

This method is only valid for NSView-based table views.

9.65.118 SelectionDidChange(notification as NSNotificationMBS)

Plugin Version: 9.2, Platform: macOS, Targets: .

Function: This event informs you that the table view's selection has changed.

9.65.119 selectionIndexesForProposedSelection(proposedSelectionIndexes as NSIndexSetMBS) as NSIndexSetMBS

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the delegate to modify the proposed selection.

Notes: proposedSelectionIndexes: An index set containing the indexes of the proposed selection.

Return an NSIndexSet instance containing the indexes of the new selection. Return proposedSelectionIndexes if the proposed selection is acceptable, or the value of the table view's existing selection to avoid changing the selection.

This method may be called multiple times with one new index added to the existing selection to find out if a particular index can be selected when the user is extending the selection with the keyboard or mouse.

Implementation of this method is optional. If implemented, this method will be called instead of shouldSelectRow.

Available in Mac OS X v10.5 and later.

9.65.120 SelectionIsChanging(notification as NSNotificationMBS)

Plugin Version: 9.2, Platform: macOS, Targets: .

Function: This event informs you that the table view's selection is in the process of changing (typically because the user is dragging the mouse across a number of rows).

9.65.121 selectionShouldChangeInTableView as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Returns whether the selection should change.

Notes: Return true to allow the table view to change its selection (typically a row being edited), false to deny selection change.

The user can select and edit different cells within the same row, but can't select another row unless the delegate approves. The subclass can implement this method for complex validation of edited rows based on the values of any of their cells.

9.65.122 shouldEditTableColumn(tableColumn as NSTableColumnMBS, row as Int64) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Returns whether the cell at the specified row and column can be edited.

Notes: TableColumn: The table column.

rowIndex: The row index.

Return true to allow editing the cell, false to deny editing.

The subclass can implement this method to disallow editing of specific cells.

Note: This method is only valid for cell-based table views.

9.65.123 shouldReorderColumn(columnIndex as Int64, newColumnIndex as Int64) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Sent to the subclass to allow or prohibit the specified column to be dragged to a new location.

Notes: columnIndex: The index of the column being dragged.

newColumnIndex: The proposed target index of the column.

Return true if the column reordering should be allowed, otherwise false.

When a column is initially dragged by the user, the delegate is first called with a newColumnIndex value of -1. Returning false will disallow that column from being reordered at all. Returning true allows it to be reordered, and the delegate will be called again when the column reaches a new location.

The actual NSTableColumn instance can be retrieved from the tableColumns array.

If this method is not implemented, all columns are considered reorderable.

Available in Mac OS X v10.6 and later.

9.65.124 shouldSelectRow(row as Int64) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Returns whether the table view should allow selection of the specified row.

Notes: rowIndex: The row index.

Return true to permit selection of the row, false to deny selection.

The delegate can implement this method to disallow selection of particular rows. For better performance and finer-grain control over the selection, use `selectionIndexesForProposedSelection`.

9.65.125 `shouldSelectTableColumn(tableColumn as NSTableColumnMBS) as boolean`

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Returns whether the specified table column can be selected.

Notes: `TableColumn`: The table column.

Return true to permit selection, otherwise false.

The subclass can implement this event to disallow selection of particular columns.

9.65.126 `shouldShowCellExpansion(tableColumn as NSTableColumnMBS, row as Int64) as Boolean`

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control cell expansion for a specific row and column.

Notes: `TableColumn`: The table column.

`row`: The row index.

Return true if the tooltip cell should expand, false otherwise.

Cell expansion can occur when the mouse hovers over the specified cell and the cell contents are unable to be fully displayed within the cell. If this method returns true, the full cell contents will be shown in a special floating tool tip view, otherwise the content is truncated.

Note: This method is only valid for cell-based table views.

Available in Mac OS X v10.5 and later.

9.65.127 shouldTrackCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64) as Boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control the tracking behavior for a specific cell.

Notes: cell: The cell to track.

tableColumn: The table column.

row: A row in tableView.

Returns true if the cell should track, false otherwise.

Normally, only selectable or selected cells can be tracked. If you implement this method, cells which are not selectable or selected can be tracked, and vice-versa.

For example, this allows you to have an NSButtonCell in a table which does not change the selection, but can still be clicked on and tracked.

Note: This method is only valid for cell-based table views.

Available in Mac OS X v10.5 and later.

9.65.128 shouldTypeSelectForEvent(e as NSEventMBS, searchString as string) as Boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to control type select for a specific event.

Notes: event: The event.

searchString: The search string or nil if no type select has began.

Return true to allow type select for event, false otherwise.

Typically, this is called from the table view keyDown implementation and the event will be a key event.
Available in Mac OS X v10.5 and later.

9.65.129 sizeToFitWidthOfColumn(column as Int64) as Double

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to provide custom sizing behavior when a column's resize divider is double clicked.

Notes: column: The index of the column.

Returns the width of the specified column.

By default, NSTableView iterates every row in the table, accesses a cell via preparedCellAtColumn, and requests the cellSize to find the appropriate largest width to use.

For accurate results and performance, it is recommended that this method is implemented when using large tables. By default, large tables use a monte carlo simulation instead of iterating every row.

Available in Mac OS X v10.6 and later.

9.65.130 textShouldBeginEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow.

9.65.131 textShouldEndEditing(control as NSControlMBS, fieldEditor as NSTextMBS) as boolean

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow.

9.65.132 toolTipForCell(cell as NSCellMBS, r as NSRectMBS, tableColumn as NSTableColumnMBS, row as Int64, mouseLocation as NSPointMBS) as string

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Returns a string that is displayed as a tooltip for the specified cell in the column and row.

Notes: Cell: The cell.

r: The proposed active area of the tooltip. You can modify rect to provide an alternative active area.

TableColumn: The table column.

row: The row index.

mouseLocation: The mouse location.

Return a string containing the tooltip. Return empty string if no tooltip is desired.

By default, rect is computed as `cell.drawingRectForBounds(cellFrame)`.

Available in Mac OS X v10.4 and later.

9.65.133 `typeSelectString(tableColumn as NSTableColumnMBS, row as Int64) as string`

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Invoked to allow the subclass to provide an alternate text value used for type selection for a specified row and column.

Notes: tableColumn: The table column.

row: The row index.

Returns a string that is used in type select comparison for row and tableColumn. Return "" if the row or tableColumn should not be searched.

Implement this method to change the string value that is searched for based on what is displayed. By default, all cells with text in them are searched.

If this event is not implemented the string value is the cell string value.

Implementation of this event is optional.

Available in Mac OS X v10.5 and later.

9.65.134 `view(tableColumn as NSTableColumnMBS, row as Integer) as NSViewMBS`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Asks the delegate for a view to display the specified row and column.

Notes: see also

<https://developer.apple.com/reference/appkit/nstableviewdelegate/1527449-tableview?language=objc>

9.65.135 willDisplayCell(cell as NSCellMBS, tableColumn as NSTableColumnMBS, row as Int64)

Plugin Version: 12.0, Platform: macOS, Targets: .

Function: Informs you that the tableview will display the specified cell at the row in the column.

Notes: Cell: The cell to be displayed.

TableColumn: The table column.

row: The row index.

The event can modify the display attributes of cell to alter the appearance of the cell.

Because cell is reused for every row in tableColumn, the event must set the display attributes both when drawing special cells and when drawing normal cells.

Note: The implementation of this method must not draw portions of the cell. It should only alter the state of the passed in cell.

9.65.136 Constants

Constants

Constant	Value	Description
<code>NSTableViewDashedHorizontalGridLineMask</code>	8	One of the constants to specify grid styles. Specifies that the horizontal grid lines should be drawn.
<code>NSTableViewDraggingDestinationFeedbackStyleGap</code>	2	One of the dragging styles. Provides a gap insertion when dragging over the table. Only officially supported for View Based TableViews, not Cell Based TableViews. The decision to use the gap style (or not) can be made in <code>draggingSessionWillBeginAtPosition</code> and <code>draggingSessionWillEndAtPosition</code> (which can be changed).
<code>NSTableViewDraggingDestinationFeedbackStyleNone</code>	-1	One of the constants to specify the drag styles displayed. Provides no feedback when the user drags over the table. Exists to allow subclasses to implement their dragging styles or to make it not show anything all. Available in Mac OS X v10.6 and later.
<code>NSTableViewDraggingDestinationFeedbackStyleRegular</code>	0	One of the constants to specify the drag styles displayed. Draws a solid round-rect background on drop target rows. This style should be used in most cases. Available in Mac OS X v10.6 and later.
<code>NSTableViewDraggingDestinationFeedbackStyleSourceList</code>	1	One of the constants to specify the drag styles displayed. Draws an outline on drop target rows, and an insertion point. This style will automatically be set for source lists when <code>HighlightStyle</code> is set to <code>NSTableViewSelectionHighlightStyleSourceList</code> . This is the standard look for Source Lists, but may be used in other contexts. Available in Mac OS X v10.6 and later.
<code>NSTableViewDropAbove</code>	1	One of the constants to specify drop operations. Specifies that the drop should occur above the specified row. For example, given a table with n rows (numbered 0 to $n-1$ visually), a row of $n-1$ and operation of <code>NSTableViewDropAbove</code> would drop on the last row. To specify a drop below the last row, use <code>NSTableViewDropOn</code> and <code>NSTableViewDropAbove</code> for the operation.
<code>NSTableViewDropOn</code>	0	One of the constants to specify drop operations. Specifies that the drop should occur on the specified row. For example, given a table with n rows (numbered 0 to $n-1$ visually), a row of $n-1$ and operation of <code>NSTableViewDropOn</code> would drop on the last row. To specify a drop below the last row, use <code>NSTableViewDropOn</code> and <code>NSTableViewDropAbove</code> for the operation.
<code>NSTableViewFirstColumnOnlyAutoresizingStyle</code>	5	One of the constants to specify the autoresizing styles. Autoresize only the first table column. When that table column can no longer be resized, stop resizing. Available in Mac OS X v10.4 and later.
<code>NSTableViewGridNone</code>	0	One of the constants for the grid styles. Specifies that no grid lines should be displayed.
<code>NSTableViewLastColumnOnlyAutoresizingStyle</code>	4	One of the constants to specify the autoresizing styles. Autoresize only the last table column. When that table column can no longer be resized, stop resizing. Available in Mac OS X v10.4 and later.
<code>NSTableViewNoColumnAutoresizing</code>	0	One of the constants to specify the autoresizing styles. Disable table column autoresizing. Available in Mac OS X v10.4 and later.
<code>NSTableViewReverseSequentialColumnAutoresizingStyle</code>	3	One of the constants to specify the autoresizing styles. Autoresize each table column sequentially, from the first to the last auto-resizable column; proceed to the next column has reached its minimum or maximum size. Available in Mac OS X v10.4 and later.
<code>NSTableViewSelectionHighlightStyleNone</code>	-1	One of the constants to specify the selection highlight styles. Displays no highlight style at all. Available in Mac OS X v10.6 and later.

RowActionEdge

Constant	Value	Description
NSTableRowActionEdgeLeading	0	Denotes the leading, or left, edge of an table row view.
NSTableRowActionEdgeTrailing	1	Denotes the trailing, or right, edge of an table row view.

Animation options

Constant	Value	Description
NSTableViewAnimationEffectFade	1	Use a fade for row or column removal. The effect can be combined with any the slide constants.
NSTableViewAnimationEffectGap	2	Creates a gap for newly inserted rows. This is useful for drag and drop animations that animate to a newly opened gap and should be used in the acceptDrop event.
NSTableViewAnimationEffectNone	0	Use no animation effects.
NSTableViewAnimationSlideDown	&h20	Animates a row insertion or removal by sliding downward.
NSTableViewAnimationSlideLeft	&h30	Animates a row insertion by sliding from the left. Animates a row removal by sliding towards the left.
NSTableViewAnimationSlideRight	&h40	Animates a row insertion by sliding from the right. Animates a row removal by sliding towards the right.
NSTableViewAnimationSlideUp	&h10	Animates a row insertion or removal by sliding upward.

Row Size Style Constants

Constant	Value	Description
<code>NSTableViewRowSizeStyleCustom</code>	0	The table will use the <code>rowHeight</code> or invoke the delegate method <code>tableView:heightOfRow:</code> , if implemented. The cell layout is not changed. Available in OS X v10.7 and later.
<code>NSTableViewRowSizeStyleDefault</code>	-1	The row size style constants define the size of the rows in the table view. They are used by the <code>effectiveRowSizeStyle</code> and <code>rowSizeStyle</code> methods. You can also query the row size in the <code>NSTableCellView</code> class' property <code>rowSizeStyle</code> . The table will use the system default layout size: small, medium or large. Available in OS X v10.7 and later.
<code>NSTableViewRowSizeStyleLarge</code>	3	The row size style constants define the size of the rows in the table view. They are used by the <code>effectiveRowSizeStyle</code> and <code>rowSizeStyle</code> methods. You can also query the row size in the <code>NSTableCellView</code> class' property <code>rowSizeStyle</code> . The table will use a row height specified for a small table. It is required that the size be fully tested and supported if <code>NSTableViewRowSizeStyleCustom</code> is not used. Available in OS X v10.7 and later.
<code>NSTableViewRowSizeStyleMedium</code>	2	The row size style constants define the size of the rows in the table view. They are used by the <code>effectiveRowSizeStyle</code> and <code>rowSizeStyle</code> methods. You can also query the row size in the <code>NSTableCellView</code> class' property <code>rowSizeStyle</code> . The table will use a row height specified for a medium table. It is required that the size be fully tested and supported if <code>NSTableViewRowSizeStyleCustom</code> is not used. Available in OS X v10.7 and later.
<code>NSTableViewRowSizeStyleSmall</code>	1	The row size style constants define the size of the rows in the table view. They are used by the <code>effectiveRowSizeStyle</code> and <code>rowSizeStyle</code> methods. You can also query the row size in the <code>NSTableCellView</code> class' property <code>rowSizeStyle</code> . The table will use a row height specified for a small table. It is required that the size be fully tested and supported if <code>NSTableViewRowSizeStyleCustom</code> is not used. Available in OS X v10.7 and later.

Styles

Constant	Value	Description
NSTableViewStyleAutomatic	0	Automatically infers the effectiveStyle from the table view hierarchy.
NSTableViewStyleFullWidth	1	Edge-to-edge style with standard content padding at the ends of each row. This content padding is constant and independent of intercellSpacing.
NSTableViewStyleInset	2	Inset style with rounded corners selection
NSTableViewStylePlain	4	A plain style. No insets, padding or any other kind of decoration applied to the row or its background. The cells are equally spaced in the row using intercellSpacing.width.
NSTableViewStyleSourceList	3	<p>The source list style of NSTableView. Setting this style will have the side effect of setting the background color to "source list".</p> <p>Additionally in NSOutlineView, the following properties may change to get the standard "source list" look: indentationPerLevel, rowHeight and intercellSpacing. After setting the style it is possible to change any of the other properties as required.</p> <p>In 10.11, if the background color has been changed from the "source list" background color to something else, the table will no longer draw the selection as a source list blur style, and instead will draw a normal selection highlight. This replaces NSTableViewSelectionHighlightStyleSourceList which is to be deprecated.</p>

9.66 class NSTableViewRowActionMBS

9.66.1 class NSTableViewRowActionMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: An NSTableViewRowAction object defines a single action to present when the user swipes horizontally on a table row.

Notes: In an editable table, performing a horizontal swipe on a row reveals a button to delete the row by default. This class lets you define one or more custom actions to display for a given row in your table. Each instance of this class represents a single action to perform and includes the text, formatting information, and behavior for the corresponding button.

To add custom actions to your table view,Äôs rows, implement the rowActionsForRow event in your table view,Äôs delegate object. In that method, create and return an array of actions for the specified row. The table handles the remaining work of displaying the action buttons and executing the appropriate handler block when the user clicks the button.

Blog Entries

- [MBS Xojo Plugins, version 17.1pr3](#)

9.66.2 Methods

9.66.3 available as boolean

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Check whether this class is available.

Notes: Returns true on macOS 10.11.

9.66.4 Constructor(Style as Integer, Title as String)

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a new table view row action object.

Notes: style: The style characteristics to apply to the button. Use this value to apply default appearance characteristics to the button. These characteristics visually communicate, such as by color, information about what the button does. For example, specify a style of NSTableViewRowActionStyleDestructive to indicate an action is destructive to the underlying data. For a list of possible style values, see NSTableViewRowActionStyle.

title: The string to display in the button. Specify a string localized for the user,Äôs current language.

Returns a new table row action object that you can return from your table view,Äôs event.

The style you specify cannot be changed later. You can change the title of the action button. You can also configure other appearance-related properties of the button using the properties of this class. You can assign the same row action object to multiple rows of your table.

9.66.5 Properties

9.66.6 BackgroundColor as NSColorMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The background color of the action button.

Notes: Use this property to specify the background color for your button. If you do not specify a value for this property, AppKit assigns a default color based on the value in the style property. Generally, this color is red for destructive actions and blue for nondestructive actions.
(Read and Write property)

9.66.7 Handle as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The internal object reference.

Notes: (Read and Write property)

9.66.8 Image as NSImageMBS

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The image.

Notes: Available in macOS 10.12 or later.
(Read and Write property)

9.66.9 Style as Integer

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The style applied to the action button.

Notes: The value of this property is set at creation time and cannot be changed later.
(Read only property)

9.66.10 Title as String

Plugin Version: 17.1, Platform: macOS, Targets: Desktop only.

Function: The title of the action button.

Notes: (Read and Write property)

9.66.11 Events

9.66.12 Action(row as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when action is invoked.

Notes: The event to execute when the user clicks the button associated with this action. When the user selects the action represented by this object, AppKit executes your event on the app's main thread.

9.66.13 Constants

Styles

Constant	Value	Description
<code>NSTableViewRowActionStyleDestructive</code>	1	Apply a style that indicates that the action might change or delete data. This style changes the value of the <code>backgroundColor</code> property to an appropriate value to reflect the destructive action. After creating the action object, you can change the background color as needed. Destructive actions require a swipe to activate, and trigger an animation when a table row is deleted.
<code>NSTableViewRowActionStyleRegular</code>	0	Apply the default style to the button. This style does not apply any special coloring to the button.

9.67 control NSTokenFieldControlMBS

9.67.1 control NSTokenFieldControlMBS

Plugin Version: 15.4, Platform: macOS, Targets: Desktop only.

Function: The control to host a NSTextField.

Blog Entries

- [MBS Xojo Plugins, version 24.1pr5](#)
- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo Plugins, version 21.4pr1](#)
- [MBS Xojo Plugins, version 21.1pr2](#)

9.67.2 Properties

9.67.3 View as NSTextFieldMBS

Plugin Version: 15.4, Platform: macOS, Targets: Desktop only.

Function: The view used for the control.

Notes: (Read only property)

9.67.4 Events

9.67.5 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

9.67.6 Close

Plugin Version: 15.4, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

9.67.7 completionsForSubstring(substring as string, tokenIndex as Integer, byref selectedIndex as Integer) as Variant()

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to provide an array of appropriate completions for the contents of the receiver.

Notes: substring: The partial string that is to be completed.

tokenIndex: The index of the token being edited.

selectedIndex: Optionally, you can return by-reference an index into the returned array that specifies which of the completions should be initially selected. If none are to be selected, return by reference -1.

Returns an array of strings that are possible completions.

If the delegate does not implement this method, no completions are provided.

Available in OS X v10.4 and later.

9.67.8 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.67.9 ContextualMenuItemAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.67.10 didCloseContextualMenu(menu as NSMenuItem, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

9.67.11 displayStringForRepresentedObject(representedObject as Variant) as string

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to provide a string to be displayed as a proxy for the given represented object.

Notes: representedObject: A represented object of the token field.

Returns the string to be used as a proxy for representedObject. If you return nil or do not implement this method, then representedObject is displayed as the string.

9.67.12 editingStringForRepresentedObject(representedObject as Variant) as string

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to provide a string to be edited as a proxy for a represented object.

Notes: representedObject: A represented object of the token field.

Returns a string that's an editable proxy of the represented object, or nil if the token should not be editable.

9.67.13 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named MenuBarSelected.

9.67.14 FrameChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

9.67.15 GotFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named FocusReceived.

Notes:

This only fires if the control itself got focus and not a sub control.

9.67.16 hasMenuForRepresentedObject(representedObject as Variant) as boolean

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to specify whether the given represented object provides a menu.

Notes: representedObject: A represented object of the token field.

Returns true if the represented object has a menu, false otherwise.

By default tokens in a token field have no menus.

9.67.17 LostFocus

Plugin Version: 16.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named FocusLost.

Notes:

This only fires if the control itself lost focus and not a sub control.

9.67.18 menuForRepresentedObject(representedObject as Variant) as NSMenuMBS

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to provide a menu for the specified represented object.

Notes: representedObject: A represented object of the token field.

Returns the menu associated with the represented object.

By default tokens in a token field do not return menus.

9.67.19MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control,Ã region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

9.67.20 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of times per second), it is your responsibility to determine if the mouse has really moved.

9.67.21 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

9.67.22 Open

Plugin Version: 15.4, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.

In Xojo version 2021r3 and newer this event is named Opening.

9.67.23 readFromPasteboard(pboard as NSPasteboardMBS) as Variant()

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to return an array of objects representing the data read from the specified pasteboard.

Notes: pboard: The pasteboard from which to read the represented objects.

Returns an array of represented objects created from the pasteboard data.

9.67.24 representedObjectForEditingString(editingString as string) as Variant

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to provide a represented object for the given editing string.

Notes: editingString: The edited string representation of a represented object.

Returns a represented object that is displayed rather than the editing string.

Note: In OS X v10.4, NSTokenField trims whitespace around tokens but it does not trim whitespace in OS X versions 10.5.0 and 10.5.1. In OS X v10.5.2, you get whitespace-trimming behavior by either linking against the v10.4 binary or linking against the v10.5 binary and not implementing the this method. If you do not want the whitespace-trimming behavior, link against the v10.5 binary and implement this method, returning the editing string if you have no represented object.

9.67.25 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.67.26 shouldAddObjects(tokens() as Variant, index as Integer) as Variant()

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows to validate the tokens to be added to the receiver at a particular location.

Notes: tokens: An array of tokens to be inserted in the receiver at index.

index: The index of the receiver in which the array of tokens to be validated (tokens) will be inserted.

Returns an array of validated tokens.

The event can return the array unchanged or return a modified array of tokens. To reject the add completely, return an empty array. Returning nil causes an error.

9.67.27 styleForRepresentedObject(representedObject as Variant) as Integer

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Allows the delegate to return the token style for editing the specified represented object.

Notes: representedObject: A represented object of the token field.

Returns the style that should be used to display the representedObject. Possible values are shown in NSTokenStyle Values.

If the event implements this method and returns an NSTokenStyle that differs from the style set by setTokenStyle:, the value the event returns is preferred.

If you don't implement this method, the token field's tokenStyle is used.

9.67.28 TextDidBeginEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Sent when a control with editable text begins an editing session.

Notes: Notification: The notification object. The name of the notification is always NSControlTextDidBeginEditingNotification.

This event is invoked when the user begins editing text in a control such as a text field or a form field. The control posts a NSControlTextDidBeginEditingNotification notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is also delivered for inspection.

See TextDidEndEditing for an explanation of why you may not always get one invocation of TextDidBeginEditing for each invocation of TextDidEndEditing.

9.67.29 `TextDidChange(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Sent when the text in the receiving control changes.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidChangeNotification`.

This event is invoked when text in a control such as a text field or form changes. The control posts a `NSControlTextDidChangeNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is provided as parameter for inspection.

9.67.30 `TextDidEndEditing(fieldEditor as NSTextMBS, notification as NSNotificationMBS)`

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Sent when a control with editable text ends an editing session.

Notes: Notification: The notification object. The name of the notification is always `NSControlTextDidEndEditingNotification`.

This event is invoked when the user stops editing text in a control such as a text field or form. The control posts a `NSControlTextDidEndEditingNotification` notification, and if the control's subclass implements this event, it is automatically registered to receive the notification. The field editor is also provided for inspection.

Warning: In some cases, such as when editing within an instance of `NSOutlineView`, this method may be invoked without a previous invocation of `TextDidBeginEditing`. You will only get the `TextDidBeginEditing` notification if the user actually types something, but you can get the `TextDidEndEditing` notification if the user just double-clicks the field and then clicks outside the field, without typing.

9.67.31 `textShouldBeginEditing(fieldEditor as NSTextMBS) as boolean`

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.67.32 textShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow end of text editing or false to deny.

Be aware that an event in Xojo without return will cause false to be returned.

9.67.33 tokenFieldAction

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: The control's action was triggered.

Notes: For a button if it was pressed.

9.67.34 tokenFieldTextShouldBeginEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: The event called to decide whether text editing should be allowed.

Notes: Return true to allow text editing.

9.67.35 tokenFieldTextShouldEndEditing(fieldEditor as NSTextMBS) as boolean

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: The event called to decide whether ending text editing should be allowed.

Notes: Return true to allow text editing.

9.67.36 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

9.67.37 writeRepresentedObjects(objects() as Variant, pboard as NSPasteboardMBS) as boolean

Plugin Version: 15.4, Platform: macOS, Targets: .

Function: Sent so the delegate can write represented objects to the pasteboard corresponding to a given array of display strings.

Notes: objects: An array of represented objects associated with the token field.

pboard: The pasteboard to which to write the represented objects.

Return true if you writes the represented objects to the pasteboard, false otherwise. If false, the token field writes the display strings to the NSStringPboardType pasteboard.

9.68 control NSViewControlMBS

9.68.1 control NSViewControlMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The control to host a generic NSView.

Blog Entries

- [MBS Xojo Plugins, version 20.5pr6](#)
- [MonkeyBread Software Releases the MBS Xojo Plugins in version 18.2](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr10](#)

9.68.2 Properties

9.68.3 View as NSViewMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The view used in the control.

Notes: Use this object to set more options on the control.

(Read only property)

9.68.4 Events

9.68.5 acceptsFirstMouse(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Overridden by subclasses to return true if the receiver should be sent a mouseDown event for an initial mouse-down event, false if not.

Notes: The receiver can either return a value unconditionally or use the location of event e to determine whether or not it wants the event. The default implementation ignores the event and returns false.

Implement this event in a subclass to allow instances to respond to click-through. This allows the user to click on a view in an inactive window, activating the view with one click, instead of clicking first to make the window active and then clicking the view. Most view objects refuse a click-through attempt, so the event simply activates the window. Many control objects, however, such as instances of NSButton and NSSlider, do accept them, so the user can immediately manipulate the control without having to release the mouse button.

9.68.6 `acceptsFirstResponder` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Whether to accept first responder.

Notes: Return true if your control can have the focus and false if not.

9.68.7 `becomeFirstResponder` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Called when the object gets focus.

Notes: Return true to accept.

9.68.8 `beginGestureWithEvent(e as NSEventMBS)` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a touch gesture.

Notes: e: An event object representing the gesture beginning.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.68.9 `canBecomeKeyView` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Returns whether the receiver can become key view.

Notes: Returns true if the receiver can become key view, false otherwise.

9.68.10 `Close`

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The control is about to close.

9.68.11 `Closed`

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The event called when the custom view is destroyed.

Notes: Renamed to Closed in plugin version 20.5. Used to be named Close.

9.68.12 concludeDragOperation(sender as NSDraggingInfoMBS)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the dragging operation is complete, signaling the receiver to perform any necessary clean-up.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

For this method to be invoked, the previous performDragOperation must have returned true.

The destination implements this method to perform any tidying up that it needs to do, such as updating its visual representation now that it has incorporated the dragged data. This message is the last message sent from sender to the destination during a dragging session.

If the sender object's animatesToDestination property was set to true in prepareForDragOperation, then the drag image is still visible. At this point you should draw the final visual representation in the view. When this method returns, the drag image is removed from the screen. If your final visual representation matches the visual representation in the drag, this is a seamless transition.

9.68.13 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

9.68.14 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

9.68.15 draggingEnded(sender as NSDraggingInfoMBS)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Implement this event to be notified when a drag operation ends in some other destination.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

This method might be used by a destination doing auto-expansion in order to collapse any auto-expands.

9.68.16 draggingEntered(sender as NSDraggingInfoMBS) as Integer

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the dragged image enters destination bounds or frame; delegate returns dragging operation to perform.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in NSDragOperation in the NSDraggingInfo reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous draggingEntered message.

Invoked when a dragged image enters the destination but only if the destination has registered for the pasteboard data type involved in the drag operation. Specifically, this method is invoked when the mouse pointer enters the destination's bounds rectangle (if it is a view object) or its frame rectangle (if it is a window object).

This method must return a value that indicates which dragging operation the destination will perform when the image is released. In deciding which dragging operation to return, the method should evaluate the overlap between both the dragging operations allowed by the source (obtained from sender with the draggingSourceOperationMask method) and the dragging operations and pasteboard data types the destination itself supports.

If none of the operations is appropriate, this method should return NSDragOperationNone (this is the default response if the method is not implemented by the destination). A destination will still receive draggingUpdated and draggingExited even if NSDragOperationNone is returned by this method.

9.68.17 draggingExited(sender as NSDraggingInfoMBS)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the dragged image exits the destination's bounds rectangle (in the case of a view object) or its frame rectangle (in the case of a window object).

Notes: sender: The object sending the message; use it to get details about the dragging operation.

9.68.18 draggingSessionEndedAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS, operation as Integer)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the dragging session has completed.

Notes: session: The dragging session.

screenPoint: The point where the drag ended, in screen coordinates.

operation: The drag operation. See constants for drag operation types.

Available in OS X v10.7 and later.

9.68.19 draggingSessionMovedToPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the drag moves on the screen.

Notes: session: The dragging session.

screenPoint: The point where the drag moved to, in screen coordinates.

Available in OS X v10.7 and later.

9.68.20 draggingSessionSourceOperationMaskForDraggingContext(session as NSDraggingSessionMBS, context as Integer) as Integer

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Declares the types of operations the source allows to be performed. (required)

Notes: session: The dragging session.

context: The dragging context. See NSDraggingContext constants for the supported values.

Return the appropriate dragging operation as defined in constants.

In the future Apple may provide more specific "within" values in the future. To account for this, for unrecognized localities, return the operation mask for the most specific context that you are concerned with.

9.68.21 `draggingSessionWillBeginAtPoint(session as NSDraggingSessionMBS, screenPoint as NSPointMBS)`

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the drag will begin.

Notes: `session`: The dragging session.

`screenPoint`: The point where the drag will begin, in screen coordinates.

Available in OS X v10.7 and later.

9.68.22 `draggingUpdated(sender as NSDraggingInfoMBS) as Integer`

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked periodically as the image is held within the destination area, allowing modification of the dragging operation or mouse-pointer position.

Notes: `sender`: The object sending the message; use it to get details about the dragging operation.

Return one (and only one) of the dragging operation constants described in `NSDragOperation` in the `NSDraggingInfo` reference. The default return value (if this method is not implemented by the destination) is the value returned by the previous `draggingEntered` message.

For this to be invoked, the destination must have registered for the pasteboard data type involved in the drag operation. The messages continue until the image is either released or dragged out of the window or view.

This method provides the destination with an opportunity to modify the dragging operation depending on the position of the mouse pointer inside of the destination view or window object. For example, you may have several graphics or areas of text contained within the same view and wish to tailor the dragging operation, or to ignore the drag event completely, depending upon which object is underneath the mouse pointer at the time when the user releases the dragged image and the `performDragOperation` method is invoked.

You typically examine the contents of the pasteboard in the `draggingEntered` method, where this examination is performed only once, rather than in the `draggingUpdated` method, which is invoked multiple times.

Only one destination at a time receives a sequence of `draggingUpdated` messages. If the mouse pointer is within the bounds of two overlapping views that are both valid destinations, the uppermost view receives these messages until the image is either released or dragged out.

9.68.23 drawFocusRingMask(g as NSGraphicsMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Draw the focus ring mask for the view.

Notes: If false is returned, the default method from NSView class runs.

This method provides the shape of the focus ring mask by drawing the focus ring mask. An implementation of this method should draw in the view's interior (bounds) coordinate space, that the focus ring style has been set (it will be set it to NSFocusRingOnly to capture the focus ring itself), and that the fill and stroke colors have been set to an arbitrary fully opaque color.

Subclasses that find the default behavior insufficient should only draw the focus ring shape.

The NSView default implementation of this method simply fills self.bounds.
Available in Mac OS X v10.7 and later.

Please use NSGraphicsMBS class for drawing.

9.68.24 DrawRect(g as NSGraphicsMBS, left as Double, top as Double, width as Double, height as Double)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The paint event with the rectangle which needs to be redrawn.

9.68.25 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event where you can enable menu items.

9.68.26 endGestureWithEvent(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the receiver that the user has ended a touch gesture.

Notes: e: An event object representing the gesture end.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.68.27 focusRingMaskBounds as NSRectMBS

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Returns the focus ring mask bounds.

Notes: Return nil to run default NSView method.

Return a rectangle containing the mask in the view's interior (bounds) coordinate space.

The mask bounds allows the focus ring's overall size and position to be determined before it is drawn.

Subclasses must override this method if they require the display of a focus ring.

The NSView default implementation of this method simply returns NSRectMBS.Zero.

Note: The information provided by focusRingMaskBounds will enable Accessibility to identify selected subelements for zoom tracking, so it is important that this method provide a reasonably tight bounding box and that noteFocusRingMaskChanged is invoked as described.

9.68.28 ignoreModifierKeysForDraggingSession(session as NSDraggingSession-MBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Returns whether the modifier keys will be ignored for this dragging session.

Notes: session: The dragging session.

Return true if the modifier keys will be ignored, false otherwise.

Available in OS X v10.7 and later.

9.68.29 isOpaque as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Whether this view is opaque.

9.68.30 keyDown(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

Please return true in becomeFirstResponder and acceptsFirstResponder, so your nsvew can become first

responder and receive key events.

9.68.31 keyUp(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the key events.

Notes: Return true if you handled this event.

Please return true in becomeFirstResponder and acceptsFirstResponder, so your nview can become first responder and receive key events.

9.68.32 magnifyWithEvent(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a pinch gesture.

Notes: e: An event object representing the magnify gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.68.33 menuForEvent(e as NSEventMBS, defaultMenu as NSMenuMBS) as NSMenuMBS

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Overridden by subclasses to return a context-sensitive pop-up menu for a given mouse-down event.

Notes: theEvent: An object representing a mouse-down event.

defaultMenu: The menu as constructed by super class.

The receiver can use information in the mouse event, such as its location over a particular element of the receiver, to determine what kind of menu to return. For example, a text object might display a text-editing menu when the cursor lies over text and a menu for changing graphics attributes when the cursor lies over an embedded image.

The default implementation returns the default menu.

9.68.34 `mouseDown(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.35 `mouseDownCanMoveWindow` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: This event is called so you can decide what happens with mouse down.

Notes: Return true if you do not need to handle a mouse down and it can pass through to superviews; False if you need to handle the mouse down.

This allows iApp-type applications to determine the region by which a window can be moved. By default, this method returns false if the view is opaque; otherwise, it returns true. Subclasses can override this method to return a different value.

9.68.36 `mouseDragged(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.37 `mouseEntered(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.38 `mouseExited(e as NSEventMBS, x as Double, y as Double)` as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.39 mouseMoved(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.40 mouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.41 Open

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The control is about to was created and you can initialize it.

9.68.42 Opened

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: The event called when the custom NSView is created.

Notes: Renamed to Opened in plugin version 20.5. Used to be named Open.

9.68.43 otherMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.68.44 otherMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.68.45 otherMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

Third mouse button.

9.68.46 performDragOperation(sender as NSDraggingInfoMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked after the released image has been removed from the screen, signaling the receiver to import the pasteboard data.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Return if the destination accepts the data, it returns true; otherwise it returns false. The default is to return false.

For this method to be invoked, the previous `prepareForDragOperation` message must have returned true. The destination should implement this method to do the real work of importing the pasteboard data represented by the image.

If the sender object's `animatesToDestination` was set to true in `prepareForDragOperation`, then setup any animation to arrange space for the drag items to animate to. Also at this time, enumerate through the dragging items to set their destination frames and destination images.

9.68.47 prepareForDragOperation(sender as NSDraggingInfoMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the image is released, allowing the receiver to agree to or refuse drag operation.

Notes: sender: The object sending the message; use it to get details about the dragging operation.

Returns true if the receiver agrees to perform the drag operation and false if not.

This method is invoked only if the most recent draggingEntered or draggingUpdated message returned an acceptable drag-operation value.

If you want the drag items to animate from their current location on screen to their final location in your view, set the sender object's animatesToDestination property to true in your implementation of this method.

9.68.48 pressureChange(e as NSEventMBS) as boolean

Plugin Version: 15.1, Platform: macOS, Targets: .

Function: Informs the current object that a pressure change occurred on a system that supports pressure sensitivity.

Notes: This method is invoked automatically in response to user actions. event is the event that initiated the change in pressure.

Available in OS X v10.10.3 and later.

9.68.49 resignFirstResponder as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Focus is going away.

Notes: Return true to accept.

9.68.50 rightMouseDown(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.51 rightMouseDragged(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.52 rightMouseUp(e as NSEventMBS, x as Double, y as Double) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: One of the mouse events.

Notes: Return true if you handled this event.

9.68.53 rotateWithEvent(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a rotation gesture.

Notes: e: An event object representing the rotate gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.68.54 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

9.68.55 scrollWheel(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the subclass that the mouse's scroll wheel has moved.

Notes: e: An object encapsulating information about the wheel-scrolling event.

The default implementation simply passes this message to the next responder.

Return true to not pass the event.

9.68.56 swipeWithEvent(e as NSEventMBS) as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the receiver that the user has begun a swipe gesture.

Notes: e: An event object representing the swipe gesture.

The event will be sent to the view under the touch in the key window.

Available in Mac OS X v10.6 and later.

Return true if you handled this event.

9.68.57 updateDraggingItemsForDrag(sender as NSDraggingInfoMBS)

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Invoked when the dragging images should be changed.

Notes: sender: The object sending the message; use this object to get details about the dragging operation.

While a destination may change the dragging images at any time, it is recommended to wait until this method is called before updating the dragging images.

This allows the system to delay changing the dragging images until it is likely that the user will drop on this destination. Otherwise, the dragging images will change too often during the drag which would be distracting to the user.

9.68.58 viewDidMoveToWindow

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Informs the receiver that it has been added to a new view hierarchy.

Notes: The default implementation does nothing; subclasses can implement this event to perform whatever actions are necessary.

Window may return nil when this method is invoked, indicating that the receiver does not currently reside in any window. This occurs when the receiver has just been removed from its superview or when the receiver has just been added to a superview that does not itself have a window. Overrides of this method may choose to ignore such cases if they are not of interest.

9.68.59 wantsPeriodicDraggingUpdates as boolean

Plugin Version: 15.0, Platform: macOS, Targets: .

Function: Asks the destination object whether it wants to receive periodic draggingUpdated messages.

Notes: Return true if the destination wants to receive periodic draggingUpdated messages, false otherwise.

If the destination returns false, these messages are sent only when the mouse moves or a modifier flag changes. Otherwise the destination gets the default behavior, where it receives periodic draggingUpdated messages even if nothing changes.

9.69 class PopupMenu

9.69.1 class PopupMenu

Platform: macOS, Targets: Desktop only.

Function: Gives access to the NSMenuMBS of the PopupMenu.

9.69.2 Methods

9.69.3 NSButtonMBS as NSButtonMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox PopupMenu1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.69.4 NSPopUpButtonMBS as NSPopUpButtonMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSPopUpButtonMBS object for the given control.

Example:

```
// get cocoa view for the popupmenu
dim p as NSPopUpButtonMBS = PopupMenu1.NSPopUpButtonMBS

// find a menu entry
dim it as NSMenuItemMBS = p.itemAtIndex(0)

// get a picture
dim pic as Picture = LogoMBS(500)
dim img as new NSImageMBS(pic)
img.setSize 16,16

// and assign icon
it.image = img
```

Notes: This way you can manipulate Cocoa controls directly.

9.70 class ProgressWheel

9.70.1 class ProgressWheel

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The built in ProgressWheel class in Xojo.

9.70.2 Methods

9.70.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSProgressIndicatorMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.71 class PushButton

9.71.1 class PushButton

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: The built in Button class in Xojo.

9.71.2 Methods

9.71.3 NSButtonMBS as NSButtonMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox PushButton1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.72 class Radiobutton

9.72.1 class Radiobutton

Plugin Version: 2.9, Platform: macOS, Targets: Desktop only.

Function: An extension of Xojo's internal control.

9.72.2 Methods

9.72.3 NSButtonMBS as NSButtonMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSButtonMBS object for the given control.

Example:

```
MsgBox RadioButton1.NSButtonMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.73 class ScrollBar

9.73.1 class ScrollBar

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: The built in Scrollbar class in Xojo.

9.73.2 Methods

9.73.3 NSScrollerMBS as NSScrollerMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSScrollerMBS object for the given control.

Example:

```
MsgBox ScrollBar1.NSScrollerMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.74 class Separator

9.74.1 class Separator

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The built in Separator class in Xojo.

9.74.2 Methods

9.74.3 NSBoxMBS as NSBoxMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSBoxMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

9.75 class Slider

9.75.1 class Slider

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: The built in slider class in Xojo.

9.75.2 Methods

9.75.3 NSSliderMBS as NSSliderMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSSliderMBS object for the given control.

Example:

```
MsgBox Slider1.NSSliderMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.76 class TextArea

9.76.1 class TextArea

Plugin Version: 9.7, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in textarea class in Xojo.

Example:

```
// make a PDF from a textarea in Cocoa Xojo target:

// find view
dim n as NSViewMBS = TextArea1.NSViewMBS
if n = nil then
  MsgBox "Only in Cocoa!"
  Return
end if

// make pdf data
dim s as string = n.dataWithPDFInsideRect(0,0,n.frame.Width, n.frame.Height)

// save
dim f as FolderItem = GetSaveFolderItem("", "test.pdf")

if f<>Nil then

  dim b as BinaryStream = BinaryStream.Create(f, true)

  b.Write s

end if
```

Notes: Requires RB 2009r4 or newer.

9.76.2 Methods

9.76.3 NSScrollViewMBS as NSScrollViewMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSScrollViewMBS object for the given control.

Example:

```
MsgBox TextArea1.NSScrollViewMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

9.77 class UpDownArrows

9.77.1 class UpDownArrows

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: The build in control class in Xojo.

9.77.2 Methods

9.77.3 NSStepperMBS as NSStepperMBS

Plugin Version: 10.0, Platform: macOS, Targets: Desktop only.

Function: Creates a NSStepperMBS object for the given control.

Notes: This way you can manipulate Cocoa controls directly.

Chapter 10

Cocoa Networking

10.1 class NSHTTPCookieMBS

10.1.1 class NSHTTPCookieMBS

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: An NSHTTPCookie object represents an HTTP cookie.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// we collect domain names
dim domains(-1) as string
for each cookie as NSHTTPCookieMBS in cookies
if domains.IndexOf(cookie.domain) <0 then
domains.Append cookie.domain
end if
next

// and display them
MsgBox "You have cookies from this domains: "+Join(domains, ", ")
```

Notes: It's an immutable object initialized from a dictionary containing the cookie attributes.

Two versions of cookies are supported:

Version 0: This version refers to "traditional" or "old-style" cookies, the original cookie format defined by

Netscape. The majority of cookies encountered are in this format.

Version 1: This version refers to cookies as defined in RFC 2965, HTTP State Management Mechanism.

Blog Entries

- [MBS Xojo Plugins, version 20.6pr1](#)
- [MBS Xojo Plugins, version 20.5pr7](#)
- [News from the MBS Xojo Plugins Version 20.1](#)
- [Cookies in HTMLViewer](#)
- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr3](#)

10.1.2 Methods

10.1.3 Constructor(properties as dictionary)

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Creates an initialized NSHTTPCookie object using the provided properties.

Example:

```
// create dictionary with properties:
dim prop as new Dictionary
dim d as new date
d.Year = d.Year + 1

prop.Value(NSHTTPCookieMBS.NSHTTPCookieVersion)="0"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieName)="test"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieValue)="some value"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieOriginURL)="http://www.mbsplugins.de/"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieDomain)="www.mbsplugins.de"
prop.Value(NSHTTPCookieMBS.NSHTTPCookiePath)="/"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieExpires)=d

// create cookie from properties
dim cookie as new NSHTTPCookieMBS(prop)

// and display properties
dim dic as Dictionary = cookie.properties
dim list(-1) as string

for each key as Variant in dic.keys
List.Append key.StringValue+": "+dic.Value(key).StringValue
next
```

MsgBox Join(list,EndOfLine)

Notes: properties: The properties for the new cookie object, expressed as key value pairs.

Handle is non zero on success.

10.1.4 cookiesWithResponseHeaderFields(headerFields as dictionary, URL as string) as NSHTTPCookieMBS()

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array of NSHTTPCookie objects corresponding to the provided response header fields for the provided URL.

Notes: headerFields: The header fields used to create the NSHTTPCookie objects.

URL: The URL associated with the created cookies.

Returns the array of created cookies.

This method ignores irrelevant header fields in headerFields, allowing dictionaries to contain additional data.

If headerFields does not specify a domain for a given cookie, the cookie is created with a default domain value of theURL.

If headerFields does not specify a path for a given cookie, the cookie is created with a default path value of "/".

10.1.5 cookieWithProperties(dic as dictionary) as NSHTTPCookieMBS

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Creates and initializes an NSHTTPCookie object using the provided properties.

Example:

```
// create dictionary with properties:
```

```
dim prop as new Dictionary
```

```
dim d as new date
```

```
d.Year = d.Year + 1
```

```
prop.Value(NSHTTPCookieMBS.NSHTTPCookieVersion)="0"
```

```
prop.Value(NSHTTPCookieMBS.NSHTTPCookieName)="test"
```

```
prop.Value(NSHTTPCookieMBS.NSHTTPCookieValue)="some value"
```

```

prop.Value(NSHTTPCookieMBS.NSHTTPCookieOriginURL)="http://www.mbsplugins.de/"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieDomain)="www.mbsplugins.de"
prop.Value(NSHTTPCookieMBS.NSHTTPCookiePath)="/"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieExpires)=d

// create cookie from properties
dim cookie as NSHTTPCookieMBS = NSHTTPCookieMBS.cookieWithProperties(prop)

// and display properties
dim dic as Dictionary = cookie.properties
dim list(-1) as string

for each key as Variant in dic.keys
List.Append key.StringValue+": "+dic.Value(key).StringValue
next

MsgBox Join(list,EndOfLine)

```

Notes: dic: The properties for the new cookie object, expressed as key value pairs.

Returns the newly created cookie object. Returns nil if the provided properties are invalid.

See NSHTTPCookie* shared method for more information on the available header field constants and the constraints imposed on the values in the dictionary.

10.1.6 NSHTTPCookieComment as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing the comment for the cookie.

Only valid for Version 1 cookies and later. This header field is optional.

10.1.7 NSHTTPCookieCommentURL as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing the comment URL for the cookie.

Only valid for Version 1 cookies or later. This header field is optional.

10.1.8 NSHTTPCookieDiscard as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String stating whether the cookie should be discarded at the end of the session.

String value must be either "TRUE" or "FALSE". This header field is optional. Default is "FALSE", unless this is cookie is version 1 or greater and a value for NSHTTPCookieMaximumAge is not specified, in which case it is assumed "TRUE".

10.1.9 NSHTTPCookieDomain as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing the domain for the cookie.

A value must be specified for either NSHTTPCookieDomain or NSHTTPCookieOriginURL. If this header field is missing the domain is inferred from the value for NSHTTPCookieOriginURL.

10.1.10 NSHTTPCookieExpires as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An Date object or String specifying the expiration date for the cookie.

This header field is only used for Version 0 cookies. This header field is optional.

10.1.11 NSHTTPCookieMaximumAge as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing an integer value stating how long in seconds the cookie should be kept, at most.

Only valid for Version 1 cookies and later. Default is "0". This field is optional.

10.1.12 NSHTTPCookieName as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String object containing the name of the cookie. This field is required.

10.1.13 NSHTTPCookieOriginURL as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing the URL that set this cookie.

A value must be specified for either NSHTTPCookieDomain or NSHTTPCookieOriginURL.

10.1.14 NSHTTPCookiePath as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing the path for the cookie. This field is required if you are using the NSHTTPCookieDomain key instead of the NSHTTPCookieOriginURL key.

If you are using the NSHTTPCookieOriginURL key, the path is inferred if it is not provided. The default value is `"/`.

10.1.15 NSHTTPCookiePort as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing comma-separated integer values specifying the ports for the cookie.

Only valid for Version 1 cookies or later. The default value is an empty string (`""`). This header field is optional.

10.1.16 NSHTTPCookieSecure as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String indicating that the cookie should be transmitted only over secure channels.

Providing any value for this key indicates that the cookie should remain secure.

10.1.17 NSHTTPCookieValue as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String containing the value of the cookie.

This header field is required.

10.1.18 NSHTTPCookieVersion as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the dictionary constants for cookies.

Notes: An String that specifies the version of the cookie.

Must be either "0" or "1". The default is "0". This header field is optional.

10.1.19 portList as Integer()

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's port list.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display port list

dim PortList(-1) as Integer = cookie.portList
if UBound(PortList)=-1 then
MsgBox "Port List: all ports."
else
dim list(-1) as string
for each port as Integer in portList
List.Append str(port)
next

MsgBox "Port List: "+Join(list, ", ")
end if
```

Notes: The list of ports for the cookie, returned as an array of integers. If the cookie has no port list this method returns nil and the cookie will be sent to any port. Otherwise, the cookie is only sent to ports specified in the port list.

10.1.20 requestHeaderFieldsWithCookies(cookies() as NSHTTPCookieMBS) as dictionary

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a dictionary of header fields corresponding to a provided array of cookies.

Example:

```
// create dictionary with properties:
dim prop as new Dictionary
dim d as new date
d.Year = d.Year + 1

prop.Value(NSHTTPCookieMBS.NSHTTPCookieVersion)="0"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieName)="test"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieValue)="some value"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieOriginURL)="http://www.mbsplugins.de/"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieDomain)="www.mbsplugins.de"
prop.Value(NSHTTPCookieMBS.NSHTTPCookiePath)="/"
prop.Value(NSHTTPCookieMBS.NSHTTPCookieExpires)=d

// create cookie from properties
dim cookie as new NSHTTPCookieMBS(prop)
dim cookies(-1) as NSHTTPCookieMBS
cookies.Append cookie

// get request headers
dim dic as Dictionary = NSHTTPCookieMBS.requestHeaderFieldsWithCookies(cookies)

// and show them:
dim list(-1) as string

for each key as Variant in dic.keys
List.Append key.StringValue+": "+dic.Value(key).StringValue
next

MsgBox Join(list,EndOfLine)
```

Notes: cookies: The cookies from which the header fields are created.

Returns the dictionary of header fields created from the provided cookies. This dictionary can be used to add cookies to a request.

10.1.21 Properties**10.1.22 comment as string**

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's comment string.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display comment
MsgBox "Comment: "+Cookie.comment
```

Notes: The receiver's comment string or "" if the cookie has no comment. This string is suitable for presentation to the user, explaining the contents and purpose of this cookie.
(Read only property)

10.1.23 commentURL as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's comment URL.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display comment URL
MsgBox "commentURL: "+Cookie.commentURL
```

Notes: The receiver's comment URL or "" if the cookie has none. This value specifies a URL which is suitable for presentation to the user as a link for further information about this cookie.
(Read only property)

10.1.24 domain as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the domain of the receiver's cookie.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display domain:
MsgBox "Domain: "+Cookie.Domain
```

Notes: If the domain does not start with a dot, then the cookie is only sent to the exact host specified by the domain. If the domain does start with a dot, then the cookie is sent to other hosts in that domain as well, subject to certain restrictions. See RFC 2965 for more detail.
(Read only property)

10.1.25 expiresDate as date

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's expiration date.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display date:
MsgBox "Domain: "+Cookie.expiresDate.ShortTime+" "+Cookie.expiresDate.LongTime
```

Notes: The receiver's expiration date, or nil if there is no specific expiration date such as in the case of "session-only" cookies. The expiration date is the date when the cookie should be deleted.
(Read only property)

10.1.26 expiresDateTime as dateTime

Plugin Version: 20.5, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's expiration date.

Notes: (Read only property)

10.1.27 Handle as Integer

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal object reference.

Notes: (Read and Write property)

10.1.28 isHTTPOnly as boolean

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns whether the receiver should only be sent to HTTP servers per RFC 2965.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display isHTTPOnly value:
MsgBox "isHTTPOnly: "+str(cookie.isHTTPOnly)
```

Notes: Returns true if this cookie should only be sent via HTTP headers, false otherwise.

Cookies may be marked as HTTP only by a server (or by a javascript). Cookies marked as such must only be sent via HTTP Headers in HTTP requests for URL's that match both the path and domain of the respective cookies.

Important: Cookies specified as HTTP only should not be delivered to any javascript applications to prevent cross-site scripting vulnerabilities.

(Read only property)

10.1.29 isSecure as boolean

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns whether his cookie should only be sent over secure channels.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display isSecure value:
MsgBox "isSecure: "+str(Cookie.isSecure)
```

Notes: True if this cookie should only be sent over secure channels, otherwise false.
(Read only property)

10.1.30 isSessionOnly as boolean

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns whether the receiver should be discarded at the end of the session (regardless of expiration date).

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display isSessionOnly value:
MsgBox "isSessionOnly: "+str(Cookie.isSessionOnly)
```

Notes: True if the receiver should be discarded at the end of the session (regardless of expiration date), otherwise false.
(Read only property)

10.1.31 name as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's name.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display name:
MsgBox "name: "+Cookie.name
```

Notes: (Read only property)

10.1.32 path as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's path.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display path:
MsgBox "path: "+Cookie.path
```

Notes: The cookie will be sent with requests for this path in the cookie's domain, and all paths that have this prefix. A path of "/" means the cookie will be sent for all URLs in the domain.
(Read only property)

10.1.33 properties as dictionary

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's cookie properties.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display properties

dim dic as Dictionary = cookie.properties
dim list(-1) as string

for each key as Variant in dic.keys
List.Append key.StringValue+": "+dic.Value(key).StringValue
next

MsgBox Join(list,EndOfLine)
```

Notes: Returns dictionary representation of the receiver's cookie properties.

This dictionary can be used with Constructor or cookieWithProperties to create an equivalent NSHTTPCookie object.

(Read only property)

10.1.34 value as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's value.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display value
MsgBox "Value: "+Cookie.value
```

Notes: (Read only property)

10.1.35 version as Integer

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the receiver's version.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// pick first one
dim cookie as NSHTTPCookieMBS = cookies(0)

// display version value:
MsgBox "Version: "+str(Cookie.Version)
```

Notes: Returns the receiver's version. Version 0 maps to "old-style" Netscape cookies. Version 1 maps to RFC 2965 cookies.

(Read only property)

10.2 class NSHTTPCookieStorageMBS

10.2.1 class NSHTTPCookieStorageMBS

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: NSHTTPCookieStorage implements a singleton object (shared instance) that manages the shared cookie storage.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// we collect values
dim list(-1) as string
for each cookie as NSHTTPCookieMBS in cookies
List.Append cookie.name+": "+cookie.value
next

// and display them
MsgBox join(list, EndOfLine)
```

Notes: These cookies are shared among all applications and are kept in sync cross-process.

Note: Changes made to the cookie accept policy affect all currently running applications using the cookie storage.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [Cookies in HTMLViewer](#)
- [MBS Plugins 11.1 Release notes](#)
- [MBS REALbasic Plugins, version 11.1pr3](#)

10.2.2 Methods

10.2.3 Constructor

Plugin Version: 12.3, Platform: macOS, Targets: Desktop, Console & Web.

Function: The private constructor.

10.2.4 cookies as NSHTTPCookieMBS()

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the cookie storage's cookies.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookies

// we collect values
dim list(-1) as string
for each cookie as NSHTTPCookieMBS in cookies
List.Append cookie.name+": "+cookie.value
next

// and display them
MsgBox join(list, EndOfLine)
```

Notes: Returns an array containing all of the cookie storage's cookies.

10.2.5 cookiesForURL(URL as string) as NSHTTPCookieMBS()

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns all the cookie storage's cookies that are sent to a specified URL.

Example:

```
// query list of all cookies in shared storage
dim cookies(-1) as NSHTTPCookieMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookiesForURL("http://www.apple.com/")

// we collect values for apple.com:
dim list(-1) as string
for each cookie as NSHTTPCookieMBS in cookies
List.Append cookie.name+": "+cookie.value
next

// and display them
MsgBox join(list, EndOfLine)
```

Notes: An application can use NSHTTPCookie method requestHeaderFieldsWithCookies to turn this array into a set of header fields to add to an NSMutableURLRequest object.

10.2.6 cookiesToArray(cookies() as NSHTTPCookieMBS) as Integer

Plugin Version: 16.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Puts the cookie storage's cookies in the given array.

Example:

```
// get storage
dim s as NSHTTPCookieStorageMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage

// predim an array with a lot of space
dim cookies(5000) as NSHTTPCookieMBS

// now ask plugin to put cookies inside
dim c as Integer = s.cookiesToArray(cookies)

// show count
MsgBox str(c)+" cookies"

// pick first and show name
dim cookie as NSHTTPCookieMBS = cookies(0)
MsgBox cookie.name
```

Notes: Returns total number of cookies.

This is for Xojo 2007 where the cookies function doesn't work.

For Xojo and Xojo you can use cookies function.

10.2.7 deleteCookie(cookie as NSHTTPCookieMBS)

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Deletes the specified cookie from the cookie storage.

10.2.8 NSHTTPCookieManagerAcceptPolicyChangedNotification as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the notification names you can register with the NSNotificationObserverMBS class.

Notes: This notification is posted when the acceptance policy of the NSHTTPCookieStorage instance has changed.

In Mac OS X, cookies are shared among applications, meaning this notification can be sent in response to another application's actions. Cookies are not shared among applications in iOS.

The notification object is the NSHTTPCookieStorage instance. This notification does not contain a userInfo dictionary.

10.2.9 NSHTTPCookieManagerCookiesChangedNotification as string

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the notification names you can register with the NSNotificationObserverMBS class.

Notes: This notification is posted when the cookies stored in the NSHTTPCookieStorage instance have changed.

In Mac OS X, cookies are shared among applications, meaning this notification can be sent in response to another application's actions. Cookies are not shared among applications in iOS.

The notification object is the NSHTTPCookieStorage instance. This notification does not contain a userInfo dictionary.

10.2.10 removeCookiesSinceDate(d as date)

Plugin Version: 17.4, Platform: macOS, Targets: Desktop, Console & Web.

Function: Delete all cookies from the cookie storage since the provided date.

Notes: Available on macOS 10.10 or newer.

See also:

- 10.2.11 removeCookiesSinceDate(d as dateTime)

771

10.2.11 removeCookiesSinceDate(d as dateTime)

Plugin Version: 20.5, Platform: macOS, Targets: Desktop, Console & Web.

Function: Delete all cookies from the cookie storage since the provided date.

Notes: Available on macOS 10.10 or newer.

See also:

- 10.2.10 removeCookiesSinceDate(d as date)

771

10.2.12 setCookie(cookie as NSHTTPCookieMBS)

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Stores a specified cookie in the cookie storage if the cookie accept policy permits.

Notes: The cookie replaces an existing cookie with the same name, domain, and path, if one exists in

the cookie storage. This method accepts the cookie only if the receiver's cookie accept policy is `NSHTTPCookieAcceptPolicyAlways` or `NSHTTPCookieAcceptPolicyOnlyFromMainDocumentDomain`. The cookie is ignored if the receiver's cookie accept policy is `NSHTTPCookieAcceptPolicyNever`.

10.2.13 `setCookies(cookies() as NSHTTPCookieMBS, URL as string, mainDocumentURL as string)`

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Adds an array of cookies to the receiver if the receiver's cookie acceptance policy permits.

Notes: `cookies`: The cookies to add.

`URL`: The URL associated with the added cookies.

`mainDocumentURL`: The URL of the main HTML document for the top-level frame, if known. Can be `""`. This URL is used to determine if the cookie should be accepted if the cookie accept policy is `NSHTTPCookieAcceptPolicyOnlyFromMainDocumentDomain`.

The cookies will replace existing cookies with the same name, domain, and path, if one exists in the cookie storage. The cookie will be ignored if the receiver's cookie accept policy is `NSHTTPCookieAcceptPolicyNever`.

To store cookies from a set of response headers, an application can use `cookiesWithResponseHeaderFields` passing a header field dictionary and then use this method to store the resulting cookies in accordance with the receiver's cookie acceptance policy.

10.2.14 `sharedHTTPCookieStorage as NSHTTPCookieStorageMBS`

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the shared cookie storage instance.

Example:

```
dim s as NSHTTPCookieStorageMBS = NSHTTPCookieStorageMBS.sharedHTTPCookieStorage
dim cookies() as NSHTTPCookieMBS = s.cookies
MsgBox str(UBound(cookies)+1)+" cookies"
```

10.2.15 Properties

10.2.16 Handle as Integer

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal reference to the NSHTTPCookieStorage object.

Notes: (Read and Write property)

10.2.17 cookieAcceptPolicy as Integer

Plugin Version: 11.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The cookie storage's cookie accept policy.

Example:

```
Select case NSHTTPCookieStorageMBS.sharedHTTPCookieStorage.cookieAcceptPolicy
case NSHTTPCookieStorageMBS.NSHTTPCookieAcceptPolicyOnlyFromMainDocumentDomain
MsgBox "Cookies: only from main document domain"
case NSHTTPCookieStorageMBS.NSHTTPCookieAcceptPolicyNever
MsgBox "Cookies: never"
case NSHTTPCookieStorageMBS.NSHTTPCookieAcceptPolicyAlways
MsgBox "Cookies: always"
else
MsgBox "Cookies: unknown setting"
end Select
```

Notes: The default cookie accept policy is NSHTTPCookieAcceptPolicyAlways.

Changing the cookie policy affects all currently running applications using the cookie storage.
(Read and Write computed property)

10.2.18 Constants

One of the cookie accept policy constants.

Constant	Value	Description
NSHTTPCookieAcceptPolicyAlways	0	Accept all cookies. This is the default cookie ac
NSHTTPCookieAcceptPolicyNever	1	Reject all cookies.
NSHTTPCookieAcceptPolicyOnlyFromMainDocumentDomain	2	Accept cookies only from the main document d

Chapter 11

Cocoa Toolbar

11.1 class CustomNSToolbarItemMBS

11.1.1 class CustomNSToolbarItemMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The class for a custom toolbar item.

Notes: Use this class if you need to fill events.

Subclass of the NSToolbarItemMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 13.1pr5](#)
- [MBS Real Studio Plugins, version 11.3pr14](#)
- [MBS Real Studio Plugins, version 11.3pr4](#)

11.1.2 Methods

11.1.3 Constructor(itemIdentifier as string)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The constructor.

Notes: itemIdentifier: The identifier for the receiver. itemIdentifier is never seen by users and should not be localized.

The identifier is used by the toolbar and its delegate to identify the kind of the toolbar item.

11.1.4 Destructor

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The destructor.

11.1.5 Events

11.1.6 Action

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: The action event called when user clicks toolbar item.

11.1.7 `allowsDuplicatesInToolbar` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the receiver can be represented in the toolbar at more than one position.

Notes: You use this event to override the default behavior in a subclass.

By default, if an item with the same identifier is already in the toolbar, dragging it in again will effectively move it to the new position.

If you leave this event empty (no code), you get the default behavior.

11.1.8 `validate` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Validates a toolbar item.

Notes: If this event is implemented and returns false, `NSToolbar` will disable `theItem`; returning true causes `theItem` to be enabled.

`NSToolbar` only calls this method for image items.
`validate` is called very frequently, so it must be efficient.

If you don't have code in this event, the plugin will simply return enabled value.
This methods may not be called if you set `Autovalidates` to true.

11.2 class CustomNSToolbarMBS

11.2.1 class CustomNSToolbarMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The toolbar class for the case you need to use the events.

Notes: Subclass of the NSToolbarMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 13.1pr13](#)
- [MBS Real Studio Plugins, version 13.1pr5](#)
- [MBS Real Studio Plugins, version 11.3pr4](#)

11.2.2 Methods

11.2.3 Constructor(Identifier as string)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

11.2.4 Destructor

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The destructor.

11.2.5 Events

11.2.6 allowsSizeMode(mode as Integer, SuperAllows as boolean) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Whether to allow the given size mode.

Example:

```
Function allowsSizeMode(mode as Integer, SuperAllows as boolean) As boolean
#if DebugBuild then
System.DebugLog CurrentMethodName+" "+str(SuperAllows)
#endif
```

```

if mode = NSToolbarMBS.NSToolbarSizeModeSmall then
Return false
else
Return SuperAllows
end if

```

End Function

Notes: If you have nothing in the event, the default behavior is to allow all. You can return false for a mode to disable it. e.g. NSToolbarSizeModeSmall SuperAllows parameter tells you if mode is enabled by default.

11.2.7 itemForItemIdentifier(identifier as string, willBeInsertedIntoToolbar as boolean) as NSToolbarItemMBS

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Sent to request a new toolbar item; returns a toolbar item of the identified kind for the specified toolbar.

Notes: itemIdentifier: The identifier for the requested item.

willBeInsertedIntoToolbar: True if the item will be immediately inserted into the toolbar. If flag is false the toolbar item is being requested for display in the toolbar customization sheet and should always be enabled or provide some other canonical representation. If you ignore this parameter the same toolbar item will be used in the toolbar and in the customization sheet.

Return Value

The toolbar item for the specified toolbar and identifier. Return nil to indicate that the identified kind of toolbar item is not supported. When an item is requested again, you may return the same NSToolbarItem object returned earlier or a different instance.

Implement this method to create new toolbar item instances. This method is called lazily on behalf of a toolbar instance, which must be the sole owner of the toolbar item. A toolbar may ask again for a kind of toolbar item already supplied to it, in which case this method may return the same toolbar item it returned before or a different one. If your delegate services multiple toolbars, each attached to a different window, it is best to return a different item for each toolbar—an NSToolbarItem object can only be in one toolbar at a time.

If the item is a custom view item, the NSView object must be fully formed when the item is returned. Do not assume that the returned item is going to be added as an active item in the toolbar, as it could be that it will be used only in the customization palette. (The customization palette makes a copy of the returned item.)

This event must be implemented if the associated toolbar is created programmatically.

11.2.8 toolbarAllowedItemIdentifiers as string()

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Sent to discover the allowed item identifiers for a toolbar.

Notes: An array of toolbar item identifiers for toolbar, specifying the contents and the order of the items in the configuration palette.

Every allowed item must be explicitly listed, even the standard ones. The identifiers returned should include all of those returned by toolbarDefaultItemIdentifiers:.

This event must be implemented if the associated toolbar is created programatically.

11.2.9 toolbarDefaultItemIdentifiers as string()

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Sent to discover the default item identifiers for a toolbar.

Notes: Return an array of toolbar item identifiers for toolbar, specifying the contents and the order of the items in the default toolbar configuration.

During initialization of toolbar, this method is called only if a toolbar configuration for the identifier of toolbar is not found in the user preferences. This method is called during initialization of the toolbar customization palette.

This event must be implemented if the associated toolbar is created programatically.

11.2.10 toolbarDidRemoveItem(item as NSToolbarItemMBS, notification as NSNotificationMBS)

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Sent just after an item has been removed from a toolbar.

Notes: This method allows you to remove information related to the item that may have been cached.

11.2.11 toolbarItemAction(item as NSToolbarItemMBS)

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: The action event called when user clicks toolbar item.

Notes: Only for instances of CustomNSToolbarItemMBS, the plugin can forward the allowsDuplicatesIn-

Toolbar event to the CustomNSToolbarMBS, so you can handle it in a central place for all toolbar items.

11.2.12 toolbarItemAllowsDuplicatesInToolbar(item as NSToolbarItemMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Returns a Boolean value that indicates whether the receiver can be represented in the toolbar at more than one position.

Notes: You use this event to override the default behavior in a subclass.

By default, if an item with the same identifier is already in the toolbar, dragging it in again will effectively move it to the new position.

If you leave this event empty (no code), you get the default behavior.

Only for instances of CustomNSToolbarItemMBS, the plugin can forward the allowsDuplicatesInToolbar event to the CustomNSToolbarMBS, so you can handle it in a central place for all toolbar items.

11.2.13 toolbarItemValidate(item as NSToolbarItemMBS) as boolean

Plugin Version: 13.1, Platform: macOS, Targets: .

Function: Validates a toolbar item.

Notes: If this event is implemented and returns false, NSToolbar will disable theItem; returning true causes theItem to be enabled.

NSToolbar only calls this method for image items.
validateis called very frequently, so it must be efficient.

If you don't have code in this event, the plugin will simply return enabled value.
This methods may not be called if you set Autovalidates to true.

Only for instances of CustomNSToolbarItemMBS, the plugin can forward the allowsDuplicatesInToolbar event to the CustomNSToolbarMBS, so you can handle it in a central place for all toolbar items.

11.2.14 toolbarSelectableItemIdentifiers as string()

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Sent to discover the selectable item identifiers for a toolbar.

Notes: Return an array of item identifiers that should indicate selection in the specified toolbar.

Toolbars that need to indicate item selection should return an array containing the identifiers of the selectable toolbar items.

If implemented, toolbar will display the currently selected item with a visual highlight. Clicking on an item whose identifier is selectable will automatically update the toolbar's selected item identifier, when possible. Clicking an item whose identifier is not selectable will not update the toolbar's selected item identifier.

11.2.15 toolbarWillAddItem(item as NSToolbarItemMBS, notification as NSNotificationMBS)

Plugin Version: 11.3, Platform: macOS, Targets: .

Function: Sent just before a new item is added to a toolbar.

Notes: If you need to cache a reference to a toolbar item or need to set up some initial state before a toolbar item is added, this is where to do it.

11.3 class NSToolbarItemGroupMBS

11.3.1 class NSToolbarItemGroupMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: NSToolbarItemGroup is a subclass of NSToolbarItem which contains subitems.

Notes: The views and labels of the subitems are used, but the parent's attributes take precedence. Subclass of the NSToolbarItemMBS class.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 15.0](#)
- [MBS Xojo / Real Studio Plugins, version 14.5pr3](#)

11.3.2 Methods

11.3.3 Constructor(itemIdentifier as string)

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The constructor.

Notes: itemIdentifier: The identifier for the receiver. itemIdentifier is never seen by users and should not be localized.

The identifier is used by the toolbar and its delegate to identify the kind of the toolbar item.

11.3.4 SetSubItems(items() as NSToolbarItemMBS)

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: Sets the subitems for the receiver.

Notes: subitems: An array of instances of NSToolbarItem objects that form the subitems for the receiver.

You should call this method to set the subitems before returning the item to the toolbar. NSToolbarItemGroup objects cannot contain other NSToolbarItemGroup objects as subitems.

11.3.5 subitems as NSToolbarItemMBS()

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: Returns the subitems for the receiver.

Notes: By default, an NSToolbarItemGroup instance has an empty array of subitems.

11.4 class NSToolbarItemMBS

11.4.1 class NSToolbarItemMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The class for Cocoa toolbar items.

Notes: Each item in an NSToolbar is an instance of NSToolbarItem.

Blog Entries

- [News from the MBS Xojo Plugins Version 21.2](#)
- [MBS Xojo Plugins, version 21.2pr1](#)
- [MBS Xojo Plugins, version 21.1pr4](#)
- [MBS Real Studio Plugins, version 11.3pr4](#)

11.4.2 Methods

11.4.3 Constructor(Handle as Integer)

Plugin Version: 21.1, Platform: macOS, Targets: Desktop only.

Function: Constructor for creating an instanced based on an existing handle.

Notes: Useful if you get a NSToolbarItem reference from a declare.

The object is retained.

See also:

- 11.4.4 Constructor(itemIdentifier as string)

784

11.4.4 Constructor(itemIdentifier as string)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The constructor.

Notes: itemIdentifier: The identifier for the receiver. itemIdentifier is never seen by users and should not be localized.

The identifier is used by the toolbar and its delegate to identify the kind of the toolbar item.

See also:

- 11.4.3 Constructor(Handle as Integer)

784

11.4.5 NSToolbarCloudSharingItemIdentifier as string

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: A standard toolbar item identifier for cloud sharing via NSSharingServiceNameCloudSharing.

Notes: It validates itself and modifies its appearance by using the NSCloudSharingValidation protocol. It sends performCloudSharing to the firstResponder.

Requires macOS 10.12 or newer.

11.4.6 NSToolbarCustomizeToolbarItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Customize item. Shows the customization palette.

Deprecated. In OS X v10.7 and later the customization icon has been removed from the toolbar and customization palettes. This constant is ignored.

11.4.7 NSToolbarFlexibleSpaceItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Flexible Space item.

11.4.8 NSToolbarPrintItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Print item. Sends printDocument to firstResponder, but you can change this in toolbarWillAddItem if you need to do so.

11.4.9 NSToolbarSeparatorItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Separator item.

11.4.10 NSToolbarShowColorsItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Colors item. Shows the color panel.

11.4.11 NSToolbarShowFontsItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Fonts item. Shows the font panel.

11.4.12 NSToolbarSidebarTrackingSeparatorItemIdentifier as string

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: One of the standard identifiers.

Notes: Requires macOS 11.0 or newer.

11.4.13 NSToolbarSpaceItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the standard toolbar item identifiers.

Notes: The Space item.

11.4.14 NSToolbarToggleSidebarItemIdentifier as string

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: A standard toolbar item identifier for sidebars.

Notes: It sends toggleSidebar to the firstResponder.

Requires macOS 10.11 or newer.

11.4.15 validate

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: This method is called by the receiver’s toolbar during validation.

Notes: You may invoke this method directly if you have disabled automatic validation for an item—typically you do this for performance reasons if your validation code is slow.

11.4.16 Properties

11.4.17 `allowsDuplicatesInToolbar` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether the receiver can be represented in the toolbar at more than one position.

Notes: Returns true to allow dragging the receiver into the toolbar at more than one position, otherwise false.

You use this method by overriding it in a subclass to always return true; typically, you wouldn’t call it. By default, if an item with the same identifier is already in the toolbar, dragging it in again will effectively move it to the new position.

(Read only property)

11.4.18 `autovalidates` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Whether this item auto validates.

Notes: By default NSToolbar automatically invokes the receiver’s validate method on a regular basis. If your validate method is time consuming, you can disable auto validation on a per toolbar item basis.

(Read and Write property)

11.4.19 `Bordered` as Boolean

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: When set on an item without a custom view, the button produced will have a bordered style.

Notes: Defaults to false.

Requires macOS 10.15 or newer.

(Read and Write property)

11.4.20 `ClassName` as String

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The name of this `NSWindow` class.

Notes: (Read only property)

11.4.21 `ClassPath` as String

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: The path of this `NSView` class.

Notes: Useful for debugging to know what super classes the window has.
(Read only property)

11.4.22 `Enabled` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the receiver is enabled.

Notes: For a view item, this method calls `isEnabled/setEnabled` on the view if it responds and returns the result.
(Read and Write property)

11.4.23 `Handle` as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The internal reference to the `NSToolBarItem` object.

Notes: (Read and Write property)

11.4.24 `image` as `NSImageMBS`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The image for this toolbar item.

Notes: For a custom view item (one whose view has already been set), this method calls `setImage` on the view if it responds. If image contains multiple representations, `NSToolBarItem` chooses the most appropriately sized representation when displaying.
(Read and Write property)

11.4.25 itemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The receiver's identifier.

Notes: Returns the receiver's identifier, which was provided in the initializer.
(Read only property)

11.4.26 label as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The label, which normally appears in the toolbar and in the overflow menu.

Notes: The length of the label should be appropriate and not too long. The label may be empty.
(Read and Write property)

11.4.27 MaxSize as NSSizeMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The maximum size.

Notes: (Read and Write property)

11.4.28 menuFormRepresentation as NSMenuItemMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The menu form representation.

Notes: By default, this method returns nil, even though there is a default menu form representation.
(Read and Write property)

11.4.29 MinSize as NSSizeMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The minimum size.

Notes: (Read and Write property)

11.4.30 Navigational as Boolean

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: Whether or not the item behaves as a navigation item (i.e. back/forward) in the toolbar.

Notes: Navigation items may be specially positioned by the system outside the normal list of items of the toolbar in the order specified by `toolbarDefaultItemIdentifiers`.

Requires macOS 11.0 or newer.

(Read and Write property)

11.4.31 paletteLabel as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The label that appears when the receiver is in the customization palette.

Notes: An item must have a palette label if the customization palette is to be used, and for most items it is reasonable to set `paletteLabel` to be the same value as `label`. One reason for `paletteLabel` to be different from `label` would be if it's more descriptive; another might be if there is no label.

(Read and Write property)

11.4.32 tag as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The tag value.

Notes: You can use the tag for your own custom purpose.

(Read and Write property)

11.4.33 Title as String

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: Set and get the title of an item.

Notes: For custom views, this method will call `setTitle:/title` on the view if it responds.

(forwards to `-view` if it responds)

Requires macOS 10.15 or newer.

(Read and Write property)

11.4.34 toolbar as NSToolbarMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the toolbar that is using the receiver.

Notes: (Read only property)

11.4.35 toolTip as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The tooltip to be used when the receiver is displayed in the toolbar.

Notes: (Read and Write property)

11.4.36 view as NSViewMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The custom view for this item.

Notes: view: The view for the receiver. The view and all of its contents must conform to the NSCodering protocol if the toolbar supports customization.

Note that many of the set/get methods are implemented by calls forwarded to view, if it responds to it.
(Read and Write property)

11.4.37 visibilityPriority as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The visibility priority.

Notes: (Read and Write property)

11.4.38 Constants

Constants

Constant	Value	Description
<code>NSToolbarItemVisibilityPriorityHigh</code>	1000	<p>One of the visibility priority constants.</p> <p>When a toolbar does not have enough space to fit all its items, it moves some items into the overflow menu. These values allow you to suggest a priority for a toolbar item.</p> <p>Items with this priority are less inclined to be pushed to the overflow menu.</p>
<code>NSToolbarItemVisibilityPriorityLow</code>	-1000	<p>One of the visibility priority constants.</p> <p>When a toolbar does not have enough space to fit all its items, it moves some items into the overflow menu. These values allow you to suggest a priority for a toolbar item.</p> <p>Items with this priority will be the first items to be pushed to the overflow menu.</p>
<code>NSToolbarItemVisibilityPriorityStandard</code>	0	<p>One of the visibility priority constants.</p> <p>When a toolbar does not have enough space to fit all its items, it moves some items into the overflow menu. These values allow you to suggest a priority for a toolbar item.</p> <p>The default visibility priority.</p>
<code>NSToolbarItemVisibilityPriorityUser</code>	2000	<p>One of the visibility priority constants.</p> <p>When a toolbar does not have enough space to fit all its items, it moves some items into the overflow menu. These values allow you to suggest a priority for a toolbar item.</p> <p>Items with this priority are the last to be pushed to the overflow menu. The user should set items to this priority.</p>

11.5 class NSToolbarMBS

11.5.1 class NSToolbarMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: NSToolbar and NSToolbarItem provide the mechanism for a titled window to display a toolbar just below its title bar.

Blog Entries

- [MBS Xojo Plugins, version 21.2pr1](#)
- [MBS Xojo Plugins, version 21.1pr4](#)
- [MBS Xojo Plugins, version 17.5pr9](#)
- [MBS Xojo / Real Studio Plugins, version 14.3pr10](#)
- [Please don't call constructors outside new or constructor](#)
- [MBS Real Studio Plugins, version 11.3pr4](#)

11.5.2 Methods

11.5.3 Constructor(Handle as Integer)

Plugin Version: 21.1, Platform: macOS, Targets: Desktop only.

Function: Constructor for creating an instanced based on an existing handle.

Notes: Useful if you get a NSToolbar reference from a declare.

The object is retained.

See also:

- 11.5.4 Constructor(Identifier as string)

793

11.5.4 Constructor(Identifier as string)

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The constructor.

Notes: Initializes a newly allocated toolbar with the specified identifier.

identifier is never seen by users and should not be localized.

See also:

- 11.5.3 Constructor(Handle as Integer)

793

11.5.5 `insertItemWithIdentifier(identifier as string, atIndex as Integer)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Inserts the specified item at the specified index.

Notes: Identifier: The identifier of the item to insert.

index: The index at which to insert the item.

If the toolbar needs a new instance, it will get it from `itemForItemIdentifier`. Typically, you should not call this method; you should let the user reconfigure the toolbar.

11.5.6 `items as NSToolbarItemMBS()`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the receiver's current items, in order.

11.5.7 `NSToolbarDidRemoveItemNotification as string`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the notification identifiers.

Notes: Posted after an item is removed from a toolbar. The notification item is the `NSToolbar` object that had an item removed from it. The `userInfo` dictionary contains the following information: `item`: The `NSToolbarItem` object that was removed.

11.5.8 `NSToolbarWillAddItemNotification as string`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: One of the notification identifiers.

Notes: Posted before a new item is added to the toolbar. The notification item is the `NSToolbar` object having an item added to it. The `userInfo` dictionary contains the following information: `item`: The `NSToolbarItem` object being added.

11.5.9 `removeItemAtIndex(index as Integer)`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Removes the specified item.

Notes: Typically, you should not call this method; you should let the user reconfigure the toolbar.

Index is zero based.

11.5.10 runCustomizationPalette

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Runs the receiver's customization palette.

11.5.11 validateVisibleItems

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Called on window updates to validate the visible items.

Notes: You typically use this method by overriding it in a subclass. The default implementation of this method iterates through the list of visible items, sending each a validate message. Override it and call super if you want to know when this method is called.

In Mac OS X v 10.6 and later toolbars no longer automatically validate for some events, including: NSLeftMouseDown, NSRightMouseDown, NSOtherMouseDown, NSMouseEntered, NSMouseExited, NSScrollWheel, NSCursorUpdate, NSKeyDown. In addition, validation for NSKeyUp and NSFlagsChanged events is deferred with the timer restarting for every new deferrable event. So a sequence of key events will not trigger any validation at all, until either a pause of .85 seconds, or an event other than NSKeyUp or NSFlagsChanged is processed. This change was made as an optimization.

To trigger validation for a single toolbar manually, send the toolbar a validateVisibleItems message. To trigger validation for all toolbars, invoke NSApplication's setWindowsNeedUpdate passing true.

11.5.12 visibleItems as NSToolBarItemMBS()

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the receiver's currently visible items.

Notes: Items in the overflow menu are not considered visible.

11.5.13 Properties

11.5.14 `allowsExtensionItems` as Boolean

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the toolbar can add items for Action extensions.

Notes: When true, the toolbar can dynamically create toolbar items for Action extensions in the toolbar configuration panel. To be included, an extension needs to set the `NSExtensionServiceAllowsToolbarItem` key to true in its `Info.plist`.

The default value is false.

(Read and Write property)

11.5.15 `allowsUserCustomization` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether users are allowed to modify the toolbar.

Notes: True if users are allowed to modify the toolbar, false otherwise. The default is false.

If the value is false, then the Customize Toolbar,  menu item is disabled and other modification is disabled. This attribute does not affect the user's ability to show or hide the toolbar.

(Read and Write property)

11.5.16 `autosavesConfiguration` as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the receiver autosaves its configuration.

Notes: True if the receiver autosaves its configuration, otherwise false. The default is false.

When autosaving is enabled, the receiver will automatically write the toolbar settings to user defaults if the toolbar configuration changes. The toolbar's configuration is identified in user defaults by the toolbar identifier. If there are multiple toolbars active with the same identifier, they all share the same configuration.

(Read and Write property)

11.5.17 `centeredItemIdentifier` as String

Plugin Version: 21.2, Platform: macOS, Targets: Desktop only.

Function: The center item identifier.

Notes: The item with the specified identifier will be positioned in the absolute center of the Toolbar relative to the window assuming space allows. When the window shrinks, the highest priority is to have the most items visible. Thus, centering is broken first (it'll be pushed off to the left/right as necessary). Next, items will be shrunk down a little at a time towards their min size, at the same rate. Finally, items will be removed based on their visibility priority.

This property is archived.
(Read and Write property)

11.5.18 configurationDictionary as dictionary

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The receiver's configuration as a dictionary.

Notes: Set/get a dictionary containing configuration information for the toolbar.

Contains displayMode, isVisible, and a list of the item identifiers currently in the toolbar.
Do not depend on any details of the normal contents of a configuration dictionary.
(Read and Write property)

11.5.19 configurationDictionaryData as Memoryblock

Plugin Version: 14.3, Platform: macOS, Targets: Desktop only.

Function: The receiver's configuration as a memoryblock.

Notes: Set/get a memoryblock containing configuration information for the toolbar.

Contains displayMode, isVisible, and a list of the item identifiers currently in the toolbar.
Do not depend on any details of the normal contents of a configuration dictionary.

you can read/write the data to preferences.
(Read and Write property)

11.5.20 customizationPaletteIsRunning as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns a Boolean value that indicates whether the receiver's customization palette is running (in use).

Notes: True if the receiver's customization palette is running, otherwise false.

(Read only property)

11.5.21 `displayMode` as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The receiver's display mode.

Notes: (Read and Write property)

11.5.22 `fullScreenAccessoryView` as `NSViewMBS`

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The fullscreen mode accessory view.

Notes: Requires Mac OS X 10.7.

(Read and Write property)

11.5.23 `fullScreenAccessoryViewMaxHeight` as Double

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The fullscreen mode accessory view's maximum height.

Notes: Requires Mac OS X 10.7.

(Read and Write property)

11.5.24 `fullScreenAccessoryViewMinHeight` as Double

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The fullscreen mode accessory view's minimum height.

Notes: Requires Mac OS X 10.7.

(Read and Write property)

11.5.25 `Handle` as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The internal reference for this object.

Notes: (Read and Write property)

11.5.26 identifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Returns the receiver's identifier.

Notes: Returns the receiver's identifier, a string used by the class to identify the kind of toolbar.

Within the application all toolbars with the same identifier are synchronized to maintain the same state, including for example, the display mode and item order. The identifier is used as the autosave name for toolbars that save their configuration.

(Read only property)

11.5.27 selectedItemIdentifier as string

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The selected item to the specified toolbar item.

Notes: Typically, a toolbar will manage the selection of items automatically. This method can be used to select identifiers of custom view items, or to force a selection change. See `toolbarSelectableItemIdentifiers` for more details. If `itemIdentifier` is not recognized by the receiver, the current selected item identifier does not change.

(Read and Write property)

11.5.28 showsBaselineSeparator as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: A Boolean value that indicates whether the toolbar shows the separator between the toolbar and the main window contents.

Notes: True if the toolbar shows the separator between the toolbar and the main window contents, otherwise false. The default is true.

(Read and Write property)

11.5.29 sizeMode as Integer

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The size mode.

Notes: If there is no icon of the given size for a toolbar item, the toolbar item creates one by scaling an icon of another size.

(Read and Write property)

11.5.30 toolbarView as NSViewMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The internal NSView used for displaying the toolbar.

Notes: Useful to query values like the bounds of the toolbar.

Works for 64-bit with 17.5 version, but returns nil in older versions.
(Read only property)

11.5.31 visible as boolean

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Whether the receiver is visible or hidden.

Notes: True to indicate the receiver should be made visible, false to indicate it should be hidden.
(Read and Write property)

11.5.32 Constants

Display Mode Constants

Constant	Value	Description
NSToolbarDisplayModeDefault	0	The default display mode.
NSToolbarDisplayModeIconAndLabel	1	The toolbar will display icons and labels.
NSToolbarDisplayModeIconOnly	2	The toolbar will display only icons.
NSToolbarDisplayModeLabelOnly	3	The toolbar will display only labels.

SizeMode Constants

Constant	Value	Description
NSToolbarSizeModeDefault	0	The toolbar uses the system-defined default size, which is NSToolbarSizeModeRegular.
NSToolbarSizeModeRegular	1	The toolbar uses regular-sized controls and 32 by 32 pixel icons.
NSToolbarSizeModeSmall	2	The toolbar uses small-sized controls and 24 by 24 pixel icons.

Chapter 12

Controls

12.1 class DesktopListbox

12.1.1 class DesktopListbox

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in listbox class in Xojo.

12.1.2 Methods

12.1.3 HorizontalNSScrollerMBS as NSScrollerMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns horizontal scrollbar for a listbox.

Notes: Only for Cocoa target.

12.1.4 VerticalNSScrollerMBS as NSScrollerMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Returns vertical scrollbar for a listbox.

Notes: Only for Cocoa target.

12.2 class DesktopProgressbar

12.2.1 class DesktopProgressbar

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in progressbar class in Xojo.

12.2.2 Methods

12.2.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: Creates a NSProgressIndicatorMBS object for the given control.

Example:

```
MsgBox ProgressBar1.NSProgressIndicatorMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

12.3 class Listbox

12.3.1 class Listbox

Plugin Version: 13.0, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in listbox class in Xojo.

12.3.2 Methods

12.3.3 HorizontalNSScrollerMBS as NSScrollerMBS

Plugin Version: 13.0, Platform: macOS, Targets: Desktop only.

Function: Returns horizontal scrollbar for a listbox.

Notes: Only for Cocoa target.

Blog Entries

- [MBS Real Studio Plugins, version 13.0fc1](#)

12.3.4 VerticalNSScrollerMBS as NSScrollerMBS

Plugin Version: 13.0, Platform: macOS, Targets: Desktop only.

Function: Returns vertical scrollbar for a listbox.

Notes: Only for Cocoa target.

Blog Entries

- [MBS Real Studio Plugins, version 13.0fc1](#)

12.4 class ProgressBar

12.4.1 class ProgressBar

Plugin Version: 9.7, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: The built in progressbar class in Xojo.

12.4.2 Methods

12.4.3 NSProgressIndicatorMBS as NSProgressIndicatorMBS

Plugin Version: 9.7, Platform: macOS, Targets: Desktop only.

Function: Creates a NSProgressIndicatorMBS object for the given control.

Example:

```
MsgBox ProgressBar1.NSProgressIndicatorMBS.className
```

Notes: This way you can manipulate Cocoa controls directly.

Chapter 13

CoreGraphics

13.1 class QuartzFilterManagerMBS

13.1.1 class QuartzFilterManagerMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The class to manage the quartz filters on the system.

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr9](#)

13.1.2 Methods

13.1.3 filterPanel as NSPanelMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop only.

Function: Returns the panel where users can select a filter.

Example:

```
dim m as QuartzFiltermanagerMBS
```

```
dim p as NSPanelMBS
```

```
m=new QuartzFiltermanagerMBS
```

```
p=m.filterPanel
```

```
p.Show
```

Notes: Returns nil on any error.

13.1.4 filters as QuartzFilterMBS()

Plugin Version: 11.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array with all filters.

Example:

```
dim q as new QuartzFilterManagerMBS
dim a() as QuartzFilterMBS
```

```
a=q.filters
```

```
MsgBox str(ubound(a)+1)+" filters found."
```

13.1.5 filtersInDomains(domains() as string) as QuartzFilterMBS()

Plugin Version: 11.2, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array with all filters in the given domains.

Example:

```
dim domains() as string = array(QuartzFilterManagerMBS.kQuartzFilterPrintingDomain)
dim a() as QuartzFilterMBS = QuartzFilterManagerMBS.filtersInDomains(domains)
```

```
MsgBox str(ubound(a)+1)+" filters found."
```

13.1.6 filterView as QuartzFilterViewMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop only.

Function: The view which you can use to select views.

Notes: You can use the panel as an extra window or add this view to one of your settings windows.

13.1.7 selectedFilter as QuartzFilterMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The currently selected filter.

13.1.8 selectFilter(filter as QuartzFilterMBS) as boolean

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: Selects a filter.

Notes: Returns true on success.

13.1.9 Properties

13.1.10 Handle as Integer

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal used handle to the filter manager.

Notes: (Read and Write property)

13.1.11 Events

13.1.12 didAddFilter(filter as QuartzFilterMBS)

Plugin Version: 8.6, Platform: macOS, Targets: .

Function: The given filter was added to the filter list.

13.1.13 didModifyFilter(filter as QuartzFilterMBS)

Plugin Version: 8.6, Platform: macOS, Targets: .

Function: The given filter was modified.

13.1.14 didRemoveFilter(filter as QuartzFilterMBS)

Plugin Version: 8.6, Platform: macOS, Targets: .

Function: The given filter was removed.

13.1.15 didSelectFilter(filter as QuartzFilterMBS)

Plugin Version: 8.6, Platform: macOS, Targets: .

Function: The selection changed to the new filter.

13.1.16 Constants

Filter Domain Constants

Constant	Value	Description
kQuartzFilterApplicationDomain	"kQuartzFilterApplicationDomain"	
kQuartzFilterPDFWorkflowDomain	"kQuartzFilterPDFWorkflowDomain"	
kQuartzFilterPrintingDomain	"kQuartzFilterPrintingDomain"	

13.2 class QuartzFilterMBS

13.2.1 class QuartzFilterMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The Quartz filter class.

Example:

```
// Create a couple of PDF files to test different Quartz filters

// get some picture
dim pic as Picture = LogoMBS(500)
dim image as new NSImageMBS(pic)

// create page with picture
dim page as new PDFPageMBS(image)

// create new document
dim doc as new PDFDocumentMBS

// add page
doc.insertPage(page,0)

// get filters
dim manager as new QuartzFilterManagerMBS
dim filters() as QuartzFilterMBS = Manager.filters

for each filter as QuartzFilterMBS in filters

// save PDF with this filter
dim file as FolderItem = SpecialFolder.Desktop.Child(filter.localizedName+".pdf")

call doc.write(file, filter)

next
```

Notes: This filters can be used to change PDFs on writing like to reduce the file size.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Real Studio Plugins, version 11.2pr7](#)

13.2.2 Methods

13.2.3 applyToContext(CGContextHandle as Integer) as boolean

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: Applies a filter to a given context.

Example:

```
dim c as CGContextMBS // your context
dim fi as QuartzFilterMBS

fi=QuartzFilterMBS.quartzFilterWithFile(SpecialFolder.Desktop.Child("Reduce File Size.qfilter"))

call fi.applyToContext(c.Handle)
```

Notes: Returns true on success.

13.2.4 Constructor

Plugin Version: 13.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The private constructor.

13.2.5 localizedName as string

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The localized name of the filter.

Example:

```
dim q as new QuartzFilterManagerMBS
dim s(-1) as string

dim a() as QuartzFilterMBS = q.filters

for each f as QuartzFilterMBS in a
s.append f.localizedName
next

MsgBox Join(s,EndOfLine)
```

13.2.6 quartzFilterWithFile(file as folderitem) as QuartzFilterMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: Loads a quartz filter from the given folderitem.

Example:

```
dim c as CGContextMBS // your context
dim fi as QuartzFilterMBS
dim file as folderitem

file=SpecialFolder.Desktop.Child("Reduce File Size.qfilter")
fi=QuartzFilterMBS.quartzFilterWithFile(file)

call fi.applyToContext(c.Handle)
```

Notes: Returns nil on any error.

Requires Mac OS X 10.5.

13.2.7 quartzFilterWithURL(url as string) as QuartzFilterMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: Loads the quartz filter from the given URL.

Notes: Returns nil on any error.

Requires Mac OS X 10.5.

13.2.8 removeFromContext(CGContextHandle as Integer)

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: Removes a filter from a CGContext.

Notes: Pass CGContextMBS.handle and make sure it is not 0.

13.2.9 url as string

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The URL where this filter is located.

Example:

```
dim q as new QuartzFilterManagerMBS
```

```
dim s(-1) as string
```

```
dim a() as QuartzFilterMBS = q.filters
```

```
for each f as QuartzFilterMBS in a  
s.append f.url  
next
```

```
MsgBox Join(s,EndOfLine)
```

13.2.10 Properties

13.2.11 Handle as Integer

Plugin Version: 8.6, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal handle to the filter object.

Notes: (Read and Write property)

13.3 class QuartzFilterViewMBS

13.3.1 class QuartzFilterViewMBS

Plugin Version: 8.6, Platform: macOS, Targets: Desktop only.

Function: The view to select quartz filters.

Notes: You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard. Subclass of the NSViewMBS class.

Blog Entries

- [MBS Real Studio Plugins, version 12.1pr5](#)

13.3.2 Methods

13.3.3 Constructor

Plugin Version: 12.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new Quartz Filter View with size 100/100 and position 0/0

Example:

```
dim t as new QuartzFilterViewMBS
```

Notes: On success the handle property is not zero.

See also:

- [13.3.4 Constructor\(Handle as Integer\)](#) 813
- [13.3.5 Constructor\(left as Double, top as Double, width as Double, height as Double\)](#) 814

13.3.4 Constructor(Handle as Integer)

Plugin Version: 12.1, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given QuartzFilterViewMBS handle.

Example:

```
dim t as new QuartzFilterViewMBS(0, 0, 100, 100)
```

```
dim v as new QuartzFilterViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a QuartzFilterView and the plugin retains this handle.

See also:

- 13.3.3 Constructor 813
- 13.3.5 Constructor(left as Double, top as Double, width as Double, height as Double) 814

13.3.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 12.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new Quartz Filter View with the given size and position.

Example:

```
dim x as new QuartzFilterViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

The new movie view object must be inserted into the view hierarchy of an NSWindow before it can be used.

This method is the designated initializer for the QuartzFilterView class.

See also:

- 13.3.3 Constructor 813
- 13.3.4 Constructor(Handle as Integer) 813

13.3.6 sizeToFit

Plugin Version: 8.6, Platform: macOS, Targets: Desktop only.

Function: Resizes the view to the best size.

Chapter 14

DiscRecording

14.1 class DRBurnMBS

14.1.1 class DRBurnMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: A class to perform and monitor the burning of a CD or DVD disc.

Notes: Each time you want to burn to a disc, an instance of DRBurnMBS needs to be created.

When an instance is created, you pass in an instance of DRDevice to let the DRBurn object know what device to use. This object is retained for the life of the DRBurn instance. Before burning, you can set several options that control the behavior of the burn and the handling of the disc once the burn completes.

A DRBurn object will send out notifications through the DRNotificationCenter mechanism to broadcast the burn state to any interested observers. However, if for some reason you don't want to use notifications, you can poll the burn object at any time for the current status using the status properties. This is not recommended in any application using a run loop, because it involves polling.

All methods in this class will catch exceptions from Cocoa and raise a NSErrorMBS instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr1](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)
- [MonkeyBread Software Releases the MBS Plugins 8.2](#)

14.1.2 Methods

14.1.3 abort

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Stops the burn.

Notes: When this method returns the burn might not actually be fully stopped but it has been cancelled and only cleanup is going on. If a burn has not completed writing data to disc, you just made a coaster.

Typically this method is only used as a result of the user hitting a cancel/stop button somewhere in the user interface.

14.1.4 burnForDevice(device as DRDeviceMBS) as DRBurnMBS

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Creates an autoreleased burn object.

Notes: Once a burn is created with this method, the object is ready to write data to the disc.

14.1.5 Constructor(device as DRDeviceMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a burn object.

Notes: Once a burn is created with this method, the object is ready to write data to the disc.

14.1.6 device as DRDeviceMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the device being used for the burn.

Notes: Returns nil on any error.

14.1.7 DRBurnAppendableKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a boolean indicating if the disc will still be appendable after the burn. If this key is not present, the burn will default to a value of false and the disc will be marked as not

appendable.

14.1.8 DRBurnCompletionActionEject as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: An string value for the CompletionAction indicating that the burn object should eject the disc from the drive when the burn completes.

14.1.9 DRBurnCompletionActionKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a String containing one of the completion actions possible for the disc handling. If this key is not present, the burn will default to a value of DRBurnCompletionActionEject and the disc will be ejected.

14.1.10 DRBurnCompletionActionMount as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: An string value for the CompletionAction property indicating that the burn object should mount the disc on the desktop when the burn completes.

14.1.11 DRBurnDoubleLayerL0DataZoneBlocksKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property key whose value is an double containing the number of blocks desired for the layer 0 data zone on a double layer writable disc.

The size of the layer 0 data zone dictates where the transition point is from layer 0 to layer 1. If this key is present, the data zone size will be set prior to the start of the burn using the value for this key. If it is not present, the default layer 0 data zone will be used (half the available blocks on an empty disc).

The transition point can be specified two ways. If the value specified in this key is greater than 1.0, then it will designate an absolute block number for the transition point. In this case, the block number should be a multiple of 16 and at least 40000h per specification. If the value is less than 1.0, it will specify the

percentage of the burn that should reside on layer 0. A typical value is 0.5, designating half the burn for each layer. A value of 0.0 or 1.0 will not change the layer 0 transition point.

14.1.12 `DRBurnFailureActionEject` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: An string constant for FailureAction indicating that the burn object should eject the disc from the drive if the burn fails.

14.1.13 `DRBurnFailureActionKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a string containing a one of the failure actions possible for the disc handling.

If this key is not present, the burn will default to a value of `DRBurnFailureActionEject` and the disc will be ejected.

14.1.14 `DRBurnFailureActionNone` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: An string constant for FailureAction indicating that the burn object should do nothing with the disc if the burn fails.

14.1.15 `DRBurnOverwriteDiscKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a boolean indicating if the disc will be overwritten from block zero for the burn. If this key is not present, the burn will default to a value of false and the disc will be appended.

14.1.16 DRBurnRequestedSpeedKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a number containing the speed at which the burn should run, expressed as a float value of kilobytes per second. If this key is not present, the speed will default to DRDeviceBurnSpeedMax.

14.1.17 DRBurnStatusChangedNotification as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: The notification name for a status update on a burn operation.

Notes: See the "DataBurn with Events and Notification" example project.

14.1.18 DRBurnStrategyBDDAO as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the string constants for the burn strategy.

Notes: An string value for DRBurnStrategyKey representing the DAO (disc-at-once) burn strategy for BD (Blu-ray). This strategy applies only to BDs.

14.1.19 DRBurnStrategyCDSA0 as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the string constants for the burn strategy.

Notes: An String value for DRBurnStrategyKey representing the SAO (session-at-once) burn strategy for CD.

14.1.20 DRBurnStrategyCDTAO as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the string constants for the burn strategy.

Notes: An String value for DRBurnStrategyKey representing the TAO (track-at-once) burn strategy for CD.

14.1.21 `DRBurnStrategyDVDDAO` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the string constants for the burn strategy.

Notes: An String value for `DRBurnStrategyKey` representing the DAO (disc-at-once) burn strategy for DVD. This strategy applies only to DVDs; it is invalid when burning to CD media.

14.1.22 `DRBurnStrategyIsRequiredKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a `BOOL` indicating whether the burn strategy/strategies listed for `DRBurnStrategyKey` are the only strategies allowed. If this key is not present, the burn will default to a value of false.

If this value is set to true, and the device does not support the type(s) of burn requested, the burn will fail with `kDRDeviceBurnStrategyNotAvailableErr` .

If this value is set to false, and the device does not support the type(s) of burn requested, the engine will choose an alternate burn strategy automatically - one that will provide an equivalent disc.

14.1.23 `DRBurnStrategyKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a string, or array of strings, indicating the burn strategy or strategies that are suggested. If this key is not present, the burn engine picks an appropriate burn strategy automatically. Most clients will not need to specify a specific burn strategy.

When more than one strategy is suggested, the burn engine will attempt to use the first strategy in the list which is available. A burn strategy will never be used if it cannot write the required data: for example, TAO cannot be used to write CD-Text.

The presence of this key by itself is just a suggestion, and if the burn engine cannot fulfill the request it will burn using whatever strategy is available. To make the suggestion into a requirement, add `DRBurnStrategyIsRequiredKey` with a value of true.

14.1.24 DRBurnTestingKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a boolean indicating if the burn will run as a test burn.

When this is set and the burn object is sent writeLayout, the entire burn process proceeds as if data would be written to the disc, but the laser is not turned on to full power, so the physical disc is not modified.

If this key is not present or the selected burning device does not support test burning, the burn will default to a value of false and a normal burn will occur.

14.1.25 DRBurnUnderrunProtectionKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a boolean indicating if burn underrun protection will be on or off for devices which support it.

For those devices which support it, burn underrun protection is enabled by default.

If the device supports burn underrun protection and this key is not present, the burn will default to a value of true and burn underrun protection will be enabled.

14.1.26 DRBurnVerifyDiscKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a boolean indicating if the disc will be verified after being burned.

If this key is not present, the burn will default to a value of true and the disc will be verified.

14.1.27 DRCDTextKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: This key points to a DRCDTextBlockMBS, or array of DRCDTextBlockMBS objects containing the

CD-Text information for the disc. If this key is not present, the burn will not write CD-Text.

Before using this key, you should to make sure that the device supports CD-Text by checking the value of `DRDeviceCanWriteCDTextKey` in the device's write capabilities dictionary.

If this value is set to true, and the device does not support writing CD-Text, the burn will fail with `kDRDeviceCantWriteCDTextErr`.

14.1.28 `DRErrorStatusAdditionalSenseStringKey` as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: String describing the RBC additional sense code and additional sense code qualifier pair returned by the device. If no sense is reported, this key will not be present.

14.1.29 `DRErrorStatusErrorInfoStringKey` as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: String describing extended error information in a user appropriate manner.

14.1.30 `DRErrorStatusErrorKey` as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: Number containing the OS error code for the error.

14.1.31 `DRErrorStatusErrorStringKey` as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: String describing the error in a user appropriate manner.

14.1.32 DRErrorStatusKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: The key in the status dictionary for the error dictionary.

14.1.33 DRErrorStatusSenseCodeStringKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: String describing the RBC sense code returned by the device. If no sense is reported, this key will not be present.

14.1.34 DRErrorStatusSenseKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: String containing the raw RBC sense information structure reported by the device. If no sense is reported, this key will not be present.

14.1.35 DRMediaCatalogNumberKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a memoryblock containing exactly 13 bytes of data, which will be written to the disc as the Media Catalog Number. If this key is not present, it will default to all zeroes, indicating that a MCN is not supplied.

This value is the UPC/EAN product number, and should conform to the specifications of the UCC and EAN. See ean-int.org and uc-council.org for more details on the UPC/EAN standard.

<http://www.ean-int.org/>

<http://www.uc-council.org/>

14.1.36 DRStatusCurrentSessionKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: Number indicating the current session being burned.

14.1.37 DRStatusCurrentSpeedKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: Number indicating the current burn speed.

14.1.38 DRStatusCurrentTrackKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: Number indicating the current track being burned.

14.1.39 DRStatusEraseTypeKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: String indicating the type of erase operation.

14.1.40 DRStatusPercentCompleteKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: A key for the status dictionaries.

Number containing the percent complete of the operation expressed as a floating point number from 0 to 1.

14.1.41 DRStatusProgressCurrentKPS as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: The current burn speed in kilobytes per second.

Notes: This is an optional key within the DRStatusProgressInfoKey dictionary. The value of this key, if present, is a Number containing the write speed of the burn.

14.1.42 DRStatusProgressCurrentXFactor as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: The current burn speed in a media appropriate x-factor

Notes: This is an optional key within the DRStatusProgressInfoKey dictionary. The value of this key, if present, is a number containing the appropriate x-factor for the media.

14.1.43 DRStatusProgressInfoKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: A dictionary of extended progress information.

A key for the status dictionary. The value of this key is a reference to a Dictionary object containing extended progress information.

14.1.44 DRStatusStateDone as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey. Indicates the operation is finished and it succeeded.

14.1.45 DRStatusStateErasing as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the erase status dictionary.

Notes: Indicates the erase is currently in progress.

14.1.46 DRStatusStateFailed as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey. Indicates the operation is finished and it failed.

14.1.47 DRStatusStateFinishing as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the burn status dictionary.

Notes: Indicates the burn is finishing up (closing the last session, writing the TOC, etc).

14.1.48 DRStatusStateKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: A key for the status dictionaries.

String indicating the current state of the operation.

14.1.49 DRStatusStateNone as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey. Indicates the operation has not yet begun.

14.1.50 DRStatusStatePreparing as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey. Indicates the operation is preparing to begin.

14.1.51 DRStatusStateSessionClose as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the burn status dictionary.

Notes: Indicates the burn is closing a session on disc. The exact session being closing is contained in DRStatusCurrentSessionKey.

14.1.52 DRStatusStateSessionOpen as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the burn status dictionary.

Notes: Indicates the burn is opening a session on disc. The exact session being opened is contained in DRStatusCurrentSessionKey.

14.1.53 DRStatusStateTrackClose as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the burn status dictionary.

Notes: Indicates the burn is closing a track on disc. The exact track being closed is contained in DRStatusCurrentTrackKey.

14.1.54 DRStatusStateTrackOpen as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the burn status dictionary.

Notes: Indicates the burn is opening a track on disc. The exact track being opened is contained in DRStatusCurrentTrackKey.

14.1.55 DRStatusStateTrackWrite as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey in the burn status dictionary.

Notes: Indicates the burn is writing a track on disc. The exact track being written is contained in DRStatusCurrentTrackKey.

14.1.56 DRStatusStateVerifying as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One possible value for DRStatusStateKey. Indicates the operation is verifying what it did.

14.1.57 DRStatusTotalSessionsKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: Number indicating the total number of sessions being burned.

14.1.58 DRStatusTotalTracksKey as string

Plugin Version: 11.0, Platform: macOS, Targets: Desktop only.

Function: One of the dictionary keys for the burn status dictionary.

Notes: Number indicating the total number of tracks in the current session being burned.

14.1.59 DRSynchronousBehaviorKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property key constants.

Notes: The burn property whose value is a BOOL indicating if burn operations will behave synchronously. If this key is not present, it will default to a value of false and burn operations will behave asynchronously.

Synchronous operations do not post status notifications, and will not return until they are completed. Status can still be queried at any time, and will remain valid even after the burn operation has finished.

14.1.60 status as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns a dictionary describing the status of the burn.

Notes: The same dictionary is returned through the DRBurnStatusChangedNotification notification.

14.1.61 writeImageFile(ImageFile as FolderItem) as boolean

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: Writes the image tracks to the disc.

Notes: Requires Mac OS X 10.4.

ImageFile: The path to the image file. This file must be one that can be read by DiscRecording. The supported image types include: .dmg, .iso, .cue, and .toc. For .cue and .toc files the corresponding data files (.bin, .img, etc) must also be present and correctly referenced in the .cue/.toc file.

Returns true on success and false on failure.

See also:

- 14.1. CLASS DRBURNMBS 829
- 14.1.62 writeImageFile(ImagePath as String) as boolean 829

14.1.62 writeImageFile(ImagePath as String) as boolean

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: Writes the image tracks to the disc.

Notes: Requires Mac OS X 10.4.

ImageFile: The path to the image file. This file must be one that can be read by DiscRecording. The supported image types include: .dmg, .iso, .cue, and .toc. For .cue and .toc files the corresponding data files (.bin, .img, etc) must also be present and correctly referenced in the .cue/.toc file.

Returns true on success and false on failure.

See also:

- 14.1.61 writeImageFile(ImageFile as FolderItem) as boolean 828

14.1.63 writeLayout(track as DRTrackMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Writes the track to the disc.

Notes: A single session disc will be created with the given track.

See also:

- 14.1.64 writeLayout(tracks() as DRTrackMBS) 829

14.1.64 writeLayout(tracks() as DRTrackMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Writes the tracks to the disc.

Notes: A multi session disc will be created with the given tracks.

See also:

- 14.1.63 writeLayout(track as DRTrackMBS) 829

14.1.65 Properties

14.1.66 appendable as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Indicates whether the burn is appendable.

Example:

```
dim b as DRBurnMBS // get a burn object
b.appendable=true
```

Notes: When a burn completes, it can mark the disc so that no more data can be written to it. This creates a closed or non-appendable disc (which is the most compatible with audio CD players). If this method returns false, then the disc will be marked as closed and no data can be appended to it. A return value of true indicates further burns can be appended to the disc.
(Read and Write computed property)

14.1.67 BurnFailureAction as string

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: What to do on a failure.

Notes: The burn property whose value is a string containing a one of the failure actions possible for the disc handling.

If this key is not present, the burn will default to a value of DRBurnFailureActionEject and the disc will be ejected.
(Read and Write computed property)

14.1.68 completionAction as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The action to be performed at the end of the burn.

Notes: Value must be DRBurnCompletionActionEject, DRBurnCompletionActionMount or empty for default.
(Read and Write computed property)

14.1.69 DoubleLayerL0DataZoneBlocks as Double

Plugin Version: 8.2, Platform: macOS, Targets: Desktop only.

Function: A double containing the number of blocks desired for the layer 0 data zone on a double layer writable disc.

Notes: The size of the layer 0 data zone dictates where the transition point is from layer 0 to layer 1. If this key is present, the data zone size will be set prior to the start of the burn using the value for this key.

If it is not present, the default layer 0 data zone will be used (half the available blocks on an empty disc). The transition point can be specified two ways. If the value specified in this key is greater than 1.0, then it will designate an absolute block number for the transition point. In this case, the block number should be a multiple of 16 and at least 40000 per specification. If the value is less than 1.0, it will specify the percentage of the burn that should reside on layer 0. A typical value is 0.5, designating half the burn for each layer. A value of 0.0 or 1.0 will not change the layer 0 transition point.
(Read and Write computed property)

14.1.70 MediaCatalogNumber as memoryblock

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: The media catalog number as a binary data string.

Notes: The burn property whose value is a memoryblock containing exactly 13 bytes of data, which will be written to the disc as the Media Catalog Number. If this key is not present, it will default to all zeroes, indicating that a MCN is not supplied.

This value is the UPC/EAN product number, and should conform to the specifications of the UCC and EAN. See

<http://www.ean-int.org/>
and

<http://www.uc-council.org/>
for more details on the UPC/EAN standard.
(Read and Write computed property)

14.1.71 Overwrite as boolean

Plugin Version: 8.2, Platform: macOS, Targets: Desktop only.

Function: The burn property whose value is a boolean indicating if the disc will be overwritten from block zero for the burn.

Notes: If this key is not present, the burn will default to a value of false and the disc will be appended.
(Read and Write computed property)

14.1.72 properties as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The properties dictionary of the burn.

Notes: (Read and Write computed property)

14.1.73 requestedBurnSpeed as single

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The speed at which this burn will attempt to write data.

Notes: The actual speed also depends on the capabilities of the bus the device is on, the maximum speed of the device itself, and the media used.

Value is a float indicating the speed the burn should run at in kilobytes per second.
(Read and Write computed property)

14.1.74 Testing as boolean

Plugin Version: 8.2, Platform: macOS, Targets: Desktop only.

Function: Whether the burn will run as a test burn.

Notes: When this is set and the burn starts, the entire burn process proceeds as if data would be written to the disc, but the laser is not turned on to full power, so the physical disc is not modified.

If this key is not present or the selected burning device does not support test burning, the burn will default to a value of false and a normal burn will occur.
(Read and Write computed property)

14.1.75 UnderrunProtection as boolean

Plugin Version: 8.2, Platform: macOS, Targets: Desktop only.

Function: A boolean indicating if burn underrun protection will be on or off for devices which support it.

Notes: For those devices which support it, burn underrun protection is enabled by default.

If the device supports burn underrun protection and this key is not present, the burn will default to a value of `<i>true</i>` and burn underrun protection will be enabled.
(Read and Write computed property)

14.1.76 verifyDisc as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Indicates whether the resulting disc will be verified.

Example:

```
dim b as DRBurnMBS // get a burn object
b.verifyDisc=true
```

Notes: After data is written to disc, the data can be verified. The verification process will read the data on the disc back into memory and compare it to the data originally used to write to disc. The type of verification is determined by a track property on a track-by-track basis. See the DRTrackMBS documentation for more information on verification types.
(Read and Write computed property)

14.2 class DRBurnProgressPanelMBS

14.2.1 class DRBurnProgressPanelMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Run and display progress while burning data to media.

Example:

```

dim track as DRTrackMBS
dim bsp as DRBurnSetupPanelMBS
dim bpp as DRBurnProgressPanelMBS

// we need a track
track=CreateTrack

if track<>nil then
  bsp=new DRBurnSetupPanelMBS

  // set a few options
  bsp.setCanSelectAppendableMedia true
  bsp.setCanSelectTestBurn true

  if bsp.runSetupPanel=bsp.NSOKButton then
    bpp=new DRBurnProgressPanelMBS

    // And start off the burn itself. This will put up the progress dialog
    // and do all the nice pretty things that a happy app does.
    bpp.beginProgressPanelForBurn bsp.burnObject, track

  else
    MsgBox "You pressed cancel."
  end if
end if

```

Notes: All methods in this class will catch exceptions from Cocoa and raise a `NSExcptionMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Subclass of the `NSPanelMBS` class.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

Xojo Developer Magazine

- [6.1, pages 29 to 30: DiscRecording, How to burn a CD from REALbasic on Mac OS X by Christian Schmitz](#)

14.2.2 Methods

14.2.3 beginProgressPanelForBurn(burn as DRBurnMBS, track as DRTrackMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel on screen and begins the burn process.

Notes: This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns a single session disc with one track.

See also:

- [14.2.4 beginProgressPanelForBurn\(burn as DRBurnMBS, tracks\(\) as DRTrackMBS\)](#) 835

14.2.4 beginProgressPanelForBurn(burn as DRBurnMBS, tracks() as DRTrackMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel on screen and begins the burn process.

Notes: This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns a multi session disc with several tracks.

See also:

- [14.2.3 beginProgressPanelForBurn\(burn as DRBurnMBS, track as DRTrackMBS\)](#) 835

14.2.5 beginProgressPanelForImageFile(burn as DRBurnMBS, file as folderitem) as boolean

Plugin Version: 8.6, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel on screen and begins the burn process.

Notes: This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns the image file located on the given position.

This file must be one that can be read by DiscRecording. The supported image types include: .dmg, .iso, .cue, and .toc. For .cue and .toc files the corresponding data files (.bin, .img, etc) must also be present and correctly referenced in the .cue/.toc file.

See also:

- 14.2.6 `beginProgressPanelForImageFile(burn as DRBurnMBS, file as string)` as boolean 836

14.2.6 `beginProgressPanelForImageFile(burn as DRBurnMBS, file as string)` as boolean

Plugin Version: 8.6, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel on screen and begins the burn process.

Notes: This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns the image file located on the given position.

This file must be one that can be read by DiscRecording. The supported image types include: .dmg, .iso, .cue, and .toc. For .cue and .toc files the corresponding data files (.bin, .img, etc) must also be present and correctly referenced in the .cue/.toc file.

See also:

- 14.2.5 `beginProgressPanelForImageFile(burn as DRBurnMBS, file as folderitem)` as boolean 835

14.2.7 `beginProgressSheetForBurn(burn as DRBurnMBS, track as DRTrackMBS, docWindow as NSWindowMBS)`

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel as a sheet and begins the burn process.

Notes: `docWindow`: The window the sheet will be attached to. If `docWindow` is not nil, the panel slides down as a sheet running as a document modal window. If `owner` is nil, this is an error.

This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns a single session disc with one track.

See also:

- 14.2.8 `beginProgressSheetForBurn(burn as DRBurnMBS, tracks() as DRTrackMBS, docWindow as NSWindowMBS)` 837

14.2.8 beginProgressSheetForBurn(burn as DRBurnMBS, tracks() as DRTrackMBS, docWindow as NSWindowMBS)

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel as a sheet and begins the burn process.

Notes: docWindow: The window the sheet will be attached to. If docWindow is not nil, the panel slides down as a sheet running as a document modal window. If owner is nil, this is an error.

This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns a multi session disc with several tracks.

See also:

- 14.2.7 beginProgressSheetForBurn(burn as DRBurnMBS, track as DRTrackMBS, docWindow as NSWindowMBS) 836

14.2.9 beginProgressSheetForImageFile(burn as DRBurnMBS, file as folderitem, docWindow as NSWindowMBS) as boolean

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel as a sheet and begins the burn process.

Notes: docWindow: The window the sheet will be attached to. If docWindow is not nil, the panel slides down as a sheet running as a document modal window. If owner is nil, this is an error.

This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns the image file located on the given position.

This file must be one that can be read by DiscRecording. The supported image types include: .dmg, .iso, .cue, and .toc. For .cue and .toc files the corresponding data files (.bin, .img, etc) must also be present and correctly referenced in the .cue/.toc file.

See also:

- 14.2.10 beginProgressSheetForImageFile(burn as DRBurnMBS, file as string, docWindow as NSWindowMBS) as boolean 838

14.2.10 `beginProgressSheetForImageFile(burn as DRBurnMBS, file as string, docWindow as NSWindowMBS) as boolean`

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel as a sheet and begins the burn process.

Notes: `docWindow`: The window the sheet will be attached to. If `docWindow` is not nil, the panel slides down as a sheet running as a document modal window. If `owner` is nil, this is an error.

This method returns control to the caller after it has displayed the progress sheet and begun the burn. Once the method has returned the caller can perform other operations while the burn continues.

Burns the image file located on the given position.

This file must be one that can be read by DiscRecording. The supported image types include: `.dmg`, `.iso`, `.cue`, and `.toc`. For `.cue` and `.toc` files the corresponding data files (`.bin`, `.img`, etc) must also be present and correctly referenced in the `.cue/.toc` file.

See also:

- 14.2.9 `beginProgressSheetForImageFile(burn as DRBurnMBS, file as folderitem, docWindow as NSWindowMBS) as boolean` 837

14.2.11 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The constructor to create a new burn progress panel.

14.2.12 `DRBurnProgressPanelDidFinishNotification as string`

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names for use with a burn progress panel.

Notes: Posted when the `DRBurnProgressPanel` has finished and is about to go away.

This notification contains a notification object but no `userInfo` dictionary. The notification object is the `DRBurnProgressPanel` that will be closed.

14.2.13 `DRBurnProgressPanelWillBeginNotification as string`

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names for use with a burn progress panel.

Notes: Posted when the DRBurnProgressPanel is about to begin displaying progress. This notification contains a notification object but no userInfo dictionary. The notification object is the DRBurnProgressPanel that will be displayed.

14.2.14 stopBurn

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's stop button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.2.15 Properties

14.2.16 Description as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The description string displayed in the panel.

Notes: If no description is explicitly set, this method will return the standard text string.

The panel's description is typically a short text string that gives an indication to the user what operation is being performed. If no description is explicitly set, the progress panel uses a standard text string suitable to the burn.

(Read and Write computed property)

14.2.17 VerboseProgressStatus as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The verbosity of the progress feedback.

Notes: If verbose is true, the panel will update status for every change. If verbose is false, the panel will filter some status messages and only update for major changes. The default for the panel is filter the status messages.

(Read and Write computed property)

14.2.18 Events

14.2.19 `burnProgressPanelBurnDidFinish(burn as DRBurnMBS)` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Allows the delegate to handle the end-of-burn feedback.

Notes: This method allows the delegate to handle or modify the end-of-burn feedback performed by the progress panel. Return true to indicate the delegate handled the burn completion and the standard feedback should be suppressed. If this method returns false, the normal end-of-burn handling is performed (displaying an error if appropriate, playing an "I'm done" sound, etc). The delegate is messaged before the progress panel is ordered out so a sheet may be displayed on a progress panel displayed as a window.

14.2.20 `burnProgressPanelDidFinish`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Notification sent by the panel after ordering out.

Notes: If the delegate implements this method it will receive the message after the panel is removed from display.

14.2.21 `burnProgressPanelWillBegin`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Notification sent by the panel before display.

Notes: If the delegate implements this method it will receive the message immediately before the panel is displayed.

14.3 class DRBurnSetupPanelMBS

14.3.1 class DRBurnSetupPanelMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Manages a panel that allows users to specify the parameters of an burn.

Example:

```
dim track as DRTrackMBS
dim bsp as DRBurnSetupPanelMBS
dim bpp as DRBurnProgressPanelMBS

// we need a track
track=CreateTrack

if track<>nil then
    bsp=new DRBurnSetupPanelMBS

    // set a few options
    bsp.setCanSelectAppendableMedia true
    bsp.setCanSelectTestBurn true

    if bsp.runSetupPanel=bsp.NSOKButton then
        bpp=new DRBurnProgressPanelMBS

        // And start off the burn itself. This will put up the progress dialog
        // and do all the nice pretty things that a happy app does.
        bpp.beginProgressPanelForBurn bsp.burnObject, track

    else
        MsgBox "You pressed cancel."
    end if
end if
```

Notes: This class supports choosing the the device to use, whether or not to verify the burned data and how to handle the burned disc when it completes.

All methods in this class will catch exceptions from Cocoa and raise a NSEExceptionMBS instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Subclass of the DRSetupPanelMBS class.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)

- [MBS REALbasic Plugins, version 10.4pr7](#)

Xojo Developer Magazine

- [6.1, pages 29 to 30: DiscRecording, How to burn a CD from REALbasic on Mac OS X by Christian Schmitz](#)

14.3.2 Methods

14.3.3 appendable

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's appendable checkbox.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.3.4 burnObject as DRBurnMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a new DRBurn object that's configured to write data to the currently selected device.

Notes: The new DRBurn object is configured based on the settings in the setup panel when the user clicks the OK button.

Do not invoke this method within a modal session because the burn object information is only updated just before the modal session ends.

Returns nil on any error.

14.3.5 burnSpeed

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's burn speed popup button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.3.6 completionAction

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks one of the panel's completion action radio buttons.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.3.7 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The constructor to create a new burn setup panel.

14.3.8 DRBurnSetupPanelDefaultButtonDefaultTitle as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The constant to use for the build title so it is replaced by the real default title string.

14.3.9 expand

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's expand button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.3.10 setCanSelectAppendableMedia(flag as boolean)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Specifies whether the user can choose to leave the disc appendable.

Notes: This method controls whether the appendable checkbox is enabled.

If the data being written to disc does not lend itself to having more data appended on to it, you can disable the ability of the user to leave the disc open.

This method must be called before the panel is displayed.

Set to yes to enable the appendable checkbox, false to disable.

14.3.11 setCanSelectTestBurn(flag as boolean)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Specifies whether the user can choose to make a test burn.

Notes: This method controls whether a checkbox should be added to the receiver that allows the user to set the burn to be a test burn. By default, the test burn button is not displayed.

This method must be called before the panel is displayed.

Set to yes to show the test burn checkbox, false to hide it.

14.3.12 setDefaultButtonTitle(title as string)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Sets the title for the receiver's default button to title.

Example:

```
dim panel as new DRBurnSetupPanelMBS
```

```
panel.setDefaultButtonTitle "Do Burn"
```

Notes: Normally, the default button is "Burn".

14.3.13 testBurn

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's test burn checkbox.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.3.14 verifyBurn

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's verify burn checkbox.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.4 class DRCDTextBlockMBS

14.4.1 class DRCDTextBlockMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: A class for a Text block.

Notes: All methods in this class will catch exceptions from Cocoa and raise a `NSEExceptionMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

14.4.2 Methods

14.4.3 Constructor

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: The private constructor.

14.4.4 encoding as Integer

Plugin Version: 8.1, Platform: macOS, Targets: Desktop only.

Function: The text encoding to use.

14.4.5 language as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The language of this text block.

14.5 class DRDeviceMBS

14.5.1 class DRDeviceMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Represents a physical CD/DVD drive connected to the computer.

Notes: All methods in this class will catch exceptions from Cocoa and raise a `NSExcptionMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Real Studio Plugins, version 11.3pr1](#)
- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

Xojo Developer Magazine

- [6.1, page 30: DiscRecording, How to burn a CD from REALbasic on Mac OS X by Christian Schmitz](#)

14.5.2 Methods

14.5.3 acquireExclusiveAccess as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Attempts to acquire an exclusive access session with the device.

Notes: Acquiring exclusive access to the device prevents any process other than the one acquiring access from communicating with the device. So once exclusive access is granted, the device is unusable by any other process. Because of this all volumes mounted from media in the drive must be unmounted before exclusive access can be granted.

Exclusive access can be acquired multiple times. Each time this method is called, a call to `releaseExclusiveAccess` must be made at a later time, otherwise the process will never release its exclusive access.

Returns true if the exclusive access is acquired and false if not.

14.5.4 acquireMediaReservation

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Indicate an interest in the blank media reservation.

Example:

```
dim d as DRDeviceMBS // get a device
d.AcquireMediaReservation
```

Notes: Blank media participates in a reservation system that allows applications to express their claim on blank media to other applications. Indicating an interest in the reservation isn't enough to assume its been acquired, as there are likely to be other applications in the system whom have also indicated an interest in the blank media reservation.

14.5.5 `bsdName` as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the `bsd /dev` node name.

14.5.6 `closeTray` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Commands the device to close its tray.

Notes: Does nothing if the device does not have a tray (slotload).

Returns true if the tray could be closed and false if not.

14.5.7 Constructor

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: The private constructor.

14.5.8 `device(index as UInt32)` as DRDeviceMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Queries the device with the given index.

Notes: Index from 0 to `deviceCount-1`.

Same as `devices()`, but if you only need one, this method is more efficient.

14.5.9 deviceCount as UInt32

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: Queries the number of devices on this computer.

14.5.10 deviceForBSDName(bsdName as string) as DRDeviceMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Obtains a DRDevice for the device corresponding to the bsd /dev node.

Notes: If the device is not an authoring device (i.e., CDR, CDRW, DVR-R, etc), returns nil.

14.5.11 deviceForIORegistryEntryPath(path as string) as DRDeviceMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Obtains a DRDevice for the device at the path.

Notes: If the device is not an authoring device (i.e., CDR, CDRW, DVR-R, etc), returns nil.

14.5.12 devices as DRDeviceMBS()

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Obtains a static list of devices connected to the computer.

Example:

```
// shows all devices with their product names:
dim devices() as DRDeviceMBS = DRDeviceMBS.devices

for each dd as DRDeviceMBS in devices
MsgBox dd.info.lookup(DRDeviceMBS.DRDeviceProductNameKey, "?")
next
```

Notes: Returns all CD/DVD devices connected to the computer at the time this method is called. Since devices can come and go at any time, the output of this method is simply a snapshot of the set of devices connected.

14.5.13 displayName as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns a string suitable for display in the user interface.

14.5.14 DRDeviceAppearedNotification as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names used with a DRBurn object.

Notes: Posted by a DRNotificationCenter when a device is added to the system.

This notification is registered for only by name.

The object associated with this notification is the the device that has appeared. The userInfo is the same dictionary returned by info for that device.

14.5.15 DRDeviceBurnSpeedBD1x as single

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible burn speed values.

Notes: 4496.0 KB/sec.

14.5.16 DRDeviceBurnSpeedCD1x as single

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible burn speed values.

Notes: 176.4 KB/sec.

14.5.17 DRDeviceBurnSpeedDVD1x as single

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible burn speed values.

Notes: 1385.0 KB/sec.

14.5.18 DRDeviceBurnSpeedHDDVD1x as single

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible burn speed values.

Notes: 4568.0 KB/sec.

14.5.19 DRDeviceBurnSpeedMax as single

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible burn speed values.

Notes: A value representing the maximum speed at which a device can burn. The actual speed will vary from device to device.

14.5.20 DRDeviceBurnSpeedsKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: An array containing the possible burn speeds available to use. This key may not be present if no media is inserted.

14.5.21 DRDeviceCanTestWriteCDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Value for this key is a boolean value indicating whether the device can perform a test burn to CD media.

14.5.22 DRDeviceCanTestWriteDVDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: A boolean value indicating whether the device can perform a test burn to DVD media.

14.5.23 DRDeviceCanUnderrunProtectCDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device supports burn underrun protection when writing to CD media.

14.5.24 DRDeviceCanUnderrunProtectDVDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device supports burn underrun protection when writing to DVD media.

14.5.25 DRDeviceCanWriteBDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to some type of BD (Blu-ray) based media.

14.5.26 DRDeviceCanWriteBDREKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to BD-RE media.

14.5.27 DRDeviceCanWriteBDRKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to BD-R media.

14.5.28 DRDeviceCanWriteCDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to some type of CD based media.

14.5.29 `DRDeviceCanWriteCDRawKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device supports a raw mode burn strategy for CD. Raw mode is sometimes incorrectly referred to as DAO (disc-at-once).

14.5.30 `DRDeviceCanWriteCDRKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to CD-R media.

14.5.31 `DRDeviceCanWriteCDRWKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to CD-RW media.

14.5.32 `DRDeviceCanWriteCDSAOKKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device supports a SAO (session-at-once) burn strategy for CD.

14.5.33 `DRDeviceCanWriteCDTAOKKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device supports a TAO (track-at-once) burn strategy for CD.

14.5.34 DRDeviceCanWriteCDTextKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write CD-Text information to media.

14.5.35 DRDeviceCanWriteDVDDAOKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device supports a DAO (disc-at-once) burn strategy on DVD media.

14.5.36 DRDeviceCanWriteDVDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to some type of DVD based media.

14.5.37 DRDeviceCanWriteDVDPlusRDoubleLayerKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to DVD+R DL media.

14.5.38 DRDeviceCanWriteDVDPlusRKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to DVD+R media.

14.5.39 DRDeviceCanWriteDVDPlusRWDDoubleLayerKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to DVD+RW DL media.

14.5.40 `DRDeviceCanWriteDVDPlusRWKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to DVD+RW media.

14.5.41 `DRDeviceCanWriteDVDRAMKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to DVD-RAM media.

14.5.42 `DRDeviceCanWriteDVDRDualLayerKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to DVD-R DL media.

14.5.43 `DRDeviceCanWriteDVDRKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to DVD-R media.

14.5.44 `DRDeviceCanWriteDVDRWDualLayerKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to DVD-RW DL media.

14.5.45 DRDeviceCanWriteDVDRWKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to DVD-RW media.

14.5.46 DRDeviceCanWriteHDDVDKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to some type of HDDVD based media.

14.5.47 DRDeviceCanWriteHDDVDRAMKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to HD DVD-RAM media.

14.5.48 DRDeviceCanWriteHDDVDRDualLayerKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to HD DVD-R DL media.

14.5.49 DRDeviceCanWriteHDDVDRKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceWriteCapabilitiesKey dictionary.

Notes: Boolean value indicating whether the device can write to HD DVD-R media.

14.5.50 DRDeviceCanWriteHDDVDRWDualLayerKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to HD DVD-RW DL media.

14.5.51 `DRDeviceCanWriteHDDVDRWKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to HD DVD-RW media.

14.5.52 `DRDeviceCanWriteIndexPointsKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write index points to CD media.

14.5.53 `DRDeviceCanWriteISRCKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write ISRC to CD media.

14.5.54 `DRDeviceCanWriteKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceWriteCapabilitiesKey` dictionary.

Notes: Boolean value indicating whether the device can write to some type of media.

14.5.55 `DRDeviceCurrentWriteSpeedKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: Number containing the current burning speed of this device.

14.5.56 DRDeviceDisappearedNotification as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names used with a DRBurn object.

Notes: Posted by a DRNotificationCenter when a device is removed from the system.

The object associated with this notification is the the device that has disappeared. The userInfo is the same dictionary returned by info for that device.

14.5.57 DRDeviceFirmwareRevisionKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: String containing the firmware revision extracted from the device.

14.5.58 DRDeviceIORegistryEntryPathKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: String containing the path of the device in the IO Registry.

Notes: One of the keys in the dictionary returned by the info method.

14.5.59 DRDeviceIsBusyKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: Number containing a boolean value indicating whether the device is busy or not.

14.5.60 DRDeviceIsTrayOpenKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: Number containing a boolean value indicating whether the device's tray is open or not.

14.5.61 `DRDeviceLoadingMechanismCanEjectKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: Number describing if the loading mechanism of the drive can eject.

14.5.62 `DRDeviceLoadingMechanismCanInjectKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: Number describing if the loading mechanism of the drive can inject.

14.5.63 `DRDeviceLoadingMechanismCanOpenKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: Number describing if the loading mechanism of the drive can open.

14.5.64 `DRDeviceMaximumWriteSpeedKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: Number containing the maximum burning speed of this device.

14.5.65 `DRDeviceMediaBlocksFreeKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceMediaInfoKey` dictionary.

Notes: Number containing the amount of space available (in blocks) on the media to be written to. If the media already contains data and this value will be less than the normal maximum size of the disc. This value will normally only be used if the intent is to append data onto an open disc.

14.5.66 `DRDeviceMediaBlocksOverwritableKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing the total amount of writable space available (in blocks) on the media to be written to - if that media can be overwritten. Media that can be overwritten is designated through the DRDeviceMediaIsOverwritableKey .

The overwritable space is the amount of space on the disc that would be available if any data currently on the disc is first erased.

14.5.67 DRDeviceMediaBlocksUsedKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing the amount of space currently used (in blocks) for existing data.

14.5.68 DRDeviceMediaBSDNameKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: String containing the BSD /dev node name assigned to the media in the device.

14.5.69 DRDeviceMediaClassBD as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One possible value of the DRDeviceMediaClassKey.

Notes: Indicates the media is some type of BD (Blu-ray) based media.

14.5.70 DRDeviceMediaClassCD as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One possible value of the DRDeviceMediaClassKey.

Notes: Indicates the media is some type of CD based media.

14.5.71 DRDeviceMediaClassDVD as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One possible value of the `DRDeviceMediaClassKey`.

Notes: Indicates the media is some type of DVD based media.

14.5.72 `DRDeviceMediaClassHDDVD` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One possible value of the `DRDeviceMediaClassKey`.

Notes: Indicates the media is some type of HD DVD based media.

14.5.73 `DRDeviceMediaClassKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceMediaInfoKey` dictionary.

Notes: String containing the class of media present in the drive.

14.5.74 `DRDeviceMediaClassUnknown` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One possible value of the `DRDeviceMediaClassKey`.

Notes: Indicates the media class is unknown.

14.5.75 `DRDeviceMediaDoubleLayerL0DataZoneBlocksKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceMediaInfoKey` dictionary.

Notes: Number containing the amount of space available (in blocks) on layer 0 of a double layer piece of media.

14.5.76 `DRDeviceMediaFreeSpaceKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceMediaInfoKey` dictionary.

Notes: MSF value of the amount of space available on the media to be written to. If the media already contains data and this value will be less than the normal maximum size of the disc. This value will normally

only be used if the intent is to append data onto an open disc.

14.5.77 DRDeviceMediaInfoKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: Dictionary of information describing the media currently in the device. This key may not be present if no media is inserted.

14.5.78 DRDeviceMediaIsAppendableKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing a boolean value indicating that data can be appended to the existing data (if any).

14.5.79 DRDeviceMediaIsBlankKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing a boolean value indicating whether data has previously been written to the media.

14.5.80 DRDeviceMediaIsErasableKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing a boolean value indicating whether this media can be erased.

14.5.81 DRDeviceMediaIsOverwritableKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing a boolean value indicating that the data on the disc (if any) can be overwritten.

Rewritable media can always be erased, and then rewritten in its entirety, so it is always considered overwritable.

Write-once media, if its blank, can also be written in its entirety and is also considered overwritable.

Write-once media, that has been partially written, can never again enter a state where it is entirely writable and will have lost its overwritable designation.

14.5.82 DRDeviceMediaIsReservedKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing a boolean value indicating whether the media is reserved for exclusive use by the current process.

14.5.83 DRDeviceMediaOverwritableSpaceKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: MSF value of the amount of writable space available on the media to be written to - if that media can be overwritten. Media that can be overwritten is designated through the DRDeviceMediaIsOverwritableKey .

The overwritable space is the amount of space on the disc that would be available if any data currently on the disc is first erased.

14.5.84 DRDeviceMediaSessionCountKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing the current number of sessions present on the media.

14.5.85 DRDeviceMediaStateInTransition as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values for the DRDeviceMediaStateKey.

Notes: The media is transitioning from one state to another (i.e., being spun up/down).

14.5.86 DRDeviceMediaStateKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: String describing the state of the media.

14.5.87 DRDeviceMediaStateMediaPresent as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values for the DRDeviceMediaStateKey.

Notes: Device contains media of some type.

14.5.88 DRDeviceMediaStateNone as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values for the DRDeviceMediaStateKey.

Notes: No media is present in the device.

14.5.89 DRDeviceMediaTrackCountKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: Number containing the number of tracks present on the media.

14.5.90 DRDeviceMediaTypeBDR as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a BD-R.

14.5.91 DRDeviceMediaTypeBDRE as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a BD-RE.

14.5.92 DRDeviceMediaTypeBDROM as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a BD-ROM.

14.5.93 DRDeviceMediaTypeCDR as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a CD-R.

14.5.94 DRDeviceMediaTypeCDROM as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a CD-ROM.

14.5.95 DRDeviceMediaTypeCDRW as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a CD-RW.

14.5.96 DRDeviceMediaTypeDVDPlusR as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD+R.

14.5.97 DRDeviceMediaTypeDVDPlusRDoubleLayer as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD+R Double Layer.

14.5.98 DRDeviceMediaTypeDVDPlusRW as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD+RW.

14.5.99 DRDeviceMediaTypeDVDPlusRWDoubleLayer as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD+RW Double Layer.

14.5.100 DRDeviceMediaTypeDVDR as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD+RW.

14.5.101 DRDeviceMediaTypeDVDRAM as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD-RAM.

14.5.102 DRDeviceMediaTypeDVDRDualLayer as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD-R DL.

14.5.103 DRDeviceMediaTypeDVDRROM as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD-ROM.

14.5.104 DRDeviceMediaTypeDVDRW as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD-RW.

14.5.105 DRDeviceMediaTypeDVDRWDualLayer as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a DVD-RW DL.

14.5.106 DRDeviceMediaTypeHDDVDR as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a HD DVD-R.

14.5.107 DRDeviceMediaTypeHDDVDRAM as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDeviceMediaTypeKey.

Notes: Media is a HD DVD-RAM.

14.5.108 DRDeviceMediaTypeHDDVDRDualLayer as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDeviceMediaTypeKey`.

Notes: Media is a HD DVD-R DL.

14.5.109 `DRDeviceMediaTypeHDDVDROM` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDeviceMediaTypeKey`.

Notes: Media is a HD DVD-ROM.

14.5.110 `DRDeviceMediaTypeHDDVDRW` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDeviceMediaTypeKey`.

Notes: Media is a HD DVD-RW.

14.5.111 `DRDeviceMediaTypeHDDVDRWDualLayer` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDeviceMediaTypeKey`.

Notes: Media is a HD DVD-RW DL.

14.5.112 `DRDeviceMediaTypeKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the `DRDeviceMediaInfoKey` dictionary.

Notes: String containing the type of media inserted in the device.

14.5.113 `DRDeviceMediaTypeUnknown` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDeviceMediaTypeKey`.

Notes: The type of the media cannot be determined.

14.5.114 DRDeviceMediaUsedSpaceKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the DRDeviceMediaInfoKey dictionary.

Notes: MSF value of the amount of space currently used for existing data.

14.5.115 DRDevicePhysicalInterconnectATAPI as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDevicePhysicalInterconnectKey.

Notes: Device is connected on an ATAPI interface.

14.5.116 DRDevicePhysicalInterconnectFibreChannel as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDevicePhysicalInterconnectKey.

Notes: Device is connected through a Fibre Channel interface.

14.5.117 DRDevicePhysicalInterconnectFireWire as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the DRDevicePhysicalInterconnectKey.

Notes: Device is connected through a Firewire interface.

14.5.118 DRDevicePhysicalInterconnectKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: String describing the connection of the device to the computer.

14.5.119 DRDevicePhysicalInterconnectLocationExternal as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDevicePhysicalInterconnectLocationKey`.

Notes: Device is connected to the machine externally.

14.5.120 `DRDevicePhysicalInterconnectLocationInternal` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDevicePhysicalInterconnectLocationKey`.

Notes: Device is connected to the machine internally.

14.5.121 `DRDevicePhysicalInterconnectLocationKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the `info` method.

Notes: String describing the location of the device (e.g. `internal/external`).

14.5.122 `DRDevicePhysicalInterconnectLocationUnknown` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDevicePhysicalInterconnectLocationKey`.

Notes: It's not known how the device is connected.

14.5.123 `DRDevicePhysicalInterconnectSCSI` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDevicePhysicalInterconnectKey`.

Notes: Device is connected on a SCSI interface.

14.5.124 `DRDevicePhysicalInterconnectUSB` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the `DRDevicePhysicalInterconnectKey`.

Notes: Device is connected through a USB interface.

14.5.125 DRDeviceProductNameKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Example:

```
// shows all devices with their product names:
dim devices() as DRDeviceMBS = DRDeviceMBS.devices

for each dd as DRDeviceMBS in devices
MsgBox dd.info.lookup(DRDeviceMBS.DRDeviceProductNameKey, "?")
next
```

Notes: String containing the product name extracted from the device.

14.5.126 DRDeviceStatusChangedNotification as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names used with a DRBurn object.

Notes: Posted by a DRNotificationCenter when the media in a device changes state. This can include being ejected, inserted, becoming busy, etc.

The object for this notification is the device who's media is changing state. The userInfo for this notification is the same dictionary returned by status for that device.

14.5.127 DRDeviceSupportLevelAppleShipping as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the values for DRDeviceSupportLevelKey.

Notes: This value indicates this device is shipping in some Apple machine.

14.5.128 DRDeviceSupportLevelAppleSupported as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the values for DRDeviceSupportLevelKey.

Notes: This value indicates this device has been tested by Apple for support.

14.5.129 DRDeviceSupportLevelKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: String describing the support level the device enjoys from the engine.

14.5.130 DRDeviceSupportLevelNone as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the values for DRDeviceSupportLevelKey.

Notes: This value indicates this device is not supported.

14.5.131 DRDeviceSupportLevelUnsupported as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the values for DRDeviceSupportLevelKey.

Notes: This value indicates the device is unsupported, but the engine will try to use it anyway.

14.5.132 DRDeviceSupportLevelVendorSupported as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the values for DRDeviceSupportLevelKey.

Notes: This value indicates this device has been tested by a third party for support.

14.5.133 DRDeviceTrackInfoKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: Dictionary containing dictionaries describing the tracks. DRTracks from the DRDeviceTrackRefsKey are used as keys into this dictionary.

14.5.134 DRDeviceTrackRefsKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the status method.

Notes: An array containing a list of DRTrack objects describing any tracks that are already on the disc.

14.5.135 DRDeviceVendorNameKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: String containing the vendor name extracted from the device.

14.5.136 DRDeviceWriteBufferSizeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: Number containing the size of the write buffer of the device.

14.5.137 DRDeviceWriteCapabilitiesKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the keys in the dictionary returned by the info method.

Notes: Dictionary containing the capabilities of the device for writing different kinds of media.

14.5.138 ejectMedia as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Commands the device to eject the media.

Notes: This command first unmounts any volumes associated with the media and then eject the media from the drive. If the media could not be ejected, most likely this is because a volume associated with the media could not be unmounted.

Returns true if the media could be ejected and false if not.

14.5.139 info as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns a dictionary of information describing the device.

Notes: The information returned include the types of media the device can write to, how it's connected and its identifying information such as the vendor and product name.

14.5.140 `ioRegistryEntryPath` as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the path to the device in the IO Registry.

14.5.141 `isEqualToDevice(value as DRDeviceMBS)` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Compares the receiver to another device.

Notes: Returns true if the receiver is equal to otherDevice.

14.5.142 `isValid` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns whether or not the device represented by the receiver is still attached to the computer.

Notes: Because of the way some physical interconnects work, a device which is unplugged and replugged in does not necessarily look like the same device to the computer and would be invalid in that instance.

Returns true if the device is valid and false if not.

14.5.143 `mediaIsAppendable` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true if the media in the device can have more data appended to any existing data.

14.5.144 `mediaIsBlank` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true the media in the device is blank.

14.5.145 mediaIsBusy as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true if the media is in use by some process - even the one making this call.

14.5.146 mediaIsErasable as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true if the media can be erased (i.e., CD-RW, DVD-RW, etc).

14.5.147 mediaIsOverwritable as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true if the media in the device can be fully (re)written.

14.5.148 mediaIsPresent as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Reports the presence of the media.

14.5.149 mediaIsReserved as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true if the application calling this method currently holds the reservation on the media.

14.5.150 mediaIsTransitioning as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns true if the media is in transition (spinning up or down for example).

14.5.151 `mediaSpaceFree` as `DRMSFMBS`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the amount of free space on the media.

14.5.152 `mediaSpaceOverwritable` as `DRMSFMBS`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the amount of writable space on the media.

14.5.153 `mediaSpaceUsed` as `DRMSFMBS`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the amount of used space on the media.

14.5.154 `mediaType` as `string`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the type of media currently inserted into the device.

14.5.155 `openTray` as `boolean`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Commands the device to open its tray.

Notes: Does nothing if the device does not have a tray (slotload). If there is media in the drive this method will do nothing and return false. In this case use `ejectMedia` to eject the media and open the tray.

Returns true if the tray could be opened and false if not.

14.5.156 `PhysicalInterconnect` as `string`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The connection of the device to the computer.

14.5.157 PhysicalInterconnectLocation as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The location of the device (internal/external/unknown).

Notes: Use the DRDevicePhysicalInterconnectLocation* constants.

14.5.158 releaseExclusiveAccess

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Releases the latest exclusive access request for the device.

Notes: A call to this method must be made for every call to acquireExclusiveAccess, otherwise the process will never release its exclusive access.

14.5.159 releaseMediaReservation

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Releases any media reservation that might be in place for the device.

Notes: If media is inserted and reserved, then the reservation will be passed on to the next process with a reservation request.

14.5.160 status as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns a dictionary of information describing the media in the device.

Notes: In addition to information about the media (type, space available/used, etc), the dictionary returned includes those pieces of information about the device itself which are in part determined by the media (i.e., maximum burn speed).

14.5.161 trayIsOpen as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Reports the tray state of the device.

Notes: Returns true if the device has a tray and it is open.

14.5.162 writesCD as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Reports the device's ability to burn to CD-type media.

Notes: Returns true if the device has the ability to write to CD-R media.

14.5.163 writesDVD as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Reports the device's ability to burn to DVD-type media.

Notes: Returns true if the device has the ability to write to DVD-R media.

14.6 class DRERASEMBS

14.6.1 class DRERASEMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Perform and monitor the erasing a rewritable CD or DVD disc.

Notes: All methods in this class will catch exceptions from Cocoa and raise a `NSExceptionMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

14.6.2 Methods

14.6.3 Constructor(device as DRDeviceMBS)

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Initializes an erase object.

14.6.4 device as DRDeviceMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the device being used for the erase.

14.6.5 DREraseStatusChangedNotification as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names you can use to register notifications on the erase status.

14.6.6 DREraseTypeComplete as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: A constant to be used for the `eraseType` property.

Notes: Configures the erase object to perform a complete erase, erasing every byte on the disk. This oper-

ation is slow (on the order of 30 minutes) to complete.

14.6.7 `DREraseTypeKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property names in the properties dictionary of the erase object.

14.6.8 `DREraseTypeQuick` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: A constant to be used for the `eraseType` property.

Notes: Configures the erase object to perform a quick erase, doing the minimal amount of work to make the disc appear blank. This operation typically takes only a minute or two.

14.6.9 `eraseForDevice(device as DRDeviceMBS)` as `DREraseMBS`

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Creates and returns an erase object.

Notes: Returns nil on any error.

14.6.10 `start`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Begin the process of erasing media.

Notes: This method only kicks off the erase. Once the erasure starts, control returns to the caller.

14.6.11 `status` as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns a dictionary containing the status of the erase.

Notes: The same dictionary is returned through the `DREraseStatusChangedNotification` notification.

14.6.12 Properties

14.6.13 eraseType as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The type of erase to be performed.

Notes: (Read and Write computed property)

14.6.14 properties as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The properties dictionary of the erase.

Notes: (Read and Write computed property)

14.7 class DREraseProgressPanelMBS

14.7.1 class DREraseProgressPanelMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Panel to display progress while erasing media.

Notes: All methods in this class will catch exceptions from Cocoa and raise a `NSExcptionMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Subclass of the `NSPanelMBS` class.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

14.7.2 Methods

14.7.3 beginProgressPanelForErase(erase as DREraseMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel on screen and begins the erase process.

Notes: This method returns control to the caller after it has displayed the progress sheet and begun the erase. Once the method has returned the caller can perform other operations while the erase continues.

14.7.4 beginProgressSheetForErase(erase as DREraseMBS, docWindow as NSWindowMBS)

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Presents the progress panel as a sheet and begins the erase process.

Notes: erase: The object performing the erase.

docWindow: The window the sheet will be attached to. If docWindow is not nil, the panel slides down as a sheet running as a document modal window. If owner is nil, this is an error.

This method returns control to the caller after it has displayed the progress sheet and begun the erase. Once the method has returned the caller can perform other operations while the erase continues.

14.7.5 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The constructor which creates the erase progress panel.

14.7.6 DREraseProgressPanelDidFinishNotification as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names for use with an erase progress panel.

Notes: Posted when the DREraseProgressPanel has finished and is about to go away.

This notification contains a notification object but no userInfo dictionary. The notification object is the DREraseProgressPanel that will be closed.

14.7.7 DREraseProgressPanelWillBeginNotification as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the notification names for use with an erase progress panel.

Notes: Posted when the DREraseProgressPanel is about to begin displaying progress.

This notification contains a notification object but no userInfo dictionary. The notification object is the DREraseProgressPanel that will be displayed.

14.7.8 Properties

14.7.9 Description as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The description string displayed in the panel.

Notes: If no description is explicitly set, this method will return the standard text string.

The panel's description is typically a short text string that gives an indication to the user what operation is being performed. If no description is explicitly set, the progress panel uses a standard text string suitable to the erase.

(Read and Write computed property)

14.7.10 Events

14.7.11 `eraseProgressPanelDidFinish`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Notification sent by the panel after ordering out.

14.7.12 `eraseProgressPanelEraseDidFinish(erase as DREraseMBS)` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Allows the delegate to handle the end-of-erase feedback.

Notes: This method allows the delegate to handle or modify the end-of-erase feedback performed by the progress panel. Return true to indicate the delegate handled the erase completion and the standard feedback should be suppressed. If this method returns false, the normal end-of-erase handling is performed (displaying an error if appropriate, playing an "I'm done" sound, etc). The delegate is messaged before the progress panel is ordered out so a sheet may be displayed on a progress panel displayed as a window.

14.7.13 `eraseProgressPanelWillBegin`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Notification sent by the panel before display.

14.8 class DREraseSetupPanelMBS

14.8.1 class DREraseSetupPanelMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Manages a panel that allows users to specify the parameters of an erase.

Notes: This class supports choosing the device to use and what sort of erase to perform. When the panel is closed by the user choosing to erase the media in the device, the device is exclusively held by the application for its own use to prevent possible bad or corrupt media from causing problem for the rest of the system. This means that if the erase object obtained from the panel is not used to do an erase, the device will remain unavailable to other applications until the exclusive access is released.

Subclass of the DRSetupPanelMBS class.

14.8.2 Methods

14.8.3 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The constructor to create the erase setup panel.

Example:

```
dim d as DREraseSetupPanelMBS
```

```
d=new DREraseSetupPanelMBS  
call d.runSetupPanel
```

14.8.4 eraseObject as DREraseMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates and returns a new DRErase object that's configured to erase the disc in the currently selected device.

Notes: The new DRErase object is configured based on the settings in the setup panel when the user clicks the OK button.

Do not invoke this method within a modal session runSetupPanel because the erase object information is only updated just before the modal session ends.

Returns a new DRErase object.

14.9 class DRFileMBS

14.9.1 class DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Represents a file to be created on the disc.

Notes: A file can be either a pointer to an existing file (residing on a hard drive for example) or can be created at burn time from data passed into the file object as requested. DRFiles can only exist inside of DRFolder objects.

All methods in this class will catch exceptions from Cocoa and raise a `NSErrorMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Subclass of the `DRFSObjectMBS` class.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr8](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

Xojo Developer Magazine

- [6.1, page 47: Finding Work, How to Find Work as a REALbasic Developer](#)
- [6.1, page 31: DiscRecording, How to burn a CD from REALbasic on Mac OS X by Christian Schmitz](#)

14.9.2 Methods

14.9.3 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Dummy constructor.

See also:

- 14.9.4 `Constructor(name as string)` 887
- 14.9.5 `Constructor(name as string, data as memoryblock)` 887
- 14.9.6 `Constructor(path as folderitem)` 887

14.9.4 Constructor(name as string)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a "virtual" file object.

Notes: This type of DRFile burns the data produced to the output disc, creating a file with the passed in name.

You need to subclass the DRFileMBS class and handle the events.

See also:

- 14.9.3 Constructor 886
- 14.9.5 Constructor(name as string, data as memoryblock) 887
- 14.9.6 Constructor(path as folderitem) 887

14.9.5 Constructor(name as string, data as memoryblock)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a "virtual" file object.

Notes: This type of DRFile burns the data passed in to disc, creating a file with the passed in name.

See also:

- 14.9.3 Constructor 886
- 14.9.4 Constructor(name as string) 887
- 14.9.6 Constructor(path as folderitem) 887

14.9.6 Constructor(path as folderitem)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Initializes a "real" file object

Notes: This type of DRFile reads in data from an existing file located at path and burns that data to disc.

See also:

- 14.9.3 Constructor 886
- 14.9.4 Constructor(name as string) 887
- 14.9.5 Constructor(name as string, data as memoryblock) 887

14.9.7 DRLinkTypeFinderAlias as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The constant used with `linkWithLinkType` to create a Finder alias.

14.9.8 DRLinkTypeHardLink as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The constant used with `linkWithLinkType` to create a hard link.

14.9.9 DRLinkTypeSymbolicLink as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The constant used with `linkWithLinkType` to create a symbolic link.

14.9.10 fileWithPath(path as folderitem) as DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a "real" file object.

Example:

```
dim file as FolderItem = GetFolderItem("/System/Library/Fonts/Helvetica.dfont", FolderItem.PathTypeShell)
dim f as DRFileMBS = DRFileMBS.fileWithPath(file)
```

```
MsgBox f.baseName
```

Notes: This type of `DRFile` reads in data from an existing file located at `path` and burns that data to disc. See also:

- 14.9.11 `fileWithPath(path as string) as DRFileMBS`

888

14.9.11 fileWithPath(path as string) as DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a "real" file object.

Example:

```
dim f as DRFileMBS = DRFileMBS.fileWithPath("/System/Library/Fonts/Helvetica.dfont")
```

```
MsgBox f.baseName
```

Notes: This type of DRFile reads in data from an existing file located at path and burns that data to disc.
See also:

- 14.9.10 fileWithPath(path as folderitem) as DRFileMBS 888

14.9.12 finderAliasPointingTo(original as DRFSObjectMBS, filesystem as string) as DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a Finder alias to another file on the output disc.

Notes: original: The file to point he hard link to

filesystem: The filesystem this link will exist on.

As with Mac OS X 10.6.4 this method seems to create an empty file only (no alias).

14.9.13 hardLinkPointingTo(original as DRFileMBS, filesystem as string) as DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a hard link to another file on the output disc.

Notes: original: The file to point he hard link to

filesystem: The filesystem this link will exist on.

As with Mac OS X 10.6.4 this method seems to crash always.

14.9.14 linkWithLinkType(linkType as string, original as DRFSObjectMBS, filesystem as string) as DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Initializes a file object to point to another file on the output disc.

Example:

```
dim SomeFolder as new DRFolderMBS("Some new folder")
```

```
// Let's have a file to test

dim data as string = "Hello World" // file content
dim name as string = "test.txt" // file name
dim testfile as DRFileMBS = DRFileMBS.virtualFileWithName(name, data)

// and create a sym link for it
dim SymLink as DRFileMBS = DRFileMBS.linkWithLinkType(DRFileMBS.DRLinkTypeSymbolicLink, testfile, DRFileMBS.DRHFSPPlus)
if SymLink = nil then
  MsgBox "SymLink is nil!"
else
  SomeFolder.addChild SymLink
end if
```

Notes: linkType: The type of link that will be created.
 original: The file to point the hard link to
 filesystem: The filesystem this link will exist on.

14.9.15 symLinkPointingTo(original as DRFSObjectMBS, filesystem as string) as DRFileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a symbolic link to another file on the output disc.

Example:

```
dim SomeFolder as new DRFolderMBS("Some new folder")

// Let's have a file to test

dim data as string = "Hello World" // file content
dim name as string = "test.txt" // file name
dim testfile as DRFileMBS = DRFileMBS.virtualFileWithName(name, data)

// and create a sym link for it
dim SymLink as DRFileMBS = DRFileMBS.symLinkPointingTo(testfile, DRFileMBS.DRHFSPPlus)
if SymLink = nil then
  MsgBox "SymLink is nil!"
else
  SomeFolder.addChild SymLink
end if
```

Notes: original: The file to point he hard link to
 filesystem: The filesystem this link will exist on.

As with Mac OS X 10.6.4 this method seems to work just fine.

14.9.16 virtualFileWithName(name as string, data as memoryblock) as DR-FileMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a "virtual" file object.

Example:

```
dim data as string = "Hello World" // file content
dim name as string = "test.txt" // file name

dim f as DRFileMBS = DRFileMBS.virtualFileWithName(name, data)

MsgBox f.baseName
```

Notes: This type of DRFile burns the data passed in to disc, creating a file with the passed in name.

14.9.17 Events

14.9.18 calculateSizeOfFile(fork as Integer, estimating as boolean) as uint64

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Calculates the size of a file's fork.

Notes: This method may be sent at any time after the file object has been instantiated. Requests that the reciever calculate the file size of file fork (for instance data or resource).

If estimate is true, you are being asked for an estimate of the final fork size, perhaps to provide an estimate of the track size, and do not have to be exact. Estimates should err on the high side; it's better to overestimate than underestimate.

An estimate call may be made at any time.

If estimate is false, you are being asked for the actual fork size, to be used in the burn. This call is only made in the burn phase.

fork: The fork of the file whose size is to be calculated.

estimate: If the file size should be estimated or exact.

Return the length of the file's fork.

14.9.19 cleanupFileAfterBurn

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Cleanup the file once the burn is complete.

Notes: Sent to the receiver after the burn completes. This would be an appropriate place to close files, or do any other teardown work needed. This message will always be sent regardless of whether the burn succeeded or failed.

14.9.20 prepareFileForBurn as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Prepare the file object for burning.

Notes: Sent to the receiver before the burn begins. This would be an appropriate method to open files, or do any other prep work needed. The disc's entire hierarchy is completely in place and can be queried if needed.

After this call, the burn's content is locked down, and you should be able to respond to the calculateSizeOfFile messages with exact values.

14.9.21 prepareFileForVerification as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Prepare the file object for verification.

Notes: Called during the burn (after production, before the cleanupFileAfterBurn event is called) to indicate that verification is about to begin. Now would be a good time to rewind to the start of the file, reset state machines, or do whatever else is needed to prepare to produce again.

14.9.22 produceFile(fork as Integer, buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32) as uint32

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Calculates the size of a file's fork.

Notes: Sent during the burn (after the prepareFileForBurn message) requesting that the receiver produce the data fork contents.

The receiver should fill up the buffer passed in as full as possible and then return control to the caller. Since while burning, keeping the drive's buffer full is of utmost importance, you should not perform lengthy operations or block for data in this method. This method should return the number of bytes actually in the buffer or 0 to indicate that there was an error producing the data.

You may be asked to produce twice, once during the actual burn and once during verification.

fork: The fork of the file to produce.

buffer: The buffer to produce data into.

bufferLength: The length of the buffer to produce data into

blockSize: The size of the track blocks

Return the number of bytes produced.

14.9.23 Constants

Constants

Constant	Value	Description
DRFileForkData	0	The constant which specifies which fork is currently needed. The data fork contains the primary information for the file and is the fork used for files such as JPEGs, text files, etc. The resource fork contains secondary meta-data, which is not important to the primary content of the file and may safely be ignored when the file is sent to a filesystem or OS which does not support multiple forks.
DRFileForkResource	1	The constant which specifies which fork is currently needed. The data fork contains the primary information for the file and is the fork used for files such as JPEGs, text files, etc. The resource fork contains secondary meta-data, which is not important to the primary content of the file and may safely be ignored when the file is sent to a filesystem or OS which does not support multiple forks.

14.10 class DRFolderMBS

14.10.1 class DRFolderMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: A class for folder objects used in filesystem creation.

Notes: A DRFile object is a subclass of DRFSObject and represents a folder on the finished disc. DRFolders can be either a pointer to an existing folder (residing on a hard drive for example) or can be a "virtual" folder which existsonly on the resulting burned disc. A DRFolder pointing to an existing folder ("real" folder) cannot have it's contents changed - only those files/folders which are children of the actual folder on disk will be included on the resulting disc. "Virtual" folders are entirely created programatically and any virtual folder structure can exist and be burned to disc. It is possible to convert a "real" folder to a "virtual" folder using the makeVirtual method.

All methods in this class will catch exceptions from Cocoa and raise a NSExceptionMBS instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Subclass of the DRFSObjectMBS class.

Xojo Developer Magazine

- [6.1, page 47: Finding Work, How to Find Work as a REALbasic Developer](#)
- [6.1, pages 29 to 31: DiscRecording, How to burn a CD from REALbasic on Mac OS X by Christian Schmitz](#)

14.10.2 Methods

14.10.3 addChild(child as DRFSObjectMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Adds an object reference (either a file or folder) as a child of a virtual folder object.

Notes: This method only applies to virtual folders. Real folders are considered "leaf nodes" and cannot have children.

14.10.4 children as DRFSObjectMBS()

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: Returns an array containing the children of a virtual folder.

Notes: The order of children in the array is arbitrary – since the various filesystems being generated may have different sorting requirements, there is no one true way to sort the children. The ordering will change only when children are added or removed. You should sort the children according to the needs of your display, and in a consistent manner. This function only applies to virtual folders. Real folders are considered

”leaf nodes” and should not be passed into this call.

14.10.5 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The dummy constructor.

See also:

- 14.10.6 Constructor(name as string) 895
- 14.10.7 Constructor(path as folderitem) 895

14.10.6 Constructor(name as string)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a ”virtual” folder object with a name.

See also:

- 14.10.5 Constructor 895
- 14.10.7 Constructor(path as folderitem) 895

14.10.7 Constructor(path as folderitem)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Initializes a DRFolder object that will use the folder contents of the folder located at path as a source.

See also:

- 14.10.5 Constructor 895
- 14.10.6 Constructor(name as string) 895

14.10.8 count as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of children of a virtual folder.

Notes: This method returns a ”shallow” count of only those children that are immediately contained within the virtual folder. This method only applies to virtual folders. Real folders are considered ”leaf nodes” and should not be passed into this call.

14.10.9 folderWithPath(path as folderitem) as DRFolderMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Initializes a DRFolder object that will use the folder contents of the folder located at path as a source.

See also:

- 14.10.10 folderWithPath(path as string) as DRFolderMBS 896

14.10.10 folderWithPath(path as string) as DRFolderMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Initializes a DRFolder object that will use the folder contents of the folder located at path as a source.

See also:

- 14.10.9 folderWithPath(path as folderitem) as DRFolderMBS 896

14.10.11 makeVirtual

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Changes the real DRFolder object into a "virtual" DRFolder object.

Notes: The virtual folder created in this way is a snapshot of the on-disk folder at the moment of the call. The newly created virtual folder will contain real folder and file objects corresponding to the on-disk children of the original on-disk folder. If the on-disk folder is modified (eg, if the folder attributes change, or if children are added to or removed from the on-disk tree): during this call, the virtual folder may or may not reflect the changes. If modified after this call, the virtual folder will not reflect the changes.

14.10.12 removeChild(child as DRFSObjectMBS)

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Removes an object reference (either a file or folder) as a child of a virtual folder object.

Notes: This method only applies to virtual folders. Real folders are considered "leaf nodes" and cannot have children.

14.10.13 virtualFolderWithName(name as string) as DRFolderMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a "virtual" folder object with a name.

14.11 class DRFSObjectMBS

14.11.1 class DRFSObjectMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: A class which defines common features of all filesystem content objects.

Notes: Please read Apples documentation for more details.

All methods in this class will catch exceptions from Cocoa and raise a NSExcptionMBS instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

14.11.2 Methods

14.11.3 Constructor

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: The private constructor.

14.11.4 DRAccessDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's last-accessed date.

14.11.5 DRAllFilesystems as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the name or properties for the file in all filesystems together.

Notes: When this key is used to refer to a name, it refers to the base name (which has no naming restrictions).

14.11.6 DRAttributeModificationDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's attribute modification date.

14.11.7 DRBackupDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's backup date.

14.11.8 DRContentModificationDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's content modification date.

14.11.9 DRCreationDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's creation date.

14.11.10 DREffectiveDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's effective date.

14.11.11 DRExpirationDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's expiration date.

14.11.12 DRHFSPPlus as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the HFS+ name/properties for the file.

Notes: HFS+ names can be up to 255 decomposed unicode characters long.

14.11.13 DRHFSPPlusCatalogNodeID as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing item's catalog node ID (HFS+ only). Currently, this value if set is only a suggestion. The burn engine will attempt to use this node ID, but may use another value if it needs to resolve conflicts. Default behavior is to allocate node IDs incrementally from `kHFSFirstUserCatalogNodeID`.

14.11.14 DRHFSPPlusTextEncodingHint as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the item's text encoding hint (HFS+ only).

This value is used by the MacOS to help when converting the natively UTF-16 filename into an 8-bit-per-character representation (such as MacRoman, Shift-JIS, or UTF8). If not set, default behavior is to call `CFStringGetMostCompatibleMacStringEncoding` (`CFStringGetSmallestEncoding()`).

14.11.15 DRInvisible as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Boolean indicating whether the item is invisible or not.

14.11.16 DRISO9660 as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the ISO-9660 properties for the file.

Notes: This key cannot be used to refer to the name of the file; it is ambiguous, since the name may be in either level 1 or level 2 format.

14.11.17 DRISO9660LevelOne as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the ISO-9660 level 1 name for the file.

Notes: This key is used to refer specifically to the name generated for ISO-9660 if the ISO level is set to 1. When used for a property, it is equivalent in use to the DRISO9660 key and acts as a synonym for that key.

ISO9660 level 1 names are in the form typically known as 8.3 - eight characters of name and three characters of extension (if it's a file; directories can't have extensions). Character set is limited to A-Z, 0-9, and _.

14.11.18 DRISO9660LevelTwo as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the ISO-9660 level 2 name for the file.

Notes: This key is used to refer specifically to the name generated for ISO-9660 if the ISO level is set to 2. When used for a property, it is equivalent in use to the DRISO9660 key and acts as a synonym for that key.

ISO9660 level 2 names can be 32 chars long, are limited to a subset of the 7-bit ASCII chars (capital letters, numbers, space, punctuation), and are only allowed one "." character.

14.11.19 DRISO9660VersionNumber as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the ISO9660 version number for the object. Default value is 1.

14.11.20 DRJoliet as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the Joliet name/properties for the file.

Notes: Joliet names can be 64 precomposed unicode characters long, but are only allowed one ”” character and many punctuation characters are illegal.

14.11.21 DRMacExtendedFinderFlags as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the item’s extended Finder flags (MacOS only).

14.11.22 DRMacFileCreator as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: A binary string containing the OSType for the file creator (MacOS only).

14.11.23 DRMacFileType as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: A binary string containing the OSType for the file type (MacOS only).

14.11.24 DRMacFinderFlags as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the item’s Finder flags (MacOS only). The invisible bit is ignored - use DRInvisible instead.

14.11.25 DRMacFinderHideExtension as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: A boolean indicating whether the extension should be hidden in the Finder or not. The default is

false and only applies to files.

14.11.26 DRMacIconLocation as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: A binary string containing a Point (not NSPoint) for the item's icon location in its parent folder (MacOS only).

14.11.27 DRMacScrollPosition as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Binary string containing a Point (not NSPoint) for the folder's scroll position (MacOS only).

14.11.28 DRMacWindowBounds as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Binary string containing a Rect (not NSRect) for the window bounds for a folder (MacOS only).

14.11.29 DRMacWindowView as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the folder's window view type (MacOS only).

14.11.30 DRPosixFileMode as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the item's POSIX file mode.

14.11.31 DRPosixGID as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the item's POSIX GID.

14.11.32 DRPosixUID as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number containing the item's POSIX UID.

14.11.33 DRRecordingDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Date containing the item's recording date.

14.11.34 DRUDF as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The key for accessing the UDF name/properties for the file.

14.11.35 DRUDFApplicationIdentifierSuffix as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. Binary string of up to 8 bytes in length, for application use. The presence of this key requires the DRApplicationIdentifier key.

14.11.36 DRUDFExtendedFilePermissions as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number indicating the extended UDF file permissions of this file.

Bit 0: Change attributes for others (low order bit) Bit 1: Delete permissions for others Bit 2: Change attributes for group Bit 3: Delete permissions for group Bit 4: Change attributes for owner Bit 5: Delete permissions for owner Bit 6 & 7: Reserved If this key is not present, DRPosixFileMode will be used with the above bits being set to the corresponding write bit for owner, group, and others. If DRPosixFileMode is not present, the file mode from the file on disc will be used, again using the write bit for these permissions.

14.11.37 DRUDFInterchangeLevel as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. Number containing the volume interchange level. See the UDF specs for details.

14.11.38 DRUDFMaxInterchangeLevel as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. Number containing the maximum volume interchange level number. See the UDF specs for details.

14.11.39 DRUDFMaxVolumeSequenceNumber as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. Number containing the maximum volume sequence number. See the UDF specs for details.

14.11.40 DRUDFPrimaryVolumeDescriptorNumber as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. Number containing the primary volume sequence number. See the UDF specs for details.

14.11.41 DRUDFRealTimeFile as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Number indicating whether the file is a UDF Real-Time file.

14.11.42 DRUDFVersion102 as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: This value is used in DRUDFWriteVersion.

14.11.43 DRUDFVersion150 as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: This value is used in DRUDFWriteVersion.

14.11.44 DRUDFVolumeSequenceNumber as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. Number containing the volume sequence number. See the UDF specs for details.

14.11.45 DRUDFVolumeSetIdentifier as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. The Volume Set Identifier for the UDF volume set. If this key is not present, DRVOLUMESET will be used if present. The Volume Set Identifier is composed of the Volume Set Timestamp, the Implementation Use, and a the string contained in this property.

14.11.46 DRUDFVolumeSetImplementationUse as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. A binary string (8 bytes in length) for implementation use data. See the UDF specs for details.

14.11.47 DRUDFVolumeSetTimestamp as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. A date object for the volume set timestamp. See the UDF specs for details.

14.11.48 DRUDFWriteVersion as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the property keys.

Notes: Optional key. This property key defines the version for the UDF structures written to disk. Values are defined in UDF Version types.

14.11.49 effectiveFilesystemMask as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The effective filesystem mask set for the receiver.

14.11.50 isVirtual as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Indicates whether the receiver is real or virtual.

Notes:

True if the receiver is virtual and false if real.

14.11.51 mangledNameForFilesystem(filesystem as string) as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: A single mangled filesystem-specific name for this fobject.

Notes: Use DRISO9660LevelOne, DRISO9660, DRHFSPPlus, DRUDF, DRJoliet, DRISO9660LevelTwo or

DRAllFilesystems for the filesystem parameter.

14.11.52 `mangledNames` as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns a dictionary containing all of the filesystem-specific names for the receiver, each one mangled for uniqueness.

Notes: The dictionary will return only the names which are indicated by the receiver's effective mask. If the receiver's effective mask is zero, an empty dictionary is returned.

14.11.53 `parent` as DRFolderMBS

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The parent object.

14.11.54 `propertiesForFilesystem(filesystem as string, mergeWithOtherFilesystems as boolean)` as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns all the filesystem properties set for the specified filesystem.

Notes: filesystem: The filesystem to look in.

merge: If true, also look for properties in the umbrella DRAllFilesystems .

Normally you would call this method with merge set to true since you are interested in the set of properties that will be used when writing the object to disc. But if you have a need to determine what properties are set just for a specific filesystem, then pass in false for merge. In this case only the specific filesystem is checked. So if filesystem is set to DRHFSPlus and merge is false then the properties dictionary contains the values set for the HFS+ filesystem only. If no properties have been directly set for HFS+ yet, then this properties dictionary will be empty.

14.11.55 `propertyForKey(key as string, filesystem as string, mergeWithOtherFilesystems as boolean)` as Variant

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Returns a file/folder property specified by key for the specified filesystem.

Notes: key: The property to return.

filesystem: The filesystem to look in.

merge: If true, also look for the property in the umbrella DRAllFilesystems .

Returns the value associated with the property.

Normally you would call this method with merge set to true since you are interested in the property that will be used when writing the object to disc. But if you have a need to determine what property is set just for a specific filesystem, then pass in false for merge. In this case only the specific filesystem is checked. So if DRHFSPPlus is passed in for filesystem and merge is false then the property returned is the value set for the HFS+ filesystem only. If that property has not been directly set for HFS+ yet, then the returned value will be nil.

14.11.56 setProperties(Value as dictionary, filesystem as string)

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Sets the value of all the receiver's properties specified by the keys in properties for the specific filesystem.

Notes: properties: The value of the property.

filesystem: The filesystem to set the property in.

The properties are set only in the filesystem dictionary specified by filesystem. DRAllFilesystems may be specified as the filesystem in which case the umbrella property affecting all filesystems at once will be set. Setting properties for DRAllFilesystems does not preclude setting a filesystem specific property.

14.11.57 setProperty(Value as Variant, key as string, filesystem as string)

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Sets the value of the receiver's property specified by key for the specific filesystem.

Notes: property: The value of the property.

key: The property key.

filesystem: The filesystem to set the property in.

The property is set only in the filesystem dictionary specified by filesystem. DRAllFilesystems may be specified as the filesystem in which case the umbrella property affecting all filesystems at once will be set. Setting a property for DRAllFilesystems does not preclude setting a filesystem specific property.

14.11.58 `sourcePath` as string

Plugin Version: 11.2, Platform: macOS, Targets: Desktop only.

Function: The source path.

14.11.59 Properties

14.11.60 `baseName` as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the base name for the receiver.

Notes: The base name is the name from which any necessary filesystem-specific names are automatically generated.

Because the content creation API is able to generate multiple filesystems which require multiple varied naming conventions, a sensible system for naming is required. Thus each file has a "base name" which corresponds to its default name in any filesystem.

Whenever possible, the "base name" will be used in the generated filesystem without modification. If the name cannot be used as-is (if, for example, it contains illegal characters, exceeds the length requirements, doesn't meet the required format, or a name collision is detected) then an acceptable name that meets the filesystem's criteria will be generated automatically from the base name.

The default base name for a real file or folder is the actual on-disk name of the item.
(Read and Write computed property)

14.11.61 `explicitFilesystemMask` as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The explicit filesystem mask set for the receiver.

Notes: (Read and Write computed property)

14.11.62 `specificNameForFilesystem(filesystem as string)` as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: A single filesystem-specific name for this fsubject.

Notes: Use `DRISO9660LevelOne`, `DRISO9660`, `DRHFSPlus`, `DRUDF`, `DRJoliet`, `DRISO9660LevelTwo` or

DRAllFilesystems for the filesystem parameter.
(Read and Write computed property)

14.11.63 specificNames as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The names used for the receiver in the different filesystems all at once.

Notes: An Dictionary of filesystem keys with corresponding name strings as their values for each specific filesystem name that should be set.

Every effort will be made to use the names passed in. However, if a name is illegal, it will be modified to fit the rules for that filesystem's names. Because of this, you should always call specificNames after setSpecificNames: to ensure that you are always displaying the most current names to the user.

(Read and Write computed property)

14.11.64 Constants

Constants

Constant	Value	Description
DRFilesystemInclusionMaskHFSPlus	8	The value which indicates the object should be included in the HFS+ filesystem.
DRFilesystemInclusionMaskISO9660	1	The value which indicates the object should be included in the ISO filesystem.
DRFilesystemInclusionMaskJoliet	2	The value which indicates the object should be included in the Joliet filesystem.
DRFilesystemInclusionMaskUDF	4	The value which indicates the object should be included in the UDF filesystem.

14.12 class DRMSFMBS

14.12.1 class DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Minutes/Seconds/Frames handling.

Notes: On CDs, minutes/seconds/frames are used to identify positioning on a disc (which is most useful on an audio disc), but applies to any position on a disc no matter what type of data is present.

A frame is equivalent to a sector or block in normal disk parlance. 75 frames make up one second, so a 2 second pause (typical pregap size) is 150 frames.

All methods in this class will catch exceptions from Cocoa and raise a `NSErrorMBS` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

14.12.2 Methods

14.12.3 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The dummy constructor.

See also:

- 14.12.4 `Constructor(frames as Integer)` 912
- 14.12.5 `Constructor(s as string)` 912

14.12.4 `Constructor(frames as Integer)`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates an `msf` object whose length is frames.

See also:

- 14.12.3 `Constructor` 912
- 14.12.5 `Constructor(s as string)` 912

14.12.5 `Constructor(s as string)`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

14.12. CLASS DRMSFMBS 913

Function: Creates an msf object initialized to the value represented by string

See also:

- 14.12.3 Constructor 912
- 14.12.4 Constructor(frames as Integer) 912

14.12.6 description as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns a textual representation.

14.12.7 descriptionWithFormat(format as string) as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns a textual representation of the receiver.

Notes: The format string is very similar to

A printf-style format string with %-escaped formatting characters.

%%	A "%
%m	Minutes as a decimal number
%s	Seconds as a decimal number
%f	Frames as a decimal number

In addition to these formatting characters an optional length specifier can come between the % and the formatting character. This length specifier will force the field in question to be at least that wide. For example a format specifier of "%02m:%02s" will cause a DRMSF object representing 3 minutes 9 seconds to be formatted as "03:09".

A formatter is aware of and respects rounding. If a bit of the msf is non zero, but the format does not display that value, the next higher value will be increased by one to reflect that. Extending our example above, an DRMSF with a value of 3 minutes, 9 seconds, 15 frames using a format specifier of "%02m:%02s", will be formatted as "03:10" since the 15 frames rounds up the seconds to the next value

Returns a string containing a textual representation of the object utilizing the specified format.

14.12.8 frames as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of frames represented by the receiver.

Notes: This method differs from sectors in that it returns to the caller the number of frames remaining in the current second. For example an DRMSF value of 5:30:72 will return 72 from frames.

14.12.9 isEqualToMSF(value as DRMSFMBS) as boolean

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Compares on emsf to another.

Notes: True if the two object are equal, false otherwise.

14.12.10 minutes as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of minutes represented by the receiver.

Notes: If the receiver represents a non integral number of minutes, only the whole minute value is returned. For example an DRMSF value of 5:30:72 will return 5 from a minutes.

14.12.11 msf as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates an msf object with no length/time.

Notes: Returns nil on any error.

14.12.12 msfByAdding(value as DRMSFMBS) as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Adds an msf to the receiver.

Notes: Returns nil on any error.

14.12.13 msfBySubtracting(value as DRMSFMBS) as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Subtracts an msf to the receiver.

Notes: Returns nil on any error.

14.12.14 msfWithFrames(frames as Integer) as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates an msf object whose length is frames.

Notes: Returns nil on any error.

14.12.15 msfWithString(s as string) as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates an msf object initialized to the value represented by string.

Notes: Returns nil on any error.

14.12.16 seconds as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the number of seconds represented by the receiver.

Notes: If the receiver represents a non integral number of seconds, only the whole second value is returned. For example an DRMSF value of 5:30:72 will return 30 from seconds.

14.12.17 sectors as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the total number of frames/sectors represented by the receiver.

Notes: This method differs from frames in that it returns to the caller the total number of frames/sectors represented by the object.

For example an DRMSF value of 5:30:72 will return 24822 from sectors.

14.13 class DRSetupPanelMBS

14.13.1 class DRSetupPanelMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Base class for the DiscRecordingUI setup panels.

Notes: Provides a base framework for handling device selection, media ejection and confirming or cancelling the panel.

All methods in this class will catch exceptions from Cocoa and raise a `NSError` instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Subclass of the `NSPanelMBS` class.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)
- [MBS REALbasic Plugins, version 10.4pr7](#)

14.13.2 Methods

14.13.3 cancel

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's cancel button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.13.4 close

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's close button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.13.5 Constructor

Plugin Version: 13.1, Platform: macOS, Targets: Desktop only.

Function: The private constructor.

14.13.6 eject

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's eject button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.13.7 ok

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's default button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.13.8 open

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: Invoked when the user clicks the panel's open button.

Notes: If you overwrite this method in Xojo, your method will not be called. You can only call this method to trigger the same behavior as if the user clicked the control.

14.13.9 runSetupPanel as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Displays the panel and begins its event loop.

Notes: Returns NSOKButton (if the user clicks the default button) or NSCancelButton (if the user clicks the Cancel button).

14.13.10 Events

14.13.11 `determineBestDevice(deviceA as DRDeviceMBS, deviceB as DRDeviceMBS) as DRDeviceMBS`

Plugin Version: 10.4, Platform: macOS, Targets: .

Function: Called to let you decide which device is better to use.

Notes: Default returns device A.

14.13.12 `DeviceContainsSuitableMedia(device as DRDeviceMBS, byref prompt as string) as boolean`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: This delegate method allows the delegate to determine if the media inserted in the device is suitable for whatever operation is to be performed.

Notes:

device: The device that contains the media being asked about.
prompt: Pass back a string object describing the media state.

Return false to disable the default button.

14.13.13 `DeviceCouldBeTarget(device as DRDeviceMBS) as boolean`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Allows the delegate to determine if device can be used as a target.

Notes: This method is used to limit the menu to only those devices that you want to appear. For example, a DVD burning application might use this to limit the menu to only devices that are capable of writing DVD-Rs.

device: The candidate device.

Returns true if the device is acceptable, false if not.

14.13.14 `DeviceSelectionChanged(device as DRDeviceMBS)`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Sent by the default notification center when the device selection in the panel has changed.

14.13.15 SetupPanelShouldHandleMediaReservations as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: This event allows the delegate to control how media reservations are handled.

Notes: Return false to indicate the delegate will handle media reservations. Return true to indicate the setupPanel should handle media reservations itself.

14.13.16 Constants

Constants

Constant	Value	Description
NSCancelButton	0	The value runSetupPanel returns if the Cancel button is clicked.
NSOKButton	1	The value runSetupPanel returns if the OK button is clicked.

14.14 class DRTrackMBS

14.14.1 class DRTrackMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The DRTrack class represents a track on the burned disc.

Example:

```
dim track as DRTrackMBS
dim bsp as DRBurnSetupPanelMBS
dim bpp as DRBurnProgressPanelMBS

// we need a track
track=CreateTrack

if track<>nil then
    bsp=new DRBurnSetupPanelMBS

    // set a few options
    bsp.setCanSelectAppendableMedia true
    bsp.setCanSelectTestBurn true

    if bsp.runSetupPanel=bsp.NSOKButton then
        bpp=new DRBurnProgressPanelMBS

        // And start off the burn itself. This will put up the progress dialog
        // and do all the nice pretty things that a happy app does.
        bpp.beginProgressPanelForBurn bsp.burnObject, track

    else
        MsgBox "You pressed cancel."
    end if
end if
```

Notes: A DRTrack provides data to the for the burn and contains a description of the track on disc (length, block type, data format, etc). Data is provided for the burn in a real-time thread. It is up to the track to provide this data in a timely manner, otherwise a burn underrun can occur and ruin a disc.

All methods in this class will catch exceptions from Cocoa and raise a NSExcptionMBS instead. Using the message, name and reason properties you can see what was the reason for this exception. Please report if you find a method which does not handle exceptions correct.

Blog Entries

- [MBS REALbasic Plugins Version 10.4 release notes](#)

- [MBS REALbasic Plugins, version 10.4pr7](#)
- [MBS Plugins 10.3 Release Notes](#)

Xojo Developer Magazine

- [6.1, page 47: Finding Work, How to Find Work as a REALbasic Developer](#)
- [6.1, pages 29 to 31: DiscRecording, How to burn a CD from REALbasic on Mac OS X by Christian Schmitz](#)

14.14.2 Methods

14.14.3 Constructor

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Dummy constructor.

14.14.4 DRAbstractFile as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: DRFileMBS object pointing to the abstract file for ISO and Joliet volumes. The file must be in the root directory.

14.14.5 DRApplicationIdentifier as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the application identifier for ISO, Joliet and UDF volumes.

14.14.6 DRAudioFourChannelKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD-DA audio tracks only. This key points to a boolean value indicating whether the track data has four channels, as opposed to the two channels of normal stereo. If this key is not present, the engine will use a default value of false and standard two-channel stereo is assumed.

Note that while four-channel is technically allowed in the Red Book, it never caught on and is probably being replaced by SACD, so you probably shouldn't attempt to use it.

On the media, this key corresponds to bit 3 of the control field in sub-channel Q.

14.14.7 `DRAudioPreEmphasisKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD-DA audio tracks only. This key points to a boolean value indicating whether the track includes pre-emphasis of 50/15us. If this key is not present, the engine will use a default value of false. On the media, this key corresponds to bit 0 of the control field in sub-channel Q.

14.14.8 `DRBibliographicFile` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: DRFile pointing to the bibliographic file for ISO and Joliet volumes. The file must be in the root directory.

14.14.9 `DRBlockSize` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number indicating the block size of the track. Currently always 2048. Do not change.

14.14.10 `DRBlockSizeKey` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the size in bytes of each track block. See the Mt. Fuji (INF-8090i) specification for CD/DVD devices for possible values for this property.

14.14.11 DRBlockTypeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the type of each track block. See the Mt. Fuji (INF-8090i) specification for CD/DVD devices for possible values for this property.

14.14.12 DRCopyrightFile as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: DRFileMBS pointing to the copyright file for ISO and Joliet volumes. The file must be in the root directory.

14.14.13 DRDataFormKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the data form of each block in the track. See the Mt. Fuji (INF-8090i) specification for CD/DVD devices for possible values for this property.

14.14.14 DRDataPreparer as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the data preparer for ISO and Joliet volumes.

14.14.15 DRDefaultDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Date containing the default date to use for any unspecified dates in the filesystem. The current date and time is used if unspecified.

14.14.16 DRDVDCopyrightInfoKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For DVD tracks only. A binary string containing the DVD copyright info structure sent through the SEND DVD STRUCTURE command. The contents of this will be sent directly to the drive.

14.14.17 DRDVDTimestampKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For DVD tracks only. Binary string containing the DVD timestamp structure sent through the SEND DVD STRUCTURE command. The contents of this will be sent directly to the drive.

14.14.18 DRFreeBlocksKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the length, in blocks, which is still available in a writable track.

14.14.19 DRIndexPointsKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD tracks only. This key points to an NSArray of NSNumbers, indicating the index points inside the track. Each index point is specified as a number of blocks (frames) relative to the start of the track. There are 75 blocks in one second of CD audio. No more than 98 index points may be specified for a track.

Not all drives are capable of writing index points, and not all consumer CD players report or use them. If this key is present in any track and the drive cannot write index points, the burn will fail with `kDRDeviceCantWriteIndexPointsErr`.

14.14.20 DRISOLevel as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the ISO level of the ISO-9660 filesystem on the track. Currently should be 1 or 2.

14.14.21 DRISOMacExtensions as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Boolean indicating whether the track should have Mac extensions.

14.14.22 DRISORockRidgeExtensions as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Boolean indicating whether the track should have RockRidge (POSIX) extensions.

14.14.23 DRMaxBurnSpeedKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the maximum burn speed at which data can be produced. The speed is represented in KB/s (1 KB = 1000 bytes). This key can only be used to limit the speed at which the burn runs.

14.14.24 DRNextWritableAddressKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the LBA of the next writable address in the track. This key is not present in closed tracks.

14.14.25 DRPreGapIsRequiredKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD tracks only. Number indicating whether the pregap listed for the track is required. If this

key is not present, the track will behave as though the key were false.

If this key's value is set to true and the device does not support the exact pregap length, the burn will fail with a return value of `kDRDevicePregapLengthNotAvailableErr` .

If this key's value is set to true and the device does not support any of the suggested pregap length, the engine will choose an alternate pregap length.

14.14.26 DRPreGapLengthKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD tracks only. Number containing the length of silence or data at the beginning of the track. This defaults to 2 seconds of silence. If this key is present, the track producer will be asked for the pregap data first. If the producer implements the proper selector, then it's the responsibility of the producer to provide data for the pregap, otherwise that length of silence will be produced by Disc Recording.

14.14.27 DRPublisher as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the publisher for ISO and Joliet volumes.

14.14.28 DRSCMSCopyrightFree as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: One possible value for the `DRSerialCopyManagementStateKey`. Indicates that the track has no copying restrictions. Copies of this track should also be copyright free.

14.14.29 DRSCMSCopyrightProtectedCopy as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: One possible value for the `DRSerialCopyManagementStateKey`. Indicates that the track is a first-generation copy of an original that was subject to copy protection. No further digital copying should be

allowed.

14.14.30 DRSCMSCopyrightProtectedOriginal as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: One possible value for the DRSerialCopyManagementStateKey. Indicates that the track is an original subject to copyright protection. Digital copying of this track should be allowed, but copies should be marked with SCMS.

14.14.31 DRSerialCopyManagementStateKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD tracks only. This key points to a string value indicating the SCMS state of the track. If this key is not present, no SCMS data is written.

Not all drives are capable of writing SCMS. If this key is present in any track and the drive cannot write SCMS, the burn will fail with kDRDeviceCantWriteSCMSErr.

14.14.32 DRSessionFormatKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the session format of the track. See the Mt. Fuji (INF-8090i) specification for CD/DVD devices for possible values for this property.

14.14.33 DRSessionNumberKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the physical session number of a track.

14.14.34 DRSubchannelDataFormKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: The property whose value is the data mode of the subchannel data sent to the drive. If this key is not present, the track will default to a value of DRSubchannelDataFormNone and no subchannel information will be requested from the producer.

Subchannel data is returned from the producer in the same production method that produces normal user data. Normally a producer returns user data in chunks of DRBlockSizeKey size. When subchannel data is also produced, the producer is expected to return user data of DRBlockSizeKey in length with an additional 96 bytes of subchannel data. Depending on the data form specified in this key, the format of this 96 bytes is either in raw or pack format. When subchannel data is requested from the producer, the DRFlagSubchannelDataRequested flag is set in the flags parameter of producePreGapForTrack or produceDataForTrack and blockSize is increased by 96 bytes.

14.14.35 DRSubchannelDataFormNone as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: A value for DRSubchannelDataFormKey indicating that the subchannel data will not be provided by the producer.

14.14.36 DRSubchannelDataFormPack as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: A value for DRSubchannelDataFormKey indicating that the producer will be asked to provide pack format subchannel data for the track. If this form is selected, the drive will perform P and Q parity calculations on each pack and interleave the packs before writing them to disc. This corresponds to a subchannel data form of 0xC0.

14.14.37 DRSubchannelDataFormRaw as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: A value for DRSubchannelDataFormKey indicating that the producer will be asked to provide raw format subchannel data for the track. If this form is selected, the producer must have performed P and Q parity calculations for each pack and done proper interleaving of the subchannel data. The drive will fill in the P-Q subchannel information and write the R-W subchannel data as is to the disc. This corresponds to

a subchannel data form of &h40.

14.14.38 DRSuppressMacSpecificFiles as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Boolean indicating whether the track should suppress Mac-specific files from non-HFS filesystems.

14.14.39 DRSystemIdentifier as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the system identifier for ISO and Joliet volumes.

14.14.40 DRTrackIsEmptyKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing a boolean value and indicates whether the track is empty.

14.14.41 DRTrackISRCKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: For CD-DA audio tracks only. This key points to an memoryblock containing exactly 12 bytes, which will be written to the disc as the International Standard Recording Code (ISRC). If this key is not present, no ISRC is written.

The use of this value should conform to the specifications of the IFPI. More details:

<http://www.ifpi.org/isrc/>

Not all drives are capable of the write modes necessary to write the ISRC. If this key is present in any track and the drive cannot write the ISRC, the burn will fail with `kDRDeviceCantWriteISRC`Err.

14.14.42 DRTrackLengthKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number representing the length of the track.

14.14.43 DRTrackModeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the track mode of the track. See the Mt. Fuji (INF-8090i) specification for CD/DVD devices for possible values for this property.

14.14.44 DRTrackNumberKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the physical track number of a track.

14.14.45 DRTrackPacketSizeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the number of blocks per packet for the disc. It will only be present if the disc contains fixed packets. This key will contain 16 for DVD media, and typically contains either 16 or 32 for CD media.

14.14.46 DRTrackPacketTypeFixed as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: If this is the value of the DRTrackPacketTypeKey then the disc is written with fixed sized packets. When this value is present the DRPacketSizeKey will also be present.

14.14.47 DRTrackPacketTypeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the kind of packets being written.

14.14.48 DRTrackPacketTypeVariable as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: If this is the value of the DRTrackPacketTypeKey then the disc is written with sequential variable sized packets. The presence of this value indicates the lack of the DRPacketSizeKey.

14.14.49 DRTrackStartAddressKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Number containing the LBA of the start address for the track.

14.14.50 DRTrackTypeClosed as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: If this is the value of the DRTrackTypeKey then the track has been written and is closed.

14.14.51 DRTrackTypeIncomplete as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: If this is the value of the DRTrackTypeKey then the track is not invisible or reserved and is available for writing.

14.14.52 DRTrackTypeInvisible as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: If this is the value of the DRTrackTypeKey then the track is invisible and available to writing. If

it is packet written and not closed, DRPacketTypeKey will be present, along with DRTrackPacketType and DRTrackPacketSize keys.

14.14.53 DRTrackTypeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the type of track. Possible values are: DRTrackTypeInvisible , DRTrackTypeIncomplete , DRTrackTypeReserved or DRTrackTypeClosed.

14.14.54 DRTrackTypeReserved as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: If this is the value of the DRTrackTypeKey then the track is reserved for writing.

14.14.55 DRVerificationTypeChecksum as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the VerificationType.

Notes: The engine will verify the track data with an internally calculated checksum.

14.14.56 DRVerificationTypeKey as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the type of verification to be performed. If this is not present, the track will not be verified.

14.14.57 DRVerificationTypeNone as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the VerificationType property.

Notes: No verification is desired, so verification will be skipped.

14.14.58 DRVerificationTypeProduceAgain as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the VerificationType property.

Notes: The engine will simply begin another production cycle and start calling produceDataForTrack again.

14.14.59 DRVerificationTypeReceiveData as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the possible values of the VerificationType property.

Notes: The engine will begin reading data from the disc and calling verifyDataForTrack.

14.14.60 DRVolumeCheckedDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Date containing the volume-checked date for HFS+ volumes. DRDefaultDate is used if unspecified.

14.14.61 DRVolumeCreationDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Date containing the volume creation date. DRDefaultDate is used if unspecified.

14.14.62 DRVolumeEffectiveDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Date containing the date and time at which the volume is effective for ISO and Joliet volumes.

14.14.63 DRVolumeExpirationDate as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Date containing the volume expiration date for ISO and Joliet volumes.

14.14.64 `DRVolumeModificationDate` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: Date containing the volume modification date. `DRDefaultDate` is used if unspecified.

14.14.65 `DRVolumeSet` as string

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: One of the key constants for the properties dictionary.

Notes: String indicating the volume set name for ISO and Joliet volumes.

14.14.66 `estimateLength` as `UInt64`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Asks the track producer for a size estimate.

Notes: This method calls the track producer to ask it to estimate the size needed for its data.

For some types of track, this call may be very expensive. For example, a `DRFilesystemTrack` may need to iterate folders on disk to provide an accurate estimate, which (if a large number of files and folders are involved) can cause this call to take 30 seconds or more. Since your main thread should not be allowed to block for this long, you may wish to call this function on a separate thread.

Requires Mac OS X 10.3.

14.14.67 `testProductionSpeedForInterval(seconds as Double)` as `Double`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Tests the production speed for a specified interval.

Notes:

Runs a fake "production" cycle, repeatedly asking the receiver for data by calling `it's producer's produceDataIntoBuffer` for the specified time interval.

Use this function to verify that the the production code can produce data fast enough to satisfy the data throughput requirements of the burn.

Returns the calculated maximum speed the at which the receiver can produce data. This value should be used when setting up a burn to limit the burn speed

14.14.68 testProductionSpeedForLength(length as Integer) as Double

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Tests the production speed for a specified byte count.

Notes: Runs a fake "production" cycle, repeatedly asking the receiver for data by calling it's producer's produceDataIntoBuffer:length:atAddress:blockSize:ioFlags: until the specified length number of bytes have been produced.

Use this function to verify that the the production code can produce data fast enough to satisfy the data throughput requirements of the burn.

Returns the calculated maximum speed the at which the receiver can produce data. This value should be used when setting up a burn to limit the burn speed

Length: length of test in seconds.

14.14.69 trackForAudioFile(path as folderitem) as DRTrackMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates an audio track capable of burning RedBook CD audio from a file.

Example:

```
dim f as FolderItem
f=SelectFolder
if f<>nil then
// Create tracks for the audio files
dim tracks(-1) as DRTrackMBS
dim c as Integer=f.Count
for i as Integer=1 to c
dim g as FolderItem=f.Item(i)
if g<>nil and g.visible then
dim track as DRTrackMBS=DRTrackMBS.trackForAudioFile(g)
if track<>nil then
tracks.Append track
end if
```

```

end if
next

// display gui
if UBound(tracks)>=0 then
dim bsp as DRBurnSetupPanelMBS=new DRBurnSetupPanelMBS // you may want to use your own subclass to catch events
if bsp.runSetupPanel=bsp.NSOKButton then
dim bpp as new DRBurnProgressPanelMBS // you may want to use your own subclass to catch events
bpp.beginProgressPanelForBurn(bsp.burnObject, tracks)
end if
end if
end if

```

Notes: This function creates a track object configured and primed to output RedBook audio CD data. It accepts any file readable by QuickTime and extracts the audio data (if any) from the file, translating that into the correct format for output to the disc.

Returns nil on any error. Requires Mac OS X 10.3.
See also:

- 14.14.70 `trackForAudioFile(path as string) as DRTrackMBS` 936

14.14.70 `trackForAudioFile(path as string) as DRTrackMBS`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates an audio track capable of burning RedBook CD audio from a file.

Notes: This function creates a track object configured and primed to output RedBook audio CD data. It accepts any file readable by QuickTime and extracts the audio data (if any) from the file, translating that into the correct format for output to the disc.

Returns nil on any error. Requires Mac OS X 10.3.
See also:

- 14.14.69 `trackForAudioFile(path as folderitem) as DRTrackMBS` 935

14.14.71 `trackForRootFolder(folder as DRFolderMBS) as DRTrackMBS`

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a filesystem track capable of burning a folder.

Notes: Returns a track which contains the given folder as root.

See also:

14.14. CLASS DRTRACKMBS

937

- 14.14.72 trackForRootFolder(folder as folderitem) as DRTrackMBS

937

14.14.72 trackForRootFolder(folder as folderitem) as DRTrackMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Creates a filesystem track capable of burning a folder.

Notes: Returns a track which contains the given folder as root.

See also:

- 14.14.71 trackForRootFolder(folder as DRFolderMBS) as DRTrackMBS

936

14.14.73 Properties

14.14.74 BlockSize as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The size in bytes of each track block.

Notes: (Read and Write computed property)

14.14.75 BlockType as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The type of each track block.

Notes: (Read and Write computed property)

14.14.76 DataForm as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The data form of each block in the track.

Notes: (Read and Write computed property)

14.14.77 length as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: Returns the length of the track data.

Notes: The length returned does not include the length of the pregap. Only the length of the track data

itself is returned.

(Read and Write computed property)

14.14.78 MaxBurnSpeed as Double

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: A number containing the maximum burn speed at which data can be produced.

Notes: The speed is represented in KB/s (1 KB = 1000 bytes). This value can only be used to limit the speed at which the burn runs.

(Read and Write computed property)

14.14.79 preGap as DRMSFMBS

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The length of the pre gap.

Notes: (Read and Write computed property)

14.14.80 PreGapsRequired as boolean

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: A boolean indicating whether the pregap listed for the track is required.

Notes: For CD tracks only.

If this value is not set, the track will behave as though the value were false.

If this value is set to true and the device does not support the exact pregap length, the burn will fail with a return value of `kDRDevicePregapLengthNotAvailableErr`.

If this value is set to true and the device does not support any of the suggested pregap length, the engine will choose an alternate pregap length.

(Read and Write computed property)

14.14.81 PreGapLength as Double

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: A number containing the length of silence or data at the beginning of the track.

Notes: For CD tracks only.

This defaults to 2 seconds of silence. If this value is set, the track producer will be asked for the pregap data first.

If the producer implements the proper event, then it's the responsibility of the producer to provide data for the pregap, otherwise that length of silence will be produced by Disc Recording.

(Read and Write computed property)

14.14.82 properties as dictionary

Plugin Version: 10.4, Platform: macOS, Targets: Desktop only.

Function: The properties dictionary of the track

Notes: (Read and Write computed property)

14.14.83 SessionFormat as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The session format of the track.

Notes: (Read and Write computed property)

14.14.84 TrackISRC as memoryblock

Plugin Version: 7.5, Platform: macOS, Targets: Desktop only.

Function: The property for the tracks ISRC data.

Notes: For CD-DA audio tracks only. This key points to a memoryblock containing exactly 12 bytes, which will be written to the disc as the International Standard Recording Code (ISRC). If this key is not present, no ISRC is written.

The use of this value should conform to the specifications of the IFPI. See <http://www.ifpi.org/isrc/> for more details on the ISRC standard.

Not all drives are capable of the write modes necessary to write the ISRC. If this key is present in any track and the drive cannot write the ISRC, the burn will fail with `kDRDeviceCantWriteISRCerr`.
(Read and Write computed property)

14.14.85 TrackMode as Integer

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The track mode of the track.

Notes: (Read and Write computed property)

14.14.86 `VerificationType` as string

Plugin Version: 7.4, Platform: macOS, Targets: Desktop only.

Function: The type of verification requested.

Notes: Value should be `DRVerificationTypeReceiveData`, `DRVerificationTypeProduceAgain`, `DRVerificationTypeNone` or `DRVerificationTypeChecksum`.

(Read and Write computed property)

14.14.87 `Events`

14.14.88 `cleanupTrackAfterBurn`

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Cleans up the track after the burn completes.

Notes: Called after burning is complete. Typically you'll clean up what was setup in `prepareTrackForBurn`. Since this method is called after the laser is turned off and the burn is finished, this method can perform time consuming tasks.

True to indicate that the burn should proceed and false to indicate a failure occurred.

14.14.89 `cleanupTrackAfterVerification` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Cleans up the track after the verification completes.

Notes: Once the verification phase is complete, this method is called. The class implementing the method has a chance to do anything up to and including failing the verification.

Return true to indicate success, false to indicate failure.

14.14.90 `estimateLengthOfTrack` as uint64

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Estimates the size of the track to be burned.

Notes: This message is sent outside of a burn cycle in response to a -estimateLength message sent to the track.

Returns the number of blocks of data that the track will occupy. The estimate should be reasonably accurate, and no smaller than the actual size that will be needed.

Only on Mac OS X 10.3 and newer.

14.14.91 prepareTrack(burn as DRBurnMBS) as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Prepares the track for burning.

Notes: Called before any burning starts. Do any sort of setup that needs to be performed (such as opening files). This method can calculate and update the exact track length that will be burned.

Since this method is called before the laser is turned on, this method can perform time consuming tasks.

burn: The burn object controlling the burn

Return true to indicate that the burn should proceed and false to indicate a failure occurred.

14.14.92 prepareTrackForVerification as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Prepare the track to be verified.

Notes: This method is called after the burn completes writing data to disc and before verification phase starts. Now would be a good time to prepare to produce data again by rewinding to the start of files, etc.

Return true to indicate that the verification should proceed and false to indicate a failure occurred.

14.14.93 produceDataForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32) as uint32

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Produces the track data.

Notes: This method is called many times over the course of a burn to obtain data for the track. The buffer passed in will be a multiple of blockSize (bufferLength == blockSize * N, where N >1) and should be filled as full as possible with data. address is the sector address on the disc from the start of the track that is the buffer will be written to.

Since while burning, keeping the drive's buffer full is of utmost importance, you should not perform lengthy operations or block for data in this method. This method should return the number of bytes actually in the buffer or 0 to indicate that there was an error producing the data..

buffer: The buffer to place data into

bufferLength: The length of buffer

address: The on-disc address of where data will be written

blockSize: the size of each block on the disc. It's best to return a multiple of this size.

flags: Some flags. Not yet used.

Return the number of bytes produced.

14.14.94 producePreGapForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32) as uint32

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Produces the pregap data.

Notes: This method is called to obtain data for the track's pregap. If the DRPreGapLengthKey key is present in the track properties, the track producer will be asked for the pregap data first. If the producer implements this selector, then it's the responsibility of the producer to provide data for the pregap, otherwise that length of silence will be produced by DiscRecording.

The buffer passed in will be a multiple of blockSize (bufferLength = blockSize * N, where N >1) and should be filled as full as possible with data. address is the sector address on the disc from the start of the track that is the buffer will be written to.

Since while burning, keeping the drive's buffer full is of utmost importance, you should not perform lengthy operations or block for data in this method. This method should return the number of bytes actually in the buffer or 0 to indicate that there was an error producing the data..

buffer: The buffer to place data into

bufferLength: The length of buffer

address: The on-disc address of where data will be written

blockSize the size of each block on the disc. It's best to return a multiple of this size.

flags: Some flags (not used).

Return the number of bytes produced.

14.14.95 `verifyDataForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32)` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Cleans up the track after the burn completes.

Notes: If the class implementing this method asks for a verification type of `DRVerificationTypeReceiveData`, then a series of calls to this method will start. It's up to the class to reproduce the data again and compare it to the data passed in buffer. The buffer passed in will be a multiple of `blockSize` (`bufferLength == blockSize * N`, where $N > 1$). `address` is the sector address on the disc from the start of the track that is the buffer was written to.

`buffer`: The data read in from the track to compare with

`bufferLength`: The length of buffer

`address`: The on-disc address of where data will was read from.

`blockSize`: the size of each block on the disc. It's best to return a multiple of this size.

`flags`: Some flags. Not used.

Return true to indicate that the data compared successfully and false to indicate a failure occurred.

14.14.96 `verifyPreGapForTrack(buffer as memoryblock, Bufferlen as uint32, address as uint64, blocksize as uint32, byref flags as uint32)` as boolean

Plugin Version: 7.4, Platform: macOS, Targets: .

Function: Checks the integrity track pregap after a burn.

Notes: If the class implementing this method asks for a verification type of `DRVerificationTypeReceiveData`, then a series of calls to this method will start. It's up to the class to reproduce the pregap again and compare it to the data passed in buffer. The buffer passed in will be a multiple of `blockSize` (`bufferLength == blockSize * N`, where $N > 1$). `address` is the sector address on the disc from the start of the track that is the buffer was written to.

`buffer`: The data read in from the track to compare with

`bufferLength`: The length of buffer

`address`: The on-disc address of where data will was read from.

`blockSize`: the size of each block on the disc. It's best to return a multiple of this size.

`flags`: Some flags. (not used)

Return true to indicate that the data compared successfully and false to indicate a failure occurred.

14.14.97 Constants

Constants

Constant	Value	Description
DRFlagSubchannelDataRequested	2	A flag passed to producePreGapForTrack or produceDataForTrack. Indicates that the blockSize passed in includes room for subchannel data. Data producers should check this flag and perform subchannel data production in addition to user data production. Each block requested from the producer in this case will be formatted as [user data (as specified in track properties) [subchannel data(96 bytes)] . For example an audio producer callback should repeatedly produce 2352 bytes of audio data into bytes 0-2351 of the block and an additional 96 bytes subchannel data into bytes 2352-2447.
kDRBlockSizeAudio	2352	One of the block size constants. Audio data.
kDRBlockSizeDVDData	2048	One of the block size constants. DVD data.
kDRBlockSizeMode1Data	2048	One of the block size constants. Mode 1 data.
kDRBlockSizeMode2Data	2332	One of the block size constants. Mode 2 data. Photo CD and CD-i use this.
kDRBlockSizeMode2Form1Data	2048	One of the block size constants. Mode 2 Form 1 data.
kDRBlockSizeMode2Form2Data	2324	One of the block size constants. Mode 2 Form 2 data.
kDRBlockTypeAudio	0	One of the block type constants. Audio data.
kDRBlockTypeDVDData	8	One of the block type constants. DVD data.
kDRBlockTypeMode1Data	8	One of the block type constants. Mode 1 data.
kDRBlockTypeMode2Data	13	One of the block type constants. Mode 2 data. Photo CD and CD-i use this.
kDRBlockTypeMode2Form1Data	10	One of the block type constants. Mode 2 Form 1 data.
kDRBlockTypeMode2Form2Data	12	One of the block type constants. Mode 2 Form 2 data.
kDRDataFormAudio	0	One of the data form constants. Audio data.
kDRDataFormDVDData	16	One of the data form constants. DVD data.
kDRDataFormMode1Data	16	One of the data form constants. Mode 1 data.
kDRDataFormMode2Data	32	One of the data form constants. Mode 2 data. Photo CD and CD-i use this.
kDRDataFormMode2Form1Data	32	One of the data form constants. Mode 2 Form 1 data.
kDRDataFormMode2Form2Data	32	One of the data form constants. Mode 2 Form 2 data.
kDRSessionFormatAudio	0	One of the constants for the session format. Audio data.
kDRSessionFormatCDI	16	One of the constants for the session format. CD-I disc.
kDRSessionFormatCDXA	32	One of the constants for the session format. CD-ROM XA disc.
kDRSessionFormatDVDData	0	One of the constants for the session format. DVD data.
kDRSessionFormatMode1Data	0	One of the constants for the session format. Mode 1 data.
kDRTrackMode1Data	4	One of the track mode constants. Mode 1 data.
kDRTrackMode2Data	4	One of the track mode constants.

Chapter 15

Drag & Drop

15.1 Globals

15.1.1 InstallDragImageMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Install drag image patch.

Notes: This allows you to use retina NSImage for drag image with using Xojo's built in drag and drop classes.

So call InstallDragImageMBS once in app.open to initialize.

Than before call DragItem.Drag, call SetNextDragImageMBS and provide a 2x image as NSImageMBS object.

See example project for details.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo Plugins in version 21.1](#)
- [MBS Xojo Plugins, version 21.1pr5](#)

15.1.2 SetNextDragImageMBS(Img as NSImageMBS, DragItemCount as Integer = 1)

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Sets which image to use for next drag.

Notes: Once image has been replaced for Xojo's drag, the image reference is freed by plugin.

Please call setSize on the NSImageMBS to set the size to use.

e.g. half the pixels, so the picture is sharp on a retina screen.

Added `DragItemCount` parameter for version 23.0 to define how many drag items you have, so all can get the new picture.

Blog Entries

- [MBS Xojo Plugins, version 22.6pr3](#)

Chapter 16

Quartz Composer

16.1 control DesktopQCViewControlMBS

16.1.1 control DesktopQCViewControlMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The control to host a QCView properly.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo / Real Studio Plugins, version 15.1pr2](#)

16.1.2 Properties

16.1.3 View as QCViewMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The QCView in use by this control.

Notes: Please use this property to access the actual view and make your settings and load the data to render.

(Read only property)

16.1.4 Events

16.1.5 BoundsChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

16.1.6 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

16.1.7 DidStartRendering

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Rendering did start notification.

16.1.8 DidStopRendering

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: Rendering did stop notification.

16.1.9 FocusLost

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control lost focus.

In older Xojo versions, this event is named LostFocus.

Notes:

This only fires if the control itself lost focus and not a sub control.

16.1.10 FocusReceived

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The control itself got focus.

In older Xojo versions, this event is named GotFocus.

Notes:

This only fires if the control itself got focus and not a sub control.

16.1.11 FrameChanged

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

16.1.12 MenuBarSelected

Plugin Version: 21.5, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In older Xojo versions, this event is named EnableMenuItems.

16.1.13MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the MouseDown. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

16.1.14 MouseDrag(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

16.1.15 MouseUp(x as Integer, y as Integer)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

16.1.16 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 21.5, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

16.1.17 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

16.2 class QCCompositionMBS

16.2.1 class QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The composition class.

Notes: The QCComposition class represents a Quartz Composer composition that either:

- comes from the system-wide composition repository (/Library/Compositions and textasciitilde /Library/Compositions) where it can be accessed by any application through the methods of the QCCompositionRepository class
- is created from an arbitrary source (typically a file on disk) using one of its methods

This class cannot be subclassed.

A QCComposition object has the following information associated with it and that you can obtain by using the appropriate method of the QCComposition class:

- Attributes include the name and description of the composition, copyright information, and whether or not its provided by OS X (built-in).
- The protocols that the composition conforms to. A composition protocol defines a set of required and optional input parameters and output results.

Many methods of the QCRenderer, QCCompositionLayer, and QCView classes take a QCComposition object as a parameter.

This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 15.1pr2](#)

16.2.2 Methods

16.2.3 compositionWithData(data as MemoryBlock) as QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a composition object initialized with the contents of a Quartz Composer composition file.

Example:

```
dim data as MemoryBlock // composition file content in MemoryBlock variable
dim q as QCCompositionMBS = QCCompositionMBS.compositionWithData(data)
```

Notes: data: The contents of a file created with the Quartz Composer developer tool.

Returns a Quartz Composer composition object or nil if there is an error.

See also:

- 16.2.4 compositionWithData(data as string) as QCCompositionMBS 954

16.2.4 compositionWithData(data as string) as QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a composition object initialized with the contents of a Quartz Composer composition file.

Example:

```
dim data as string // composition file content in string variable
dim q as QCCompositionMBS = QCCompositionMBS.compositionWithData(data)
```

Notes: data: The contents of a file created with the Quartz Composer developer tool.

Returns a Quartz Composer composition object or nil if there is an error.

See also:

- 16.2.3 compositionWithData(data as MemoryBlock) as QCCompositionMBS 953

16.2.5 compositionWithFile(file as folderitem) as QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a composition object initialized with a Quartz Composer composition file.

Example:

```
dim file as folderitem // your folderitem
dim q as QCCompositionMBS = QCCompositionMBS.compositionWithFile(file)
```

Notes: file: A path to a file created with the Quartz Composer developer tool (.qtz extension).

16.2. CLASS QCCOMPOSITIONMBS

955

Returns a Quartz Composer composition object or nil if there is an error.

See also:

- 16.2.6 compositionWithFile(path as string) as QCCompositionMBS

955

16.2.6 compositionWithFile(path as string) as QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns a composition object initialized with a Quartz Composer composition file.

Example:

```
dim path as string = "/System/Library/Compositions/Rollercoaster.qtz" // your path
dim q as QCCompositionMBS = QCCompositionMBS.compositionWithFile(path)
```

Notes: file: A path to a file created with the Quartz Composer developer tool (.qtz extension).

Returns a Quartz Composer composition object or nil if there is an error.

See also:

- 16.2.5 compositionWithFile(file as folderitem) as QCCompositionMBS

954

16.2.7 Constructor

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The private constructor.

16.2.8 copy as QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Creates a copy of the composition.

16.2.9 getAttributes as dictionary

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the attributes of the composition.

16.2.10 `inputKeys as string()`

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array listing the keys that identify the input ports of the root patch of the composition.

16.2.11 `outputKeys as string()`

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array listing the keys that identify the output ports of the root patch of the composition.

16.2.12 `protocols as string()`

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the list of protocols to which the composition conforms.

16.2.13 `QCCCompositionAttributeBuiltInKey as string`

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Example:

```
dim c as QCCCompositionMBS = QCCCompositionMBS.compositionWithFile("/System/Library/Composi-
tions/Rollercoaster.qtz")
dim d as Dictionary = c.getattributes
dim b as Boolean = d.Value(c.QCCCompositionAttributeBuiltInKey)
MsgBox str(b)
```

Notes: The key for the composition origin. The associated value is a Boolean value. True indicates the composition is built-in (provided by OS X).

16.2.14 `QCCCompositionAttributeCategoryKey as string`

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Example:

```
dim c as QCCompositionMBS = QCCompositionMBS.compositionWithFile("/System/Library/Composi-
tions/Rollercoaster.qtz")
dim d as Dictionary = c.getattributes
dim b as string = d.Lookup(c.QCCompositionAttributeCategoryKey, "?")
MsgBox b
```

Notes: The composition category. The associated value is a category constant. See Categories.

16.2.15 QCCompositionAttributeCopyrightKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Example:

```
dim c as QCCompositionMBS = QCCompositionMBS.compositionWithFile("/System/Library/Composi-
tions/Rollercoaster.qtz")
dim d as Dictionary = c.getattributes
dim b as string = d.Value(c.QCCompositionAttributeCopyrightKey)
MsgBox b
```

Notes: The key for composition copyright information.
The associated value is a string.

16.2.16 QCCompositionAttributeDescriptionKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Notes: The key for the composition description.

The associated value is a string.

16.2.17 QCCompositionAttributeHasConsumersKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Notes: The key for a composition that has consumer patches. The associated value is a Boolean value.

True indicates that the composition has consumers.

16.2.18 `QCCompositionAttributeIsTimeDependentKey` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Notes: The key for the composition time dependency. The associated value is a Boolean value. True indicates that the composition is time dependent.

16.2.19 `QCCompositionAttributeNameKey` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard attribute names.

Example:

```
dim c as QCCompositionMBS = QCCompositionMBS.compositionWithFile("/System/Library/Composi-
tions/Rollercoaster.qtz")
dim d as Dictionary = c.getattributes
MsgBox d.Value(c.QCCompositionAttributeNameKey)
```

Notes: The key for the composition name.

The associated value is a string.

16.2.20 `QCCompositionCategoryDistortion` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the category names.

Notes: A composition that produces a distortion effect.

16.2.21 `QCCompositionCategoryStylize` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the category names.

Notes: A composition that produces a stylize effect.

16.2.22 QCCompositionCategoryUtility as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the category names.

Notes: A utility composition.

16.2.23 QCCompositionInputAudioPeakKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A number input port whose key is inputAudioPeak. The value must be in the [0,1] range as a mono signal with no decay applied.

16.2.24 QCCompositionInputAudioSpectrumKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A structure input port whose key is inputAudioSpectrum. The structure must contain 16 values in the [0,1] range representing 16 spectrum bands of the mono signal from low to high frequencies with no decay applied.

16.2.25 QCCompositionInputDestinationImageKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: An image input port whose key is inputDestinationImage.

16.2.26 QCCompositionInputImageKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: An image input port whose key is inputImage.

16.2.27 QCCompositionInputPaceKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A number input port whose key is inputPace. The value must be in the [0,1] range.

16.2.28 QCCompositionInputPreviewModeKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A Boolean input port whose key is inputPreviewMode. When the value of this input port is set to TRUE, the composition that provides this port must be able to run in a low-quality mode that produces a preview of the composition.

16.2.29 QCCompositionInputPrimaryColorKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A color input port whose key is inputPrimaryColor.

16.2.30 QCCompositionInputScreenImageKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: An image input port whose key is inputScreenImage.

16.2.31 QCCompositionInputSecondaryColorKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A color input port whose key is inputSecondaryColor.

16.2.32 QCCompositionInputSourceImageKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: An image input port whose key is `inputSourceImage`.

16.2.33 QCCompositionInputTrackInfoKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A structure input port whose key is `inputTrackInfo`. The structure contains optional entries, such as "name", "artist", "album", "duration", "artwork", and so on.

16.2.34 QCCompositionInputTrackPositionKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A number input port whose key is `inputTrackPosition`. The value must be expressed in seconds.

16.2.35 QCCompositionInputTrackSignalKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A Boolean input port whose key is `inputTrackSignal`.

16.2.36 QCCompositionInputXKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A number input port whose key is `inputX`. The value must be normalized to the image width with the origin on the left.

16.2.37 QCCompositionInputYKey as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard input key names.

Notes: A number input port whose key is `inputY`. The value must be normalized to the image height with

the origin at the bottom.

16.2.38 `QCCompositionOutputImageKey` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard output key names.

Notes: An image output port whose key is `outputImage`.

16.2.39 `QCCompositionOutputWebPageURLKey` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard output key names.

Notes: A string output port whose key is `outputWebPageURL`.

16.2.40 `QCCompositionProtocolGraphicAnimation` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard protocol names.

Notes: A composition that renders a generic graphical animation. It has the option to use `QCCompositionInputPrimaryColorKey` for the primary color of the animation, `QCCompositionInputSecondaryColorKey` for the secondary color of the animation, `QCCompositionInputPaceKey` for the global pace of the animation, and `QCCompositionInputPreviewModeKey` to indicate if the animation should run in lower-quality for preview purposes.

Available in OS X v10.5 and later.

16.2.41 `QCCompositionProtocolGraphicTransition` as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard protocol names.

Notes: A composition that performs a transition between two images, using a transition time in range of 0 to 1. A conforming composition must use the input keys `QCCompositionInputSourceImageKey` for the starting image and `QCCompositionInputDestinationImageKey` for the image to transition to. The composition can optionally use `QCCompositionInputPreviewModeKey` to indicate if the animation should run in lower-quality for preview purposes.

16.2.42 QCCompositionProtocolImageFilter as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard protocol names.

Notes: A composition that applies an effect to a source image. A conforming composition must use the input key `QCCompositionInputImageKey` for the source image and `QCCompositionOutputImageKey` for the output image. The composition can optionally use `QCCompositionInputXKey` to specify the X position of the center point of the effect, `QCCompositionInputYKey` to specify the Y position of the center point of the effect, and `QCCompositionInputPreviewModeKey` to indicate if the animation should run in lower-quality for preview purposes.

16.2.43 QCCompositionProtocolMusicVisualizer as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard protocol names.

Notes: A composition that acts as a visualizer for music. A conforming composition must use the input key `QCCompositionInputAudioPeakKey` for the instantaneous audio peak and the `QCCompositionInputAudioSpectrumKey` for the instantaneous audio spectrum. It can optionally use the `QCCompositionInputTrackInfoKey` to indicate it receives information about the current track and the `QCCompositionInputTrackSignalKey` to indicate the start of a new track.

Available in OS X v10.5 and later.

16.2.44 QCCompositionProtocolScreenSaver as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: One of the standard protocol names.

Notes: A composition that can be used as a screen saver. The composition has the option to use `QCCompositionInputScreenImageKey` for a screenshot image of the screen that the screen saver runs on, `QCCompositionInputPreviewModeKey` to indicate if the animation should run in lower-quality for preview purposes, and `QCCompositionOutputWebPageURLKey` for a URL to open in the default web browser when screen saver exits (only allowed if screen saver password is disabled).

Available in OS X v10.5 and later.

16.2.45 Properties

16.2.46 Description as String

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The description for the composition.

Notes: (Read only property)

16.2.47 Handle as Integer

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal object reference.

Notes: (Read and Write property)

16.2.48 identifier as String

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The identifier for the composition.

Example:

```
dim r as QCCompositionRepositoryMBS = QCCompositionRepositoryMBS.sharedCompositionRepository
dim a() As QCCompositionMBS = r.allCompositions
MsgBox a(0).identifier
```

Notes: The unique identifier for the composition if it comes from the composition repository; "" otherwise.
(Read only property)

16.2.49 Name as String

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The name for the composition.

Example:

```
dim c as QCCompositionMBS = QCCompositionMBS.compositionWithFile("/System/Library/Composi-
tions/Rollercoaster.qtz")
MsgBox c.name
```

Notes: (Read only property)

16.3 class QCCompositionRepositoryMBS

16.3.1 class QCCompositionRepositoryMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The system-wide Quartz Composer composition repository.

Example:

```
dim r as QCCompositionRepositoryMBS = QCCompositionRepositoryMBS.sharedCompositionRepository
dim a() as QCCompositionMBS = r.allCompositions
MsgBox str(a.Ubound+1)+" compositions installed"
```

Notes: The QCCompositionRepository class represents a system-wide centralized repository of built-in and installed Quartz Composer compositions (/Library/Compositions and textasciitilde /Library/Compositions). The QCCompositionRepository class cannot be subclassed.

Compositions in the repository are represented by the QCComposition class. You can use the methods of the QCCompositionRepository class to fetch all compositions or only those that meet specific criteria. This is an abstract class. You can't create an instance, but you can get one from various plugin functions.

Blog Entries

- [MBS Xojo / Real Studio Plugins, version 15.1pr6](#)
- [MBS Xojo / Real Studio Plugins, version 15.1pr2](#)

16.3.2 Methods

16.3.3 allCompositions as QCCompositionMBS()

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array that contains all compositions currently in the composition repository.

Example:

```
dim r as QCCompositionRepositoryMBS = QCCompositionRepositoryMBS.sharedCompositionRepository
dim a() as QCCompositionMBS = r.allCompositions
MsgBox str(a.Ubound+1)+" compositions installed"
```

16.3.4 Compositions(protocols() as String = nil, attributes as Dictionary = nil) as QCCompositionMBS()

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns an array of compositions that match a set of criteria.

Example:

```
dim r as QCCompositionRepositoryMBS = QCCompositionRepositoryMBS.sharedCompositionRepository
dim a() As QCCompositionMBS = r.Compositions( array( QCCompositionMBS.QCCompositionProtocol-
ImageFilter))
MsgBox str(a.Ubound+1)+" compositions found"
```

Notes: protocols: The protocols that you want compositions to conform to. Pass nil if you don't want to filter based on the protocol. You can pass any of these protocols: QCCompositionProtocolAnimation, QCCompositionProtocolImageProducer, QCCompositionProtocolImageFilter, QCCompositionProtocolImageCompositor, QCCompositionProtocolImageTransition, and QCCompositionProtocolScreenSaverRSS.
attributes: A dictionary that contains the attributes, and their associated values, that you want compositions to match. Pass nil if you don't want to filter based on the attributes. For example, you can pass any of these attributes: QCCompositionAttributeNameKey, QCCompositionAttributeDescriptionKey, QCCompositionAttributeCopyrightKey, QCCompositionAttributeBuiltInKey, and QCCompositionAttributeTimeDependentKey.

Returns an array of QCComposition objects that meet the supplied criteria.

16.3.5 compositionWithIdentifier(identifier as string) as QCCompositionMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the composition that corresponds to the identifier.

Example:

```
dim r as QCCompositionRepositoryMBS = QCCompositionRepositoryMBS.sharedCompositionRepository
dim c as QCCompositionMBS = r.compositionWithIdentifier("/swing")
MsgBox c.Name
```

Notes: identifier: A string that uniquely identifies the composition to retrieve.

Returns the composition identified by the provided string, or nil if there is no composition with that identifier in the composition repository.

16.3.6 Constructor

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The private constructor.

16.3.7 loadPlugIn(file as folderitem) as Boolean

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Attempts to load a Quartz Composer plug-in from a .plugin bundle at the specified path and returns true on success.

Example:

```
dim f as FolderItem = SpecialFolder.Desktop.Child("test.plugin")
dim r as Boolean = QCCompositionRepositoryMBS.loadPlugIn(f)

if r then
  MsgBox "OK"
else
  MsgBox "Failed"
end if
```

Notes: This method will do nothing if the plug-in was already loaded.
See also:

- 16.3.8 loadPlugIn(path as string) as Boolean 968

16.3.8 loadPlugIn(path as string) as Boolean

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Attempts to load a Quartz Composer plug-in from a .plugin bundle at the specified path and returns true on success.

Notes: This method will do nothing if the plug-in was already loaded.

See also:

- 16.3.7 loadPlugIn(file as folderitem) as Boolean 968

16.3.9 QCCompositionRepositoryDidUpdateNotification as string

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The new of the notification which is posted whenever the list of compositions in the composition repository is updated.

16.3.10 sharedCompositionRepository as QCCompositionRepositoryMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: Returns the shared instance of the composition repository.

16.3.11 Properties

16.3.12 Handle as Integer

Plugin Version: 15.1, Platform: macOS, Targets: Desktop, Console & Web.

Function: The internal object reference.

Notes: (Read and Write property)

16.4 control QCViewControlMBS

16.4.1 control QCViewControlMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: The control to host a QCView properly.

Blog Entries

- [News from the MBS Xojo Plugins in version 21.5](#)
- [New desktop controls](#)
- [MBS Xojo / Real Studio Plugins, version 15.1pr2](#)

16.4.2 Properties

16.4.3 View as QCViewMBS

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: The QCView in use by this control.

Notes: Please use this property to access the actual view and make your settings and load the data to render.

(Read only property)

16.4.4 Events

16.4.5 BoundsChanged

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the bounds, but not the frame, changed.

16.4.6 Close

Plugin Version: 15.1, Platform: macOS, Targets: .

Function:

The control is about to close.

In Xojo version 2021r3 and newer this event is named Closing.

16.4.7 ConstructContextualMenu(base as MenuItem, x as Integer, y as Integer) as Boolean

Plugin Version: 22.1, Platform: macOS, Targets: .

Function: This event is called when it is appropriate to display a contextual menu for the control.

16.4.8 ContextualMenuAction(hitItem as MenuItem) as Boolean

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: Called when a menuitem is chosen.

Notes: This allows the control to react on its relevant menu items. Please return true if you handled it or false to give others a chance.

16.4.9 didCloseContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Did close contextual menu.

Notes: Allows you to restart any animation you may have stopped in the willShowContextualMenu event.

16.4.10 DidStartRendering

Plugin Version: 15.1, Platform: macOS, Targets: .

Function: Rendering did start notification.

16.4.11 DidStopRendering

Plugin Version: 15.1, Platform: macOS, Targets: .

Function: Rendering did stop notification.

16.4.12 EnableMenuItems

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The event where you can enable menu items.

In Xojo version 2021r3 and newer this event is named `MenuBarSelected`.

16.4.13 `FrameChanged`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The event called when the frame changed.

Notes: This event notifies you, that the control changed it's bounding frame, which is position and/or size.

16.4.14 `GotFocus`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The control itself got focus.

In Xojo version 2021r3 and newer this event is named `FocusReceived`.

Notes:

This only fires if the control itself got focus and not a sub control.

16.4.15 `LostFocus`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function:

The control lost focus.

In Xojo version 2021r3 and newer this event is named `FocusLost`.

Notes:

This only fires if the control itself lost focus and not a sub control.

16.4.16 `MouseDown(x as Integer, y as Integer, Modifiers as Integer) As Boolean`

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was pressed inside the control's region at the location passed in to x, y.

Notes: The coordinates x and y are local to the control, i.e. they represent the position of the mouse click relative to the upper-left corner of the Control.

Return True if you are going to handle the `MouseDown`. In such a case:

- The Action event, if any, will not execute and the state of the object will not change.
- You will receive the MouseDrag and MouseUp events.

If you return False, the system handles the MouseDown so the above event handlers do not get called.

16.4.17 MouseDrag(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: This event fires continuously after the mouse button was pressed inside the Control.

Notes: Mouse location is local to the control passed in to x, y.

As this event is fired continuously (hundreds of time per second), it is your responsibility to determine if the mouse has really moved.

16.4.18 MouseUp(x as Integer, y as Integer)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The mouse button was released.

Notes: Use the x and y parameters to determine if the mouse button was released within the control's boundaries.

16.4.19 Open

Plugin Version: 15.1, Platform: macOS, Targets: .

Function:

The control is about to be created and you can initialize it.

In Xojo version 2021r3 and newer this event is named Opening.

16.4.20 ScaleFactorChanged(NewFactor as Double)

Plugin Version: 17.1, Platform: macOS, Targets: .

Function: The backing store scale factor has changed.

Notes: Please invalidate any cached bitmaps or other relevant state.

16.4.21 willShowContextualMenu(menu as NSMenuMBS, NSEvent as NSEventMBS)

Plugin Version: 24.1, Platform: macOS, Targets: .

Function: Will show contextual menu.

Notes: Your chance to modify the menu before it is shown, e.g. to add menu entries.

16.5 class QCViewMBS

16.5.1 class QCViewMBS

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: The QCView class is a custom NSView class that loads, plays, and controls Quartz Composer compositions.

Notes: It is an autonomous view that is driven by an internal timer running on the main thread.

The view can be set to render a composition automatically when it is placed onscreen. The view stops rendering when it is placed offscreen. When not rendering, the view is filled with the current erase color. The rendered composition automatically synchronizes to the vertical retrace of the monitor.

You can embed this view in a CustomNSViewMBS to get more events for mouse and keyboard. Subclass of the NSViewMBS class.

Blog Entries

- [MBS Releases the MBS Xojo / Real Studio plug-ins in version 15.1](#)
- [MBS Xojo / Real Studio Plugins, version 15.1pr2](#)
- [Plugins 10.1](#)
- [Playing with Quartz Compositions in REALbasic](#)

16.5.2 Methods

16.5.3 Constructor

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new Quartz composer view with size 100/100 and position 0/0

Example:

```
dim t as new QCViewMBS
```

Notes: On success the handle property is not zero.

See also:

- 16.5.4 Constructor(Handle as Integer) 976
- 16.5.5 Constructor(left as Double, top as Double, width as Double, height as Double) 976

16.5.4 Constructor(Handle as Integer)

Plugin Version: 10.3, Platform: macOS, Targets: Desktop only.

Function: Creates an object based on the given QCView handle.

Example:

```
dim t as new QCViewMBS(0, 0, 100, 100)
dim v as new QCViewMBS(t.handle)
```

```
MsgBox str(v.Bounds.Width)+" x "+str(v.Bounds.Height)
```

Notes: The handle is casted to a QCView and the plugin retains this handle.

See also:

- 16.5.3 Constructor 975
- 16.5.5 Constructor(left as Double, top as Double, width as Double, height as Double) 976

16.5.5 Constructor(left as Double, top as Double, width as Double, height as Double)

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Creates a new Quartz composer view with the given size and position.

Example:

```
dim x as new QCViewMBS(0, 0, 100, 100)
```

Notes: On success the handle property is not zero.

See also:

- 16.5.3 Constructor 975
- 16.5.4 Constructor(Handle as Integer) 976

16.5.6 erase

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Clears the view using the current erase color.

16.5.7 getAttributes as dictionary

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns the attributes of the composition associated with the renderer.

Notes: Returns a dictionary that contains the attributes that describe the composition, including the input and output ports of the root patch.

The dictionary can define any of the attributes that are specified by the composition attribute keys. See `QCCompositionAttributeNameKey`, `QCCompositionAttributeDescriptionKey`, and `QCCompositionAttributeCopyrightKey`.

The dictionary can also contain dictionaries that correspond to the keys that identify the input and output ports of the root patch of the composition. See `QCPortAttributeTypeKey`, `QCPortAttributeNameKey`, `QCPortAttributeMinimumValueKey`, `QCPortAttributeMaximumValueKey`, and `QCPortAttributeMenuItemsKey`.

Available in Mac OS X v10.4 and later.

16.5.8 inputKeys as string()

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns an array that contains the keys that identify the input ports of the root patch of the composition.

16.5.9 loadComposition(composition as QCCompositionMBS) as boolean

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Loads a `QCComposition` object into the view.

Notes: Returns true if successful; otherwise false.

If unsuccessful, any composition that's already loaded in the view remains loaded.

16.5.10 loadCompositionFromData(data as MemoryBlock) as boolean

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Loads the composition from memoryblock.

Notes: If unsuccessful, returns false; any composition that's already loaded in the view remains loaded.

Available in Mac OS X v10.4 and later.

See also:

- 16.5.11 `loadCompositionFromData(data as string)` as boolean 978

16.5.11 `loadCompositionFromData(data as string)` as boolean

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Loads the composition from data string.

Notes: If unsuccessful, returns false; any composition that's already loaded in the view remains loaded.

Available in Mac OS X v10.4 and later.

See also:

- 16.5.10 `loadCompositionFromData(data as MemoryBlock)` as boolean 977

16.5.12 `loadCompositionFromFile(file as folderitem)` as boolean

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Loads the composition file located at the specified path.

Notes: file: A folderitem that specifies the location of a Quartz Composer composition file.

If unsuccessful, returns false; any composition that's already loaded in the view remains loaded.

Available in Mac OS X v10.4 and later.

See also:

- 16.5.13 `loadCompositionFromFile(filepath as string)` as boolean 978

16.5.13 `loadCompositionFromFile(filepath as string)` as boolean

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Loads the composition file located at the specified path.

Notes: filepath: A string that specifies the location of a Quartz Composer composition file.

If unsuccessful, returns false; any composition that's already loaded in the view remains loaded.

Available in Mac OS X v10.4 and later.

See also:

- 16.5.12 `loadCompositionFromFile(file as folderitem)` as boolean 978

16.5.14 outputKeys as string()

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns an array that contains the keys that identify the output ports of the root patch of the composition.

16.5.15 pauseRendering

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Pauses rendering in the view.

Notes: You can nest calls to this method.

Available in Mac OS X v10.5 and later.

16.5.16 resumeRendering

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Resumes rendering a paused composition.

Notes: You can nest calls to this method.

Available in Mac OS X v10.5 and later.

16.5.17 setValueForInputKey(value as Variant, key as string) as boolean

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Sets the value for an input port of a composition.

Notes: value: The value to set for the input port. The input port must be at the root patch of the composition. The data type of the value argument must match the input port.

key: The key associated with the input port of the composition. This method throws an exception if key is invalid.

Returns false if it cannot set the value.

Available in Mac OS X v10.4 and later.

16.5.18 startRendering as boolean

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Starts rendering the composition that is in the view.

Notes: Returns false if the composition fails to start rendering; true otherwise.
Available in Mac OS X v10.4 and later.

16.5.19 stopRendering

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Stops rendering the composition that is in the view.

Notes: Available in Mac OS X v10.4 and later.

16.5.20 unloadComposition

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Unloads the composition from the view.

Notes: If necessary, this method calls stopRendering prior to unloading the composition.
Available in Mac OS X v10.5 and later.

16.5.21 valueForKey(key as string) as Variant

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns the value for an input port of a composition.

Notes: key: The key associated with an input port for the root patch of a composition. This method throws an exception if key is invalid.

Returns the value. The data type of returned value depends on the type of the input port.
Available in Mac OS X v10.4 and later.

16.5.22 valueForKey(key as string) as Variant

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns the value for an output port of a composition.

Notes: key

The key associated with an output port for the root patch of a composition. This method throws an exception if key is invalid.

Returns the value as Xojo variant. The data type of returned value depends on the type of the output port.

16.5.23 Properties

16.5.24 `autostartsRendering` as boolean

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Whether the composition that is in the view starts rendering automatically when the view is put on the screen.

Notes: Available in Mac OS X v10.4 and later.

(Read and Write property)

16.5.25 `eraseColor` as `NSColorMBS`

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: The color used to erase the view.

Notes: Available in Mac OS X v10.4 and later.

(Read and Write property)

16.5.26 `eventForwardingMask` as Integer

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: The mask used to filter which types of events are forwarded from the view to the composition during rendering.

Notes: mask: An event filtering mask. The mask can be a combination of any of the mask constants listed in Table below (defined in `NSEventMBS`) or the constant `NSAnyEventMask`.

(Read and Write property)

16.5.27 `isPausedRendering` as boolean

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns whether or not the rendering in the view is paused.

Notes: True if the rendering is paused; otherwise false.

Available in Mac OS X v10.5 and later.

(Read only property)

Constant	Description
<code>NSLeftMouseDownMask</code>	The user pressed the left button.
<code>NSLeftMouseDownMask</code>	The user moved the mouse with the left button down.
<code>NSLeftMouseUpMask</code>	The user released the left button.
<code>NSRightMouseDownMask</code>	The user pressed the right button.
<code>NSRightMouseDownMask</code>	The user moved the mouse with the right button down.
<code>NSRightMouseUpMask</code>	The user released the right button.
<code>NSOtherMouseDownMask</code>	The user pressed the middle button, or some button other than the left or right button.
<code>NSOtherMouseDownMask</code>	The user moved the mouse with the middle button down, or some button other than the left or right button.
<code>NSOtherMouseUpMask</code>	The user released the middle button, or some button other than the left or right button.
<code>NSMouseMovedMask</code>	The user moved the mouse without holding down a mouse button.
<code>NSScrollWheelMask</code>	The user moved the mouse scroll wheel.
<code>NSKeyDownMask</code>	The user generated a character or characters by pressing a key.
<code>NSKeyUpMask</code>	The user released a key.
<code>NSFlagsChangedMask</code>	The user pressed or released a modifier key, or toggled the Caps Lock key.

16.5.28 `isRendering` as `boolean`

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Checks whether a composition is rendering in the view.

Notes: Returns true if a composition is rendering in the view; false otherwise.

Available in Mac OS X v10.5 and later.

(Read only property)

16.5.29 `loadedComposition` as `QCCCompositionMBS`

Plugin Version: 15.1, Platform: macOS, Targets: Desktop only.

Function: Returns the composition loaded in the view.

Notes: (Read only property)

16.5.30 `maxRenderingFrameRate` as `Double`

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: The maximum rendering frame rate.

Notes: Pass 0.0 to specify that there is no limit.

Available in Mac OS X v10.4 and later.

(Read and Write property)

16.5.31 snapshotImage as NSImageMBS

Plugin Version: 10.1, Platform: macOS, Targets: Desktop only.

Function: Returns an NSImage object of the current image in the view.

Notes: Available in Mac OS X v10.5 and later.

(Read only property)

Chapter 17

Statusitem

17.1 class NSStatusBarButtonMBS

17.1.1 class NSStatusBarButtonMBS

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: StatusBarButtons are the visual representation of NSStatusItems, and are primarily displayed on the right side of the menu bar.

Notes: When a template image is set as the `\c` image property of the status bar button, it is rendered with the correct menu bar style. This guarantees that the button will look correct in various button states and appearances (such as dark menu bar).

Available on Mac OS X 10.10 and newer.

Subclass of the NSButtonMBS class.

Blog Entries

- [MonkeyBread Software Releases the MBS Xojo / Real Studio plug-ins in version 15.0](#)
- [MBS Xojo / Real Studio Plugins, version 15.0pr11](#)

17.1.2 Methods

17.1.3 Available as boolean

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: Whether this class is available.

Notes: Returns true on OS X 10.10.

17.1.4 Properties

17.1.5 `appearsDisabled` as Boolean

Plugin Version: 15.0, Platform: macOS, Targets: Desktop only.

Function: Whether icon appears disabled.

Notes: When true the status bar icon has a disabled/off appearance while still being functional, such as allowing selection and actions. Defaults to false.

(Read and Write property)

Chapter 18

Window

18.1 class DesktopWindow

18.1.1 class DesktopWindow

Plugin Version: 21.5, Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Extends Xojo's Window Class.

Example:

```
window1.HasNoTitleBarMBS = true
```

Notes: In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

18.1.2 Methods

18.1.3 NSToolbarMBS as NSToolbarMBS

Plugin Version: 21.5, Platform: macOS, Targets: Desktop only.

Function: The window's toolbar.

Notes: So you can get the toolbar without going over `NSWindowMBS`.
But there you can assign a new toolbar...

18.2 class Window

18.2.1 class Window

Platforms: macOS, Linux, Windows, Targets: Desktop only.

Function: Extends Xojo's Window Class.

Example:

```
window1.HasNoTitleBarMBS = true
```

Notes: In Xojo 2005 and newer you need to use `self.` in front of the method as the propertyname alone is not accepted.

18.2.2 Methods

18.2.3 NSToolbarMBS as NSToolbarMBS

Plugin Version: 11.3, Platform: macOS, Targets: Desktop only.

Function: The window's toolbar.

Notes: So you can get the toolbar without going over `NSWindowMBS`.
But there you can assign a new toolbar...

Chapter 19

List of Questions in the FAQ

- 20.0.1 Can anyone help me convert seconds to time in this format hh:mm:ss? 999
- 20.0.2 Do you have plugins for Android? 1000
- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000
- 20.0.4 How to catch delete key? 1001
- 20.0.5 How to convert cmyk to rgb? 1002
- 20.0.6 How to delete a folder? 1003
- 20.0.7 How to detect if CPU is 64bit processor? 1004
- 20.0.8 How to query variant type string for a variant? 1005
- 20.0.9 How to refresh a htmlviewer on Windows? 1006
- 20.0.10 Is there an example for vector graphics in Xojo? 1007
- 20.0.11 Picture functions do not preserve resolution values? 1008
- 20.0.12 A toolbox call needs a rect - how do I give it one? 1008
- 20.0.13 API client not supported? 1008
- 20.0.14 Can I access Access Database with Java classes? 1009
- 20.0.15 Can I create PDF from Xojo Report using DynaPDF? 1010
- 20.0.16 Can I use AppleScripts in a web application? 1010
- 20.0.17 Can I use graphics class with DynaPDF? 1010
- 20.0.18 Can I use sockets on a web application? 1011
- 20.0.19 Can I use your ChartDirector plugin on a web application? 1011

- 20.0.20 Can I use your DynaPDF plugin on a web application? 1012
- 20.0.21 Can I use your plugin controls on a web application? 1013
- 20.0.22 Can you get an unique machine ID? 1013
- 20.0.23 ChartDirector: Alignment Specification 1013
- 20.0.24 ChartDirector: Color Specification 1014
- 20.0.25 ChartDirector: Font Specification 1017
- 20.0.26 ChartDirector: Mark Up Language 1021
- 20.0.27 ChartDirector: Parameter Substitution and Formatting 1025
- 20.0.28 ChartDirector: Shape Specification 1029
- 20.0.29 Copy styled text? 1030
- 20.0.30 Do you have code to validate a credit card number? 1031
- 20.0.31 Do you have plugins for X-Rite EyeOne, eXact or i1Pro? 1032
- 20.0.32 Does SQL Plugin handle stored procedures with multiple result sets? 1032
- 20.0.33 Does the plugin home home? 1032
- 20.0.34 folderitem.absolutePath is limited to 255 chars. How can I get longer ones? 1033
- 20.0.35 Has anyone played round with using CoreImage to do things like add dissolve transitions say when changing from one tab to another within a window? 1033
- 20.0.36 How about Plugin support for older OS X? 1034
- 20.0.37 How can I detect whether an Intel CPU is a 64bit CPU? 1035
- 20.0.38 How can I disable the close box of a window on Windows? 1036
- 20.0.39 How can I get all the environment variables from Windows? 1036
- 20.0.40 How can i get similar behavior to Roxio Toast or iTunes where clicking a 'burn' button allows the next inserted blank CD-R to bypass the Finder and be accepted by my application? 1037
- 20.0.41 How can I get text from a PDF? 1037
- 20.0.42 How can I get text from a Word Document? 1037
- 20.0.43 How can I get the item string for a given file creator? 1038
- 20.0.44 How can I launch an app using it's creator code? 1039
- 20.0.45 How can I learn what shared libraries are required by a plugin on Linux? 1039
- 20.0.46 How can I validate an email address? 1041
- 20.0.47 How do I decode correctly an email subject? 1041

	991
• 20.0.48 How do I enable/disable a single tab in a tabpanel?	1042
• 20.0.49 How do I find the root volume for a file?	1043
• 20.0.50 How do I get the current languages list?	1043
• 20.0.51 How do I get the Mac OS Version?	1044
• 20.0.52 How do I get the printer name?	1045
• 20.0.53 How do I make a metal window if RB does not allow me this?	1046
• 20.0.54 How do I make a smooth color transition?	1046
• 20.0.55 How do I read the applications in the dock app?	1047
• 20.0.56 How do I truncate a file?	1048
• 20.0.57 How do update a Finder's windows after changing some files?	1048
• 20.0.58 How to access a USB device directly?	1049
• 20.0.59 How to add icon to file on Mac?	1049
• 20.0.60 How to ask the Mac for the Name of the Machine?	1049
• 20.0.61 How to automatically enable retina in my apps?	1050
• 20.0.62 How to avoid leaks with Cocoa functions?	1050
• 20.0.63 How to avoid trouble connecting to oracle database with SQL Plugin?	1051
• 20.0.64 How to avoid ___NSAutoreleaseNoPool console messages in threads?	1051
• 20.0.65 How to bring app to front?	1052
• 20.0.66 How to bring my application to front?	1052
• 20.0.67 How to catch Control-C on Mac or Linux in a console app?	1053
• 20.0.68 How to change name of application menu?	1053
• 20.0.69 How to change the name in the menubar of my app on Mac OS X?	1054
• 20.0.70 How to check if a folder/directory has subfolders?	1054
• 20.0.71 How to check if Macbook runs on battery or AC power?	1055
• 20.0.72 How to check if Microsoft Outlook is installed?	1056
• 20.0.73 How to check on Mac OS which country or language is currently selected?	1056
• 20.0.74 How to code sign my app with plugins?	1057
• 20.0.75 How to collapse a window?	1057
• 20.0.76 How to compare two pictures?	1058

- 20.0.77 How to compile PHP library? 1060
- 20.0.78 How to convert a `BrowserType` to a `String` with `WebSession.Browser`? 1061
- 20.0.79 How to convert a `EngineType` to a `String` with `WebSession.Engine`? 1062
- 20.0.80 How to convert a `PlatformType` to a `String` with `WebSession.Platform`? 1062
- 20.0.81 How to convert a text to iso-8859-1 using the `TextEncoder`? 1063
- 20.0.82 How to convert `ChartTime` back to Xojo date? 1064
- 20.0.83 How to convert line endings in text files? 1064
- 20.0.84 How to convert picture to string and back? 1065
- 20.0.85 How to copy an array? 1066
- 20.0.86 How to copy an dictionary? 1066
- 20.0.87 How to copy parts of a movie to another one? 1066
- 20.0.88 How to create a birthday like calendar event? 1067
- 20.0.89 How to create a GUID? 1068
- 20.0.90 How to create a Mac picture clip file? 1068
- 20.0.91 How to create a PDF file in Xojo? 1069
- 20.0.92 How to create `EmailAttachment` for PDF Data in memory? 1069
- 20.0.93 How to create PDF for image files? 1070
- 20.0.94 How to CURL Options translate to Plugin Calls? 1071
- 20.0.95 How to delete file with ftp and curl plugin? 1072
- 20.0.96 How to detect display resolution changed? 1072
- 20.0.97 How to detect retina? 1073
- 20.0.98 How to disable force quit? 1073
- 20.0.99 How to disable the error dialogs from Internet Explorer on javascript errors? 1073
- 20.0.100 How to display a PDF file in Xojo? 1073
- 20.0.101 How to do a lottery in RB? 1074
- 20.0.102 How to do an asycron DNS lookup? 1075
- 20.0.103 How to draw a dashed pattern line? 1075
- 20.0.104 How to draw a nice antialiased line? 1076
- 20.0.105 How to dump java class interface? 1077

	993
• 20.0.106 How to duplicate a picture with mask or alpha channel?	1078
• 20.0.107 How to enable assistive devices?	1079
• 20.0.108 How to encrypt a file with Blowfish?	1079
• 20.0.109 How to extract text from HTML?	1080
• 20.0.110 How to find empty folders in a folder?	1080
• 20.0.111 How to find iTunes on a Mac OS X machine fast?	1080
• 20.0.112 How to find network interface for a socket by it's name?	1081
• 20.0.113 How to find version of Microsoft Word?	1082
• 20.0.114 How to fix CURL error 60/53 on connecting to server?	1083
• 20.0.115 How to format double with n digits?	1083
• 20.0.116 How to get a time converted to user time zone in a web app?	1084
• 20.0.117 How to get an handle to the frontmost window on Windows?	1084
• 20.0.118 How to get CFAbsoluteTime from date?	1085
• 20.0.119 How to get client IP address on web app?	1085
• 20.0.120 How to get fonts to load in charts on Linux?	1085
• 20.0.121 How to get fonts to load in DynaPDF on Linux?	1086
• 20.0.122 How to get GMT time and back?	1087
• 20.0.123 How to get good crash reports?	1087
• 20.0.124 How to get list of all threads?	1088
• 20.0.125 How to get parameters from webpage URL in Xojo Web Edition?	1088
• 20.0.126 How to get the color for disabled textcolor?	1088
• 20.0.127 How to get the current free stack space?	1089
• 20.0.128 How to get the current timezone?	1090
• 20.0.129 How to get the current window title?	1091
• 20.0.130 How to get the cursor blink interval time?	1092
• 20.0.131 How to get the list of the current selected files in the Finder?	1093
• 20.0.132 How to get the Mac OS system version?	1094
• 20.0.133 How to get the Mac OS Version using System.Gestalt?	1094
• 20.0.134 How to get the screensize excluding the task bar?	1095

- 20.0.135 How to get the size of the frontmost window on Windows? 1095
- 20.0.136 How to get the source code of a HTMLViewer? 1096
- 20.0.137 How to get Xojo apps running Linux? 1096
- 20.0.138 How to handle really huge images with GraphicsMagick or ImageMagick? 1096
- 20.0.139 How to handle tab key for editable cells in listbox? 1097
- 20.0.140 How to hard link MapKit framework? 1098
- 20.0.141 How to have a PDF downloaded to the user in a web application? 1099
- 20.0.142 How to hide all applications except mine? 1099
- 20.0.143 How to hide script errors in HTMLViewer on Windows? 1100
- 20.0.144 How to hide the grid/background/border in ChartDirector? 1100
- 20.0.145 How to hide the mouse cursor on Mac? 1100
- 20.0.146 How to insert image to NSTextView or TextArea? 1100
- 20.0.147 How to jump to an anchor in a htmlviewer? 1101
- 20.0.148 How to keep a movieplayer unclickable? 1101
- 20.0.149 How to keep my web app from using 100% CPU time? 1102
- 20.0.150 How to kill a process by name? 1102
- 20.0.151 How to know how many CPUs are present? 1103
- 20.0.152 How to know the calling function? 1103
- 20.0.153 How to launch an app using it's creator code? 1104
- 20.0.154 How to launch disc utility? 1104
- 20.0.155 How to make a lot of changes to a REAL SQL Database faster? 1105
- 20.0.156 How to make a NSImage object for my retina enabled app? 1105
- 20.0.157 How to make a window borderless on Windows? 1105
- 20.0.158 How to make an alias using AppleEvents? 1106
- 20.0.159 How to make AppleScripts much faster? 1107
- 20.0.160 How to make double clicks on a canvas? 1107
- 20.0.161 How to make my Mac not sleeping? 1109
- 20.0.162 How to make my own registration code scheme? 1110
- 20.0.163 How to make small controls on Mac OS X? 1110

	995
• 20.0.164 How to mark my Mac app as background only?	1111
• 20.0.165 How to move a file or folder to trash?	1111
• 20.0.166 How to move an application to the front using the creator code?	1112
• 20.0.167 How to move file with ftp and curl plugin?	1113
• 20.0.168 How to normalize string on Mac?	1113
• 20.0.169 How to obscure the mouse cursor on Mac?	1114
• 20.0.170 How to open icon file on Mac?	1114
• 20.0.171 How to open PDF in acrobat reader?	1114
• 20.0.172 How to open printer preferences on Mac?	1115
• 20.0.173 How to open special characters panel on Mac?	1116
• 20.0.174 How to optimize picture loading in Web Edition?	1116
• 20.0.175 How to parse XML?	1116
• 20.0.176 How to play audio in a web app?	1117
• 20.0.177 How to pretty print xml?	1118
• 20.0.178 How to print to PDF?	1118
• 20.0.179 How to query Spotlight's Last Open Date for a file?	1119
• 20.0.180 How to quit windows?	1120
• 20.0.181 How to read a CSV file correctly?	1120
• 20.0.182 How to read the command line on windows?	1121
• 20.0.183 How to render PDF pages with PDF Kit?	1121
• 20.0.184 How to restart a Mac?	1122
• 20.0.185 How to resume ftp upload with curl plugin?	1122
• 20.0.186 How to rotate a PDF page with CoreGraphics?	1123
• 20.0.187 How to rotate image with CoreImage?	1124
• 20.0.188 How to run a 32 bit application on a 64 bit Linux?	1125
• 20.0.189 How to save HTMLViewer to PDF with landscape orientation?	1125
• 20.0.190 How to save RTFD?	1125
• 20.0.191 How to save RTFD?	1126
• 20.0.192 How to scale a picture proportionally with mask?	1126

- 20.0.193 How to scale a picture proportionally? 1127
- 20.0.194 How to scale/resize a CIImageMBS? 1128
- 20.0.195 How to scale/resize a picture? 1129
- 20.0.196 How to search with regex and use unicode codepoints? 1129
- 20.0.197 How to see if a file is invisible for Mac OS X? 1130
- 20.0.198 How to set cache size for SQLite or REALSQLDatabase? 1131
- 20.0.199 How to set the modified dot in the window? 1131
- 20.0.200 How to show a PDF file to the user in a Web Application? 1131
- 20.0.201 How to show Keyboard Viewer programmatically? 1132
- 20.0.202 How to show the mouse cursor on Mac? 1133
- 20.0.203 How to shutdown a Mac? 1133
- 20.0.204 How to sleep a Mac? 1134
- 20.0.205 How to speed up rasterizer for displaying PDFs with DynaPDF? 1134
- 20.0.206 How to use PDFLib in my RB application? 1134
- 20.0.207 How to use quotes in a string? 1135
- 20.0.208 How to use Sybase in Web App? 1135
- 20.0.209 How to use the Application Support folder? 1135
- 20.0.210 How to use the IOPMCopyScheduledPowerEvents function in Xojo? 1136
- 20.0.211 How to validate a GUID? 1139
- 20.0.212 How to walk a folder hierarchie non recursively? 1139
- 20.0.213 I got this error: PropVal, QDPictMBS.Name (property value), Type mismatch error. Expected CGDataProviderMBS, but got Variant, Name:QDPictMBS 1140
- 20.0.214 I registered the MBS Plugins in my application, but later the registration dialog is shown. 1140
- 20.0.215 I want to accept Drag & Drop from iTunes 1141
- 20.0.216 I'm drawing into a listbox but don't see something. 1143
- 20.0.217 I'm searching for a method or so to move a window from position x.y to somewhere else on the screen. 1143
- 20.0.218 If I use one of your plug-ins under windows, would this then impose the use of dll after compilation or my would my compiled soft still be a stand-alone single file software? 1143
- 20.0.219 Is the fn key on a powerbook keyboard down? 1144

	997
• 20.0.220 Is there a case sensitive Dictionary?	1144
• 20.0.221 Is there a way to use the MBS plugin to get only the visible item and folder count on a volume?	1145
• 20.0.222 Is there an easy way I can launch the Displays preferences panel?	1145
• 20.0.223 List of Windows Error codes?	1146
• 20.0.224 Midi latency on Windows problem?	1146
• 20.0.225 My Xojo Web App does not launch. Why?	1146
• 20.0.226 SQLiteDatabase not initialized error?	1147
• 20.0.227 Textconverter returns only the first x characters. Why?	1147
• 20.0.228 The type translation between CoreFoundation/Foundation and Xojo data types.	1148
• 20.0.229 Uploaded my web app with FTP, but it does not run on the server!	1150
• 20.0.230 What classes to use for hotkeys?	1150
• 20.0.231 What do I need for Linux to get picture functions working?	1150
• 20.0.232 What does the NAN code mean?	1151
• 20.0.233 What font is used as a 'small font' in typical Mac OS X apps?	1151
• 20.0.234 What is last plugin version to run on Mac OS X 10.4?	1152
• 20.0.235 What is last plugin version to run on PPC?	1152
• 20.0.236 What is last version of the plugins for macOS 32-bit?	1153
• 20.0.237 What is the difference between Timer and WebTimer?	1153
• 20.0.238 What is the list of Excel functions?	1153
• 20.0.239 What is the replacement for PluginMBS?	1154
• 20.0.240 What to do on Xojo reporting a conflict?	1154
• 20.0.241 What to do with a NSImageCacheException?	1155
• 20.0.242 What to do with MySQL Error 2014?	1155
• 20.0.243 What to do with SQL Plugin reporting Malformed string as error?	1155
• 20.0.244 Where is CGGetActiveDisplayListMBS?	1155
• 20.0.245 Where is CGGetDisplaysWithPointMBS?	1156
• 20.0.246 Where is CGGetDisplaysWithRectMBS?	1156
• 20.0.247 Where is CGGetOnlineDisplayListMBS?	1156
• 20.0.248 Where is GetObjectClassNameMBS?	1156

- 20.0.249 Where is NetworkAvailableMBS? 1156
- 20.0.250 Where is StringHeight function in DynaPDF? 1157
- 20.0.251 Where is XLSDocumentMBS class? 1157
- 20.0.252 Where to get information about file formats? 1157
- 20.0.253 Where to register creator code for my application? 1158
- 20.0.254 Which Mac OS X frameworks are 64bit only? 1158
- 20.0.255 Which plugins are 64bit only? 1159
- 20.0.256 Why application doesn't launch because of a missing ddraw.dll!? 1159
- 20.0.257 Why application doesn't launch because of a missing shlwapi.dll!? 1159
- 20.0.258 Why do I hear a beep on keydown? 1159
- 20.0.259 Why does folderitem.item return nil? 1159
- 20.0.260 Why doesn't showurl work? 1159
- 20.0.261 Why don't the picture functions not work on Linux? 1160
- 20.0.262 Why have I no values in my chart? 1160
- 20.0.263 Will application size increase with using plugins? 1160
- 20.0.264 XLS: Custom format string guidelines 1160
- 20.0.265 Xojo doesn't work with your plugins on Windows 98. 1161
- 20.0.266 Xojo or my RB application itself crashes on launch on Mac OS Classic. Why? 1162

Chapter 20

The FAQ

20.0.1 Can anyone help me convert seconds to time in this format hh:mm:ss?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sure, here's a routine I use (which has an advantage over the previously-posted Date-based solution in that you don't have to rely on the creation of an object – all that happens is some division and string concatenation):

Example:

```
Function SecsToTimeString(timeInSecs as Integer, padHours as boolean, padMinutes as boolean) as string
// Given an amount time (in seconds), generates a string representing that amount
// of time. The padHours and padMinutes parameters determine whether to display
// hours and minutes if their values are zero.
```

```
// Examples:
// timeInSecs = 90, padHours = true; returns "00:01:30"
// timeInSecs = 1, padHours = false, padMinutes = true; returns "00:01"
// timeInSecs = 3601, padMinutes = false; returns "01:00:01"
```

```
dim hours, minutes, seconds as Integer
dim hoursString, minutesString as string
```

```
hours = timeInSecs / 3600
minutes = (timeInSecs mod 3600) / 60
seconds = timeInSecs mod 60
```

```
if hours = 0 then
if padHours then
hoursString = "00:"
else
hoursString = ""
end if
```

```

else
hoursString = Format(hours, "##\:")
end if
if minutes = 0 then
if hours <>0 or padMinutes then
minutesString = "00:"
else
minutesString = ""
end if
else
minutesString = Format(minutes, "00\:")
end if

return hoursString + minutesString + Format(seconds, "00")
End Function

```

Notes: (from the rb mailinglist)

20.0.2 Do you have plugins for Android?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Since there is no plugin SDK for Android, we have no way to make a plugin for Android.

Notes: We support macOS, Windows, Linux and iOS.

20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use functions from NSColor to get proper highlight color in RGB:

Example:

```

Function ProperHighlightColor(active as Boolean) As Color
#if TargetCocoa
Dim theColor As NSColorMBS
If active Then
theColor = NSColorMBS.alternateSelectedControlColor
Else
theColor = NSColorMBS.secondarySelectedControlColor
End If

```

```

Dim rgbColor As NSColorMBS = theColor.colorUsingColorSpaceName(NSColorSpaceMBS.NSCalibrate-

```

```

dRGBColorSpace)
If rgbColor <>Nil Then
Dim red as Integer = rgbColor.redComponent * 255.0
Dim green as Integer = rgbColor.greenComponent * 255.0
Dim blue as Integer = rgbColor.blueComponent * 255.0
Return RGB(red, green, blue)
Else
Return HighlightColor
End If
#else
return HighlightColor
#endif
End Function

```

Notes: As you see we convert color to Calibrated RGB for best results.
See also:

- 20.0.4 How to catch delete key? 1001
- 20.0.5 How to convert cmyk to rgb? 1002
- 20.0.6 How to delete a folder? 1003
- 20.0.7 How to detect if CPU is 64bit processor? 1004
- 20.0.8 How to query variant type string for a variant? 1005
- 20.0.9 How to refresh a htmlviewer on Windows? 1006

20.0.4 How to catch delete key?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following is the code in keydown event catches delete or backspace keys.

Example:

```

Function KeyDown(Key As String) As Boolean
if asc(key) = 8 or asc(key) = 127 then
MsgBox "Delete"
Return true
end if
End Function

```

See also:

- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000

- 20.0.5 How to convert cmyk to rgb? 1002
- 20.0.6 How to delete a folder? 1003
- 20.0.7 How to detect if CPU is 64bit processor? 1004
- 20.0.8 How to query variant type string for a variant? 1005
- 20.0.9 How to refresh a htmlviewer on Windows? 1006

20.0.5 How to convert cmyk to rgb?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

The following is the code to convert cmyk values to an RGB color datatype.

It's just a basic estimate of the color values. If you are looking for completely color accurate solution, this is not it. It should work for most people. :)

Example:

Function CMYKToRGB(c as Integer, m as Integer, y as Integer, k as Integer) As color

// converts c,m,y,k values (0-100) to color data type RGB

// place this in a method. Supply C,M,Y,K values-

// it returns color datatype

```
dim color_RGB as color
```

```
dim r, g, b as Integer
```

```
r=255-round(2.55*(c+k))
```

```
if r<0 then
```

```
r=0
```

```
end if
```

```
g=255-round(2.55*(m+k))
```

```
if g<0 then
```

```
g=0
```

```
end if
```

```
b=255-round(2.55*(y+k))
```

```
if b<0 then
```

```
b=0
```

```
end if
```

```
color_RGB=RGB(r,g,b)
```

```
return color_RGB
```

```
End Function
```

Notes:

(from the rb mailinglist)
See also:

- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000
- 20.0.4 How to catch delete key? 1001
- 20.0.6 How to delete a folder? 1003
- 20.0.7 How to detect if CPU is 64bit processor? 1004
- 20.0.8 How to query variant type string for a variant? 1005
- 20.0.9 How to refresh a htmlviewer on Windows? 1006

20.0.6 How to delete a folder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The following is the code that deletes a folder recursively.

Example:

```
Sub deletefolder(f as folderitem)
dim files(-1) as FolderItem

if f=nil then Return

// delete single file
if f.Directory=false then
f.Delete
Return
end if

// get a list of all items in that folder
dim i,c as Integer
c=F.Count
for i=1 to c
files.Append f.TrueItem(i)
next

// delete each item
for each fo as FolderItem in files
if fo=nil then
' ignore
elseif fo.Directory then
deletefolder fo
fo.delete
else ' file
```

```
fo.Delete
end if
next
```

```
f.Delete
End Sub
```

See also:

- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000
- 20.0.4 How to catch delete key? 1001
- 20.0.5 How to convert cmyk to rgb? 1002
- 20.0.7 How to detect if CPU is 64bit processor? 1004
- 20.0.8 How to query variant type string for a variant? 1005
- 20.0.9 How to refresh a htmlviewer on Windows? 1006

20.0.7 How to detect if CPU is 64bit processor?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Via CPUID you can ask CPU:

Example:

```
dim c as new CPUIDMBS

if c.Flags(CPUIDMBS.kFeatureLM) then
MsgBox "64-bit CPU"
else
MsgBox "32-bit CPU"
end if
```

Notes: Should work on all intel compatible CPUs.

See also:

- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000
- 20.0.4 How to catch delete key? 1001
- 20.0.5 How to convert cmyk to rgb? 1002
- 20.0.6 How to delete a folder? 1003
- 20.0.8 How to query variant type string for a variant? 1005
- 20.0.9 How to refresh a htmlviewer on Windows? 1006

20.0.8 How to query variant type string for a variant?

Plugin Version: 20.5, Platforms: macOS, Linux, Windows.

Answer: The following example function returns type string for variant.

Example:

```
Public Function VariantTypeString(v as Variant) as string
// Xojo's VarType doesn't know Unsigned integers
'Dim type As Integer = VarType(v)

// MBS VarType can detect unsigned integer
Dim type As Integer = GetVariantTypeMBS(v)

Dim IsArray As Boolean = BitwiseAnd(type, Variant.TypeArray) = Variant.TypeArray

// type without array
type = BitwiseAnd(type, Bitwise.OnesComplement(Variant.TypeArray))

// build a dictionary to map types on first call
Static TypeMap As Dictionary
If TypeMap = Nil Then
TypeMap = New Dictionary
TypeMap.Value(Variant.TypeBoolean) = "Boolean"
TypeMap.Value(Variant.TypeCFStringRef) = "CFStringRef"
TypeMap.Value(Variant.TypeColor) = "Color"
TypeMap.Value(Variant.TypeCString) = "CString"
TypeMap.Value(Variant.TypeCurrency) = "Currency"
TypeMap.Value(Variant.TypeDate) = "Date"
TypeMap.Value(Variant.TypeDateTime) = "DateTime"
TypeMap.Value(Variant.TypeDouble) = "Double"
TypeMap.Value(Variant.TypeInt32) = "Int32"
TypeMap.Value(Variant.TypeInt64) = "Int64"
TypeMap.Value(Variant.TypeInteger) = "Integer"
TypeMap.Value(Variant.TypeNil) = "Nil"
TypeMap.Value(Variant.TypeObject) = "Object"
TypeMap.Value(Variant.TypeOSType) = "OSType"
TypeMap.Value(Variant.TypePString) = "PString"
TypeMap.Value(Variant.TypePtr) = "Ptr"
TypeMap.Value(Variant.TypeSingle) = "Single"
TypeMap.Value(Variant.TypeString) = "String"
TypeMap.Value(Variant.TypeStructure) = "Structure"
TypeMap.Value(Variant.TypeText) = "Text"
TypeMap.Value(Variant.TypeWindowPtr) = "WindowPtr"
TypeMap.Value(Variant.TypeWString) = "WString"

// MBS extra types
TypeMap.Value(Variant.TypeInt32+100) = "UInt32"
TypeMap.Value(Variant.TypeInt64+100) = "UInt64"
```

End If

```
// lookup type

#if DebugBuild then
If Not TypeMap.HasKey(type) Then
Break // missing type
End If
#endif

If IsArray Then
Return "Array of " + TypeMap.Lookup(type,"?")
Else
Return TypeMap.Lookup(type,"?")
End If
End Function
```

See also:

- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000
- 20.0.4 How to catch delete key? 1001
- 20.0.5 How to convert cmyk to rgb? 1002
- 20.0.6 How to delete a folder? 1003
- 20.0.7 How to detect if CPU is 64bit processor? 1004
- 20.0.9 How to refresh a htmlviewer on Windows? 1006

20.0.9 How to refresh a htmlviewer on Windows?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can ask the browser to reload the website with this code line:

Example:

```
call htmlViewer1.IERunJavaScriptMBS("javascript:document.location.reload()")
```

See also:

- 20.0.3 How do I get the proper highlight color on Mac OS X for active/inactive selection? 1000
- 20.0.4 How to catch delete key? 1001
- 20.0.5 How to convert cmyk to rgb? 1002

- | | |
|--|------|
| | 1007 |
| • 20.0.6 How to delete a folder? | 1003 |
| • 20.0.7 How to detect if CPU is 64bit processor? | 1004 |
| • 20.0.8 How to query variant type string for a variant? | 1005 |

20.0.10 Is there an example for vector graphics in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this example inside the paint event of a window:

Example:

```
dim v as Group2D
dim r as RectShape
dim s as StringShape
```

```
const pi=3.14
```

```
s=new StringShape
s.Text="Hello World!"
s.TextFont="Geneva"
s.TextSize=24
s.FillColor=rgb(0,0,255)
s.Italic=true
s.y=5
s.x=0
```

```
r=new RectShape
```

```
r.X=0
r.y=0
r.Height=100
r.Width=180
r.BorderColor=rgb(255,0,0)
r.FillColor=rgb(0,255,0)
r.BorderWidth=5
r.Border=50
```

```
v=new Group2d
v.Append r
v.Append s
v.Rotation=pi*-20.0/180.0
v.x=150
v.y=150
```

```
g.DrawObject v
```

20.0.11 Picture functions do not preserve resolution values?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the picture functions return pictures with no/default resolution values.

Example:

```
dim l as Picture = LogoMBS(500)
```

```
l.HorizontalResolution = 300
```

```
l.VerticalResolution = 300
```

```
dim r as Picture = l.Rotate90MBS
```

```
MsgBox str(r.HorizontalResolution)+" x "+str(r.VerticalResolution)
```

```
r.HorizontalResolution = l.HorizontalResolution
```

```
r.VerticalResolution = l.VerticalResolution
```

```
MsgBox str(r.HorizontalResolution)+" x "+str(r.VerticalResolution)
```

Notes: So please fix them yourself after calling a function.

Maybe in the future this changes, but currently you can't really set this easily from plugin code.

20.0.12 A toolbox call needs a rect - how do I give it one?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Fill a memoryblock like this:

Example:

```
Dim MB As Memoryblock
```

```
MB = NewMemoryBlock(8)
```

```
MB.Short(0) = window1.Top
```

```
MB.Short(2) = window1.Left
```

```
MB.Short(4) = window1.Height+window1.Top // bottom
```

```
MB.Short(6) = window1.Width+window1.Left // right
```

20.0.13 API client not supported?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you get this exception message on SQLConnectionMBS.Connect, we may have a problem.

Notes: First case is that the given thing is not supported (e.g. MS SQL directly on Mac).

Second case is that the plugin compilation went wrong and the support for the database was not linked into the plugin. Like MySQL missing or MS SQL on Windows missing. In that case please contact us to fix the plugin.

20.0.14 Can I access Access Database with Java classes?

Plugin Version: all, Platform: Windows.

Answer: You can use ucanaccess to access databases created with Microsoft

Example:

```

dim options(-1) as string

// load all the jar files we have in a folder called java:

dim appFolder as FolderItem = GetFolderItem("")

Dim count as Integer = appFolder.Parent.Child("java").Count
dim libjs() as string
For i as Integer = 1 to count
Dim f As FolderItem = appFolder.Parent.Child("java").item(i)
If f <> Nil and f.Exists Then
libjs.append f.NativePath+";"
End If
Next

// now init virtual machine
dim library as string = Join(libjs, "")
dim vm as new JavaVMMBS(library)

if vm.Handle = 0 then
MsgBox "Failed to initialize virtual machine"
else
// now make a new database connection with ucanaccess
dim d as new JavaDatabaseMBS(vm,"net.ucanaccess.jdbc.UcanaccessDriver")
Dim DbFile as FolderItem = appFolder.Parent.Child("Database11.accdb")
dim j as JavaConnectionMBS = d.getConnection("jdbc:ucanaccess://" + DbFile.NativePath)

// select and show values
dim r as JavaResultSetMBS = j.MySelectSQL("Select * From test")
while r.NextRecord
MsgBox r.getString("FirstName") + " " + r.getString("LastName")
wend

end if

```

Exception e as JavaExceptionMBS
MsgBox e.message+" **errorcode:** "+str(e.ErrorNumber)

Notes: see website:
<http://ucanaccess.sourceforge.net/site.html>

20.0.15 Can I create PDF from Xojo Report using DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, we have a graphics class integration for DynaPDF.

Notes: Since MBS Plugin in version 19.2, we can integrate reports with Xojo.

20.0.16 Can I use AppleScripts in a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, but they run on the server, not on the client.

Example:

```
dim a as new AppleScriptMBS

// query my application name
a.Compile "tell application ""System Events"" to return name of current application"

// run
a.Execute

// show result
label1.text = a.Result

// shows something like "My Application.fcgi.debug"
```

Notes: This can be useful to control the server from remote, if and only if the your sever is running Mac OS X.

20.0.17 Can I use graphics class with DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sorry, no. We can't provide a graphics subclass from plugin.

Notes: This is a feature request to allow graphics subclasses:

Feedback case 11391: [feedback://showreport?report_id=11391](https://feedback.apple.com/showreport?report_id=11391)

20.0.18 Can I use sockets on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, but they run on the server, not on the client.

Notes: You can use `HTTPSocket`, `SMTPSocket`, `POP3Socket`, `SMTPSecureSocket`, `SecurePOP3Socket`, `EasyTCPSocket`, `EasyUDPSocket`, `AutoDiscovery`, our Bonjour classes or our `CURL*` classes. But all of them work on the server, not on the client.

This means if you search for a printer with Bonjour, you can find the printers in the local network on your server hosting site. Using `SMTPSocket` may be a good idea for sending emails from the server like notifications.

20.0.19 Can I use your ChartDirector plugin on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, our ChartDirector plugin works just fine on the Xojo Web Edition.

Example:

```
// The data for the pie chart
dim data(-1) as Double=array(55.0, 18.0, 25.0, 22.0, 18.0, 30.0, 35.0)

// The labels for the pie chart, Words are chosen random to check font!
dim labels(-1) as string=array("Germany", "Italy", "France", "Spain", "UK", "Poland", "Russia")

// The colors to use for the sectors
dim colors(-1) as Integer

colors.Append &h66aaee
colors.Append &heebb22
colors.Append &hbbsbbb
colors.Append &h8844ff

if TargetLinux then
CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype/msttcorefonts"
end if

// Create a PieChart object of size 360 x 300 pixels
dim c as new CDPieChartMBS(700, 600)
```

```

c.setBackground(c.linearGradientColor(0, 0, 0, c.getHeight(), &h0000cc, &h000044))
c.setRoundedFrame(&hffffff, 16)
dim tt as CDTextBoxMBS = c.addTitle("ChartDirector Demonstration", "timesbi.ttf", 18)
tt.setMargin(0, 0, 16, 0)
tt.setFontColor(&hFFFFFF)

// Set the center of the pie at (180, 140) and the radius to 100 pixels
c.setPieSize 350,300,150
// Set the sector colors
c.setColors(c.kDataColor, colors)

// Draw the pie in 3D with a pie thickness of 20 pixels
c.set3D(20)

dim t as CDTextBoxMBS = c.setLabelStyle("arialbd.ttf", 10, &h000000)
t.setBackground(CDPieChartMBS.kSameAsMainColor, CDPieChartMBS.kTransparent, CDPieChartMBS.soft-
Lighting(CDPieChartMBS.kRight, 0))
t.setRoundedCorners(8)

// Use local gradient shading for the sectors, with 5 pixels wide
// semi-transparent white (bbffffff) borders
c.setSectorStyle(CDPieChartMBS.kLocalGradientShading, &hbbffffff, 0)

// Set the pie data and the pie labels
c.setData data,labels
call c.setLabelStyle "arialbd.ttf",18

dim pic as picture = c.makeChartPicture
dim wp as new WebPicture(pic, Picture.FormatJPEG) // JPEG makes it smaller and faster

ImageView1.Picture=wp

```

Notes: Be aware that our plugin produces pictures for you, which you assign to ImageViews. Transferring those pictures takes time, so you can optimize that with using WebPicture class. There you can decide between different compressions to improve speed (use JPEG instead of PNG).

e.g. if you use ubuntu, you can install the ttf-mscorefonts-installer package and call this method with "/usr/share/fonts/truetype/msttcorefonts" as the path. No backslash on the end of a path, please.

20.0.20 Can I use your DynaPDF plugin on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, our DynaPDF plugin works just fine on the Xojo Web Edition.

Notes: PDF files are created on the server. You may want to offer a preview to the user which uses reduced resolution images to reduce the time to download the PDF.

See our Create PDF example for the Xojo Web Edition.

20.0.21 Can I use your plugin controls on a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: No.

20.0.22 Can you get an unique machine ID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: There is nothing like an unique machine ID.

Notes: 1:

You can use the MAC IDs of the network interfaces.

This can be changed by the user with software tools.

And the list of network interfaces changes if user reorder the interfaces.

2:

You can use the system folder creation date/time.

This may stay equal after cloning machines or after migration to new PC.

3:

You can use the Mac Serialnumber.

Mac only and it can happen that a Mac does not have a serial number.

4:

You can use the x86 CPU ID.

This is x86 CPU only and does not avoid running on the same CPU in different PCs.

20.0.23 ChartDirector: Alignment Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Alignment Specification

Notes: In many ChartDirector objects, you may specify the alignment of the object's content relative to its boundary. For example, for a TextBox object, you may specify the text's alignment relative to the box boundary by using `TextBox.setAlignment`.

The ChartDirector API defines several constants for the alignment options.

ConstantValueDescription

BottomLeft	1	The leftmost point on the bottom line.
BottomCenter	2	The center point on the bottom line.
BottomRight	3	The rightmost point on the bottom line.
Left	4	The leftmost point on the middle horizontal line.
Center	5	The center point on the middle horizontal line.
Right	6	The rightmost point on the middle horizontal line.
TopLeft	7	The leftmost point on the top line.
TopCenter	8	The center point on the top line.
TopRight	9	The rightmost point on the top line.
Bottom	2	The center point on the bottom line. Same as BottomCenter.
Top	8	The center point on the top line. Same as TopCenter.
TopLeft2	10	An alternative top-left position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, TopLeft2 refers to refers to the left of the top side, while TopLeft refers to the top of the left side. The reverse applies for a horizontal axis.
TopRight2	11	An alternative top-right position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, TopRight2 refers to refers to the right of the top side, while TopRight refers to the top of the right side. The reverse applies for a horizontal axis.
BottomLeft2	12	An alternative bottom-left position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, BottomLeft2 refers to refers to the left of the bottom side, while BottomLeft refers to the bottom of the left side. The reverse applies for a horizontal axis.
BottomRight2	13	An alternative bottom-right position used in <code>Axis.setTitlePos</code> for axis title positioning only. For a vertical axis, BottomRight2 refers to refers to the right of the bottom side, while BottomRight refers to the bottom of the right side. The reverse applies for a horizontal axis.

20.0.24 ChartDirector: Color Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Color Specification

Notes: Many functions in the ChartDirector API accept colors as parameters. ChartDirector supports col-

ors specified in web and HTML compatible ARGB format, in which ARGB refers to the Alpha transparency, Red, Green and Blue components of the color.

In addition to ARGB colors, ChartDirector supports "dynamic" colors. A dynamic color is a color that changes depending on the position of the pixels. The "dynamic" colors that ChartDirector supports include "pattern colors", "metal colors", "gradient colors", "zone colors" and "dash line colors".

ChartDirector supports specifying colors indirectly using "palette colors". When a "palette color" is used, the color is specified as an index to a palette. The actual color is looked up from the palette. ARGB Color ARGB color consists of 4 components - alpha transparency, red, green and blue. The four components are encoded as a 32-bit number, with each component occupying 8 bits. In hexadecimal notation, it is AAR-RGGBB, where AA, RR, GG and BB are the alpha transparency, red, green and blue components.

Each component ranges from 00 - FF (0 - 255), representing its intensity. For example, pure red color is 00FF0000, pure green color is 0000FF00, and pure blue color is 000000FF. White color is 00FFFFFF, and black color is 00000000.

Most programming language requires you to put special prefix in front of hexadecimal characters. For C++, the prefix is "0x". For example, the syntax for the hexadecimal number 00FFFFFF is 0x00FFFFFF, or simply 0xFFFFFF.

For the alpha transparency component, a zero value means the color is not transparent at all. This is equivalent to traditional RGB colors. A non-zero alpha transparency means the color is partially transparent. The larger the alpha transparency, the more transparent the color will be. If a partially transparent color is used to draw something, the underlying background can still be seen.

For example, 80FF0000 is a partially transparent red color, while 00FF0000 is a non-transparent red color.

Note that ChartDirector's ARGB color is web and HTML compatible. For example, red is FF0000, the same as in HTML. There are many resources on the web that provide tables in which you can click a color and it will show its HTML color code. These color codes can be used in ChartDirector.

If alpha transparency is FF (255), the color is totally transparent. That means the color is invisible. It does not matter what the RGB components are. So in ChartDirector, only one totally transparent color is used - FF000000. All other colors of the form FFnnnnnn are reserved to represent palette colors and dynamic colors, and should not be interpreted as the normal ARGB colors.

The totally transparent color FF000000 is often used in ChartDirector to disable drawing something. For example, if you want to disable drawing the border of a rectangle, you can set the border color to totally transparent.

For convenience, ChartDirector defines a constant called Transparent, which is equivalent to FF000000. Pattern Color

A pattern color is a dynamic color that changes according to a 2D periodic pattern. When it is used to fill an area, the area will look like being tiled with a wallpaper pattern.

Pattern colors are created using `BaseChart.patternColor`, `BaseChart.patternColor2`, `DrawArea.patternColor` and `DrawArea.patternColor2`. The `patternColor` method creates pattern colors using an array of colors as a bitmap. The `patternColor2` method creates pattern colors by loading the patterns from image files.

These methods return a 32-bit integer acting as a handle to the pattern color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Metal Color
A metal color is a color of which the brightness varies smoothly across the chart surface as to make the surface look shiny and metallic. `ChartDirector` supports using any color as the base color of the metal color. In particular, using yellow and grey as the base colors will result in metal colors that look gold and silver.

Metal colors are most often used as background colors of charts. They are created using `CDBaseChartMBS.metalColor`, `CDBaseChartMBS.goldColor` and `CDBaseChartMBS.silverColor`. The first method allows you to specify an arbitrary base color. The second and third methods use yellow and grey as the base colors, resulting in gold and silver metal colors.

These methods return a 32-bit integer acting as a handle to the gradient color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Gradient Color
A gradient color is a color that changes progressively across a direction.

Gradient colors are created using `BaseChart.gradientColor`, `BaseChart.gradientColor2`, `DrawArea.gradientColor` and `DrawArea.gradientColor2`. The `gradientColor` method creates a 2-point gradient color that changes from color A to color B. The `gradientColor2` method creates a multi-point gradient colors that changes from color A to B to C

These methods return a 32-bit integer acting as a handle to the gradient color. The handle can be used in any `ChartDirector` API that expects a color as its input.

One common use of multi-point gradient colors is to define colors that have metallic look and feel. Please refer to `DrawArea.gradientColor2` for details.

Dash Line Colors
A dash line color is a color that switches on and off periodically. When used to draw a line, the line will appear as a dash line.

Dash line colors are created using `BaseChart.dashLineColor` and `DrawArea.dashLineColor`. They accept a line color and a dash pattern code as arguments, and return a 32-bit integer acting as a handle to the dash line color. The handle can be used in any `ChartDirector` API that expects a color as its input.

Zone Colors
A zone color is for XY charts only. It is a color that automatically changes upon reaching a data threshold value along the x-axis or y-axis. Zone colors are created using `Layer.xZoneColor`, `Layer.yZoneColor`, `XYChart.xZoneColor` or `XYChart.yZoneColor`.

Palette Colors
Palette colors are colors of the format `FFFFnnnn`, where the least significant 16 bits (`nnnn`) are the index to the palette. A palette is simply an array of colors. For a palette color, the actual color is obtained by

looking up the palette using the index. For example, the color FFFF0001 is the second color in the palette (first color is index 0).

The colors in the palette can be ARGB colors or "dynamic" colors (pattern, gradient and dash line colors).

The first eight palette colors have special significance. The first three palette colors are the background color, default line color, and default text color of the chart. The 4th to 7th palette colors are reserved for future use. The 8th color is a special dynamic color that is equal to the data color of the "current data set".

The 9th color (index = 8) onwards are used for automatic data colors. For example, in a pie chart, if the sector colors are not specified, ChartDirector will automatically use the 9th color for the first sector, the 10th color for the second sector, and so on. Similarly, for a multi-line chart, if the line colors are not specified, ChartDirector will use the 9th color for the first line, the 10th color for the second line, and so on.

The ChartDirector API defines several constants to facilitate using palette colors.

ConstantValueDescription

Palette	FFFF0000	The starting point of the palette. The first palette color is (Palette + 0). The nth palette color is (Palette + n - 1).
BackgroundColor	FFFF0000	The background color.
LineColor	FFFF0001	The default line color.
TextColor	FFFF0002	The default text color.
[Reserved]	FFFF0003 - FFFF0006	These palette positions are reserved. Future versions of ChartDirector may use these palette positions for colors that have special significance.
SameAsMainColor	FFFF0007	A dynamic color that is equal to the data color of the current data set. This color is useful for objects that are associated with data sets. For example, in a pie chart, if the sector label background color is SameAsMainColor, its color will be the same as the corresponding sector color.
DataColor	FFFF0008	The starting point for the automatic data color allocation.

When a chart is created, it has a default palette. You may modify the palette using BaseChart.setColor, BaseChart.setColors, or BaseChart.setColors2.

The advantages of using palette colors are that you can change the color schemes of the chart in one place. ChartDirector comes with several built-in palettes represented by the following predefined constants.

ConstantDescription

20.0.25 ChartDirector: Font Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

defaultPalette	An array of colors representing the default palette. This palette is designed for drawing charts on white backgrounds (or lightly colored backgrounds).
whiteOnBlackPalette	An array of colors useful for drawing charts on black backgrounds (or darkly colored backgrounds).
transparentPalette	An array of colors useful drawing charts on white backgrounds (or lightly colored backgrounds). The data colors in this palette are all semi-transparent.

Answer: ChartDirector: Font Specification

Notes: Font Name

In ChartDirector, the font name is simply the file name that contains the font. For example, under the Windows platform, the "Arial" font is "arial.ttf", while the "Arial Bold" font is "arialbd.ttf".

NOTE: Mac OS X Specific Information

In Mac OS X, in addition to ".ttf", ChartDirector also supports Mac OS X font file formats, such as Font Suitcase files and Datafork files (.dfont). These files often contain multiple fonts. For example, the "GillSans.dfont" file contains 6 fonts.

So in addition to the file name, an index is needed to determine the font. The index is specified by appending a " | " character to the font name, followed by the index number. For example, the third font in "GillSans.dfont" is denoted as "GillSans.dfont | 2". (Note: The first font starts at 0.) If no index number is provided, the first font is assumed.

ChartDirector also supports using Mac OS X Font Manager names. For example, one may use "Gill Sans Light Italic" instead of using "GillSans.dfont | 1" as the font name. However, the Mac OS X Font Manager is active only if someone has logged into the Mac GUI console, so this method is only recommended for developing applications that run on the GUI console.

The sample programs that come with ChartDirector are designed to run on all operating systems, so they use generic font file names (eg. "arial.ttf") instead of Mac OS X specific names. To allow them to run on Mac OS X, ChartDirector on Mac OS X has a built-in table to map common font file names to Mac OS X font names:

"arial.ttf", "arialbd.ttf", "ariali.ttf" and "arialbi.ttf" are mapped to "Arial | 0" (Arial), "Arial | 1" (Arial Bold), "Arial | 2" (Arial Italic) and "Arial | 3" (Arial Bold Italic)

"times.ttf", "timesbd.ttf", "timesi.ttf" and "timesbi.ttf" are mapped to "Times New Roman | 0" (Times New Roman), "Times New Roman | 1" (Times New Roman Bold), "Times New Roman | 2" (Times New Roman Italic) and "Times New Roman | 3" (Times New Roman Bold Italic)

"cour.ttf", "courbd.ttf", "couri.ttf" and "courbi.ttf" are mapped to "Courier New | 0" (Courier New), "Courier New | 1" (Courier New Bold), "Courier New | 2" (Courier New Italic) and "Courier New | 3" (Courier New Bold Italic)

Font Location

ChartDirector on Windows does not come with any font files. It relies on the operating system's font files in the " [windows] \Fonts" directory. To see what fonts are installed in your operating system and their file names, use the File Explorer to view that directory.

ChartDirector on Windows will also search for the font files in the "fonts" subdirectory (if it exists) under the directory where the ChartDirector DLL "chartdir.dll" is installed. This is useful for private fonts. Also, for some especially secure web servers, the web anonymous user may not have access to the " [windows] \Fonts" directory. In this case, you may copy the font files to the above subdirectory.

ChartDirector on Mac OS X relies on operating system font files in "/Library/Fonts" and "/System/Library/Fonts".

ChartDirector on Linux, FreeBSD and Solaris assume the fonts files are in the "fonts" subdirectory under the directory where the ChartDirector shared object "libchartdir.so" is installed. ChartDirector on Linux, FreeBSD and Solaris come with a number of font files in the "fonts" subdirectory.

To keep the download size small, ChartDirector on Linux, FreeBSD and Solaris only come with some commonly used fonts. You may download additional fonts from the Internet. In particular, the Microsoft fonts at

http://sourceforge.net/project/showfiles.php?group_id=34153&release_id=105355

is highly recommended. Please refer to

<http://www.microsoft.com/typography/faq/faq8.htm>

on how you could use the fonts legally in your system.

ChartDirector supports True Type fonts (.ttf), Type 1 fonts (.pfa and .pfb) and Windows bitmap fonts (.fon). On Mac OS X, ChartDirector also supports Font Suitcase and Datafork (.dfont) files. On Linux, FreeBSD and Solaris, ChartDirector also supports Portable Compiled Fonts (.pcf fonts).

If you want ChartDirector to search other directories for the font files, you may list the directories in an environment variable called "FONTSPATH".

If you specify an absolute path name for the font file, ChartDirector will use the absolute path name and will not search other directories.

Artificial Boldening and Italicizing

Whereas most popular font comes with different styles for "normal", "bold", "italic" and "bold italic", some fonts only come with one style (the normal style). For example, the Monotype Corsiva font that comes with MS Office only has the normal style (mtcorsva.ttf). For these cases, you may append the "Bold" and/or "Italic" words after the font file name (separated with a space) to ask ChartDirector to artificially bolden and/or italicize the font. For example, you may specify the font name as "mtcorsva.ttf Bold".

Font List

Instead of specifying a single font file as the font name, you may specify a list of font files as the font name, separated by semi-colons. This is useful when using international characters that are only available in some fonts.

For example, if you would like to use the Arial font ("arial.ttf") for western characters, and the MingLiu font "mingliu.ttc" for Chinese characters (since the Arial font does not have Chinese characters), you may specify the font name as "arial.ttf;mingliu.ttc". In this case, ChartDirector will try the Arial font first. If it cannot find a certain character there, it will try the MingLiu font.

ChartDirector supports several special keywords for specifying the font name indirectly. When these keywords are used as font names, ChartDirector will look up the actual font names from a font table. The keywords are as follows:

KeywordsDescription

"normal"	This default normal font, which is the first font in the font table. This is initially mapped to "arial.ttf" (Arial).
"bold"	The default bold font, which is the second font in the font table. This is initially mapped to "arialbd.ttf" (Arial Bold).
"italic"	The default italic font, which is the third font in the font table. This is initially mapped to "ariali.ttf" (Arial Italic).
"boldItalic"	The default bold-italic font, which is the fourth font in the font table. This is initially mapped to "arialbi.ttf" (Arial Bold Italic).
"fontN"	The (N + 1)th font in the font table (the first font is "font0").

The font table can be modified using `BaseChart.setFontTable` or `DrawArea.setFontTable`.

The advantage of using indirect font names is that you can change the fonts in your charts in one place.

Most font files contain one font. However, it is possible a font file contains multiple fonts (that is, a font collection). For example, in True Type fonts, font files with extension ".ttc" may represent a font collection.

If a font file contains multiple font, the font index can be used to specify which font to use. By default, the font index is 0, which means the first font in the font file will be used.

The font size decides how big a font will appear in the image. The font size is expressed in a font unit called points. This is the same unit used in common word processors.

Instead of specifying font size, some ChartDirector API (eg. `TextBox.setFontSize`) allow you to specify font height and font width separately. You may use different point sizes for font height and font width to create special effects.

This is the color to draw the font. (See Color Specification on how colors are represented in ChartDirector.)

This is the angle in degrees by which the font should be rotated anti-clockwise.

By default, text are laid out horizontally, with characters being drawn from left to right.

ChartDirector also supports vertical layout, with characters being drawn from top to bottom. For example, you may use `BaseChart.addText` to add text that are laid out vertically. Vertical layout is common for

oriental languages such as Chinese, Japanese and Korean.

20.0.26 ChartDirector: Mark Up Language

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Mark Up Language

Notes: ChartDirector Mark Up Language (CDML) is a language for including formatting information in text strings by marking up the text with tags.

CDML allows a single text string to be rendered using multiple fonts, with different colors, and even embed images in the text. **Font Styles**

You can change the style of the text by using CDML tags. For example, the line:

```
<*font=timesi.ttf,size=16,color=FF0000>Hello <*font=arial.ttf,size=12,color=8000*>world!
```

will result in the following text rendered:

In general, all tags in CDML are enclosed by <*> and *>. Attributes within the tags determine the styles of the text following the tags within the same block.

If you want to include <*> in text without being interpreted as CDML tags, use «* as the escape sequence.

The following table describes the supported font style attributes in CDML. See [Font Specification](#) for details on various font attributes.

Attribute	Description
super	Set the following text to be in superscript style. This attribute does not need to have a value. (You may use "super" as the attribute instead of "super=1".)

Note that unlike HTML tags, no double or single quotes are used in the tags. It is because CDML tags are often embedded as string literals in source code. The double or single quotes, if used, will conflict with the string literal quotes in the source code. Therefore in CDML, no quotes are necessary and they must not be used.

Also, unlike HTML tags, CDML uses the comma character as the delimiter between attributes. It is because certain attributes may contain embed spaces (such as the font file name). So space is not used as the delimiter and the comma character is used instead.

Note the font attribute above starts a new style section, while other attributes just modify the current style

font	Starts a new style section, and sets the font name. You may use this attribute without a value (that is, use "font" instead of "font=arial.ttf") to create a new style section without modifying the font name.
size	The font size.
width	The font width. This attribute is used to set the font width and height to different values. If the width and height are the same, use the size attribute.
height	The font height. This attribute is used to set the font width and height to different values. If the width and height are the same, use the size attribute.
color	The text color in hex format.
bgColor	The background color of the text in hex format.
underline	The line width of the line used to underline the following characters. Set to 0 to disable underline.
sub	Set the following text to be in subscript style. This attribute does not need to have a value. (You may use "sub" as the attribute instead of "sub=1".)
super	Set the following text to be in superscript style.
xoffset	Draw the following the text by shifting the text horizontally from the original position by the specified offset in pixels.
yoffset	Draw the following the text by shifting the text vertically from the original position by the specified offset in pixels.
advance	Move the cursor forward (to the right) by the number of pixels as specified by the value this attribute.
advanceTo	Move the cursor forward (to the right) to the position as specified by the value this attribute. The position is specified as the number of pixels to the right of the left border of the block. If the cursor has already passed through the specified position, the cursor is not moved.

section. You may use `</font*>` to terminate a style section, which will restore the font styles to the state before the style section.

Blocks and Lines

In CDML, a text string may contain multiple blocks. A block may contain multiple lines of text by separating them with new line characters ("`\n`") or with `<br*>`. The latter is useful for programming languages that cannot represent new line characters easily.

For example, the line:

```
<*size=15*><*block*><*color=FF*>BLOCK<*br*>ONE<*/*>and <*block*><*color=FF00*>BLOCK<*br*>TWO
```

will result in the following text rendered:

The above example contains a line of text. The line contains two blocks with the characters " and " in between. Each block in turn contains two lines. The blocks are defined using `<*block*>` as the start tag and

`<*/*>` as the end tag.

When a block ends, font styles will be restored to the state before entering the block. Embedding Images
CDML supports embedding images in text using the following syntax:

```
<*img=my_image_file.png*>
where my_image_file.png is the path name of the image file.
```

For example, the line:

```
<*size=20*>A <*img=sun.png*>day
will result in the following text rendered:
```

ChartDirector will automatically detect the image file format using the file extension, which must either png, jpg, jpeg, gif, wbmp or wmp (case insensitive).

Please refer to `BaseChart.setSearchPath` or `DrawArea.setSearchPath` on the directory that ChartDirector will search for the file.

The `<*img*>` tag may optionally contain width and height attributes to specify its pixel width and height. In this case, ChartDirector will stretch or compress the image if necessary to the required width and height. Blocks Attributes

CDML supports nesting blocks, that is, a block can contain other sub-blocks. Attributes are supported in the `<*block*>` tag to control the alignment and orientation of the sub-blocks. The `<*img=my_image_file.png*>` is treated as a block for layout purposes.

For example, the line:

```
<*block,valign=absmiddle*><*img=molecule.png*><*block*>Hydrazino\nMolecule<*/*><*/*>
will result in the following text rendered:
```

The the above starts `<*block,valign=absmiddle*>` which specifies its content should align with each others in the vertical direction using the absolute middle alignment. The block contains an image, followed by a space characters, and then another block which has two lines of text.

The following table describes the supported attributes inside `<*block*>` tag:

Attribute	Description
-----------	-------------

The value `baseline` means the baseline of sub-blocks should align with the baseline of the block. The `baseline`

width	The width of the block in pixels. By default, the width is automatically determined to be the width necessary for the contents of the block. If the width attribute is specified, it will be used as the width of the block. If the width is insufficient for the contents, the contents will be wrapped into multiple lines.
height	The height of the block in pixels. By default, the height is automatically determined to be the height necessary for the contents of the block. If the height attribute is specified, it will be used as the height of the block.
maxwidth	The maximum width of the block in pixels. If the content is wider than maximum width, it will be wrapped into multiple lines.
truncate	The maximum number of lines of the block. If the content requires more than the maximum number of lines, it will be truncated. In particular, if truncate is 1, the content will be truncated if it exceeds the maximum width (as specified by maxwidth or width) without wrapping. The last few characters at the truncation point will be replaced with "...".
linespacing	The spacing between lines as a ratio to the default line spacing. For example, a line spacing of 2 means the line spacing is two times the default line spacing. The default line spacing is the line spacing as specified in the font used.
bgColor	The background color of the block in hex format.
valign	The vertical alignment of sub-blocks. This is for blocks that contain sub-blocks. Supported values are baseline, top, bottom, middle and absmiddle.

is the underline position of text. This is normal method of aligning text, and is the default in CDML. For images or blocks that are rotated, the baseline is the same as the bottom.

The value top means the top line of sub-blocks should align with the top line of the block.

The value bottom means the bottom line of sub-blocks should align with the bottom line of the block.

The value middle means the middle line of sub-blocks should align with the the middle line of the block. The middle line is the middle position between the top line and the baseline.

The value absmiddle means the absolute middle line of sub-blocks should align with the absolute middle line of the block. The absolute middle line is the middle position between the top line and the bottom line.

halign The horizontal alignment of lines. This is for blocks that contain multiple lines. Supported values are left, center and right.

The value left means the left border of each line should align with the left border of the block. This is the default.

The value center means the horizontal center of each line should align with the horizontal center of the block.

The value right means the right border of each line should align with the right border of the block.

angle Rotate the content of the block by an angle. The angle is specified in degrees in counter-clockwise direction.

20.0.27 ChartDirector: Parameter Substitution and Formatting

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Parameter Substitution and Formatting

Notes: ChartDirector charts often contain a lot of text strings. For example, sector labels in pie charts, axis labels for x and y axes, data labels for the data points, HTML image maps, etc, are all text strings.

ChartDirector uses parameter substitution to allow you to configure precisely the information contained in the text and their format.

Format Strings

In parameter substitution, format strings are used to specify the entities to be include into labels and how to format numbers and dates.

For example, when drawing a pie chart with side label layout, the default sector label format string is:

```
" { label } ( { percent } %)"
```

When the sector label is actually drawn, ChartDirector will replace " { label } " with the sector name, and " { percent } " with the sector percentage. So the above label format will result is a sector label similar to "ABC (34.56%)".

You may change the sector label format by changing the format string. For example, you may change it to:

```
" { label } : US$ { value | 2 } K ( { percent } %)"
```

The sector label will then become something like "ABC: US\$ 123.00 (34.56%)".

In general, in ChartDirector parameter substitution, parameters enclosed by curly brackets will be substituted with their actual values when creating the texts.

For parameters that are numbers or dates/times, ChartDirector supports a special syntax in parameter substitution to allow formatting for these values. Please refer to the Number Formatting and Date/Time Formatting sections below for details.

Parameter Expressions

ChartDirector supports numeric expressions in format strings. They are denoted by enclosing the expression with curly brackets and using "=" as the first character. For example:

```
"USD { value } (Euro { = { value } *0.9 } )"
```

In the above, " { value } " will be substituted with the actual value of the sector. The expression " { = { value } *0.9 } " will be substituted with the actual value of the sector multiplied by 0.9.

ChartDirector parameter expressions support operators "+", "-", "*", "/", "%" (modulo) and "^" (exponentiation). Operators "*", "/", "%", "^" is computed first, followed by "+" and "-". Operators of the same precedence are computed from left to right). Parenthesis "(" and ")" can be used to change the computation order.

Parameters for Pie Charts

The following table describes the parameters available for pie charts.

Parameter	Description
sector	The sector number. The first sector is 0, while the nth sector is (n-1).
dataSet	Same as { sector } . See above.
label	The text label of the sector.
dataSetName	Same as { label } . See above.
value	The data value of the sector.
percent	The percentage value of the sector.
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using BaseChart.addExtraField or BaseChart.addExtraField2.

Parameters for All XY Chart Layers

The followings are parameters that are apply to all XY Chart layers in general. Some layer types may have additional parameters (see below).

Note that certain parameters are inapplicable in some context. For example, when specifying the aggregate label of a stacked bar chart, the { dataSetName } parameter is inapplicable. It is because a stacked bar is composed of multiple data sets. It does not belong to any particular data set and hence does not have a data set name.

{ fieldN } means the extra field is indexed by the data point number. The Pth data point corresponds to the Pth element of the extra field.

Additional Parameters for Line Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Trend Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Box-Whisker Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for HLOC and CandleStick Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Additional Parameters for Vector Layers

The followings are parameters that are in additional to the parameters for all XY Chart layers.

Parameters for All Polar Layers

The followings are parameters that are apply to all Polar Chart layers in general. Some layer types may have additional parameters (see below).

{ fieldN } means the extra field is indexed by the data point number. The Pth data point corresponds to the Pth element of the extra field.

Additional Parameters for PolarVector Layers

The followings are parameters that are in additional to the parameters for all Polar Chart layers.

Parameters for Axis

The following table describes the parameters available for pie charts.

Number Formatting

For parameters that are numbers, ChartDirector supports a number of formatting options in parameter substitution.

For example, if you want a numeric field { value } to have a precision of two digits to the right of the decimal point, use ',' (comma) as the thousand separator, and use '.' (dot) as the decimal point, and you may use { value | 2, . } . The number 123456.789 will then be displayed as 123,456.79.

For numbers, the formatting options are specified using the following syntax:

```
{ [ param ] | [ a ] [ b ] [ c ] [ d ] }
```

where:

If this field starts with "E" or "e", followed by a number, it means formatting the value using scientific notation with the specified number of decimal places. If the "E" or "e" is not followed by a number, 3 is assumed.

For example, { value | E4 } will format the value 10.3 to 1.0300E+1, and { value | e4 } will format the same value to 1.0300e+1.

If this field starts with "G" or "g", followed by a number, it means formatting the value using the scientific notation only if the value is large and requires more than the specified number of digits, or the value is less than 0.001. If scientific notation is used, the number following "G" or "g" also specifies the number of significant digits to use. If the "G" or "g" is not followed by a number, 4 is assumed.

For example, consider the format string { value | G4 } . The value 10 will be formatted to 10. The value 100000 will be formatted to 1.000E+5. Similarly, for { value | g4 } , the value 10 will be formatted to 10, while the value 100000 will be formatted to 1.000e+5.

If you skip this argument, ChartDirector will display the exact value using at most 6 decimal places.

You may skip [b] [c] [d] . In this case, the default will be used.

Date/Time Formatting

For parameters that are dates/times, the formatting options can be specified using the following syntax:

```
{ [ param ] | [ datetime_format_string ] }
```

where [datetime_format_string] must start with an english character (A-Z or a-z) that is not "G", "g", "E" or "e", and may contain any characters except ' ' . (If it starts with "G", "g", "E" or "e", it will be considered as a number format string.)

Certain characters are substituted according to the following table. Characters that are not substituted will be copied to the output.

For example, a parameter substitution format of { value | mm-dd-yyyy } will display a date as something similar to 09-15-2002. A format of { value | dd/mm/yy hh:nn:ss a } will display a date as something similar to 15/09/02 03:04:05 pm.

If you want to include characters in the format string without substitution, you may enclose the characters in single or double quotes.

For example, the format { value | mmm '<*color=dd0000*>'yyyy } will display a date as something like Jan <*color=dd0000*>2005 (the <*color=dd0000*> is a CDML tag to specify red text color). Note that the <*color=dd0000*> tag is copied directly without substitution, even it contains "dd" which normally will be substituted with the day of month.

Escaping URL/HTML/CDML characters

Parameter substitution is often used to create HTML image maps. In HTML, some characters has special meanings and cannot be used reliably. For example, the '>' is used to represent the end of an HTML tag.

Furthermore, if the field happens to be used as an URL, characters such as '?', '&' and '+' also have special meanings.

By default, ChartDirector will escape template fields used in URL and query parameters when generating image maps. It will modify URL special characters to the URL escape format "%XX" (eg. "?" will become "%3F"). After that, it will modify HTML special characters to the HTML escape format "&#nn;" (eg. ">" will become ">"). Similarly, it will escape other attributes in the image map using HTML escape format (but not URL escape format).

In addition to escaping HTML and URL special characters, ChartDirector will also remove CDML fields in creating image maps. It is because CDML is only interpreted in ChartDirector, should not be useful outside of ChartDirector (such as in browser tool tips).

In some cases, you may not want ChartDirector to escape the special characters. For example, if the parameters have already been escaped before passing to ChartDirector, you may want to disable ChartDirector from escaping them again.

ChartDirector supports the following special fields to control the escape methods - " { escape_url } ", " { noescape_url } ", " { escape_html } ", " { noescape_html } ", " { escape_cdml } " and " { noescape_cdml } ". These fields enable/disable the escape methods used in the template fields that follow them.

20.0.28 ChartDirector: Shape Specification

Plugin Version: 8.2, Platforms: macOS, Linux, Windows.

Answer: ChartDirector: Shape Specification

Notes: Several ChartDirector API accept shape specification as arguments. For example, BarLayer.setBarShape and BarLayer.setBarShape2 can be used to specify shapes of bars in bar charts, while DataSet.setDataSymbol, DataSet.setDataSymbol4, PolarLayer.setDataSymbol and PolarLayer.setDataSymbol4 can be used to specify shapes for data symbols.

Note that in addition to shapes, in many cases ChartDirector also accepts images or custom draw objects for data representation. For example, see DataSet.setDataSymbol2, DataSet.setDataSymbol3, PolarLayer.setDataSymbol2 and PolarLayer.setDataSymbol3.

Built-In Shapes

Built-in shapes are specified as integers. The integers can be explicit constants, or can be generated by a `ChartDirector` method for parameterized shapes. For example, a circle is represented by an explicit constant `CircleShape (=7)`. On the other hand, the number representing a polygon depends on the number of sides the polygon has, so it is generated by using the `PolygonShape` method, passing in the number of sides as argument.

The following table illustrates the various `ChartDirector` shapes:

Custom Shapes

In `ChartDirector`, custom shapes are specified as an array of integers `x0, y0, x1, y1, x2, y2 ...` representing the coordinates of the vertices of the custom polygonal shape.

The polygon should be defined with a bounding square of 1000 x 1000 units, in which the x-axis is from -500 to 500 going from left to right, and the y-axis is from 0 to 1000 going from bottom to top.

`ChartDirector` will automatically scale the polygon so that 1000 units will become to the pixel size as requested by the various `ChartDirector` API.

As an example, the shape of the standard diamond shape in `ChartDirector` is represented as an array with 8 numbers:

```
0, 0, 500, 500, 0, 1000, -500, 500
```

20.0.29 Copy styled text?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: How to quickly copy styled text from one textarea to another?

Example:

```
#if TargetWin32 then
TextArea1.WinRTFDataMBS = TextArea2.WinRTFDataMBS
#elseif TargetMacOS then
TextArea1.NSTextViewMBS.textStorage.setAttributedString TextArea2.NSTextViewMBS.textStorage
#else
TextArea1.StyledText = TextArea2.StyledText
#endif
```

Notes: The code above uses special plugin functions on Mac and Windows and falls back to framework for Linux.

20.0.30 Do you have code to validate a credit card number?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can check the checksum to tell if a credit card number is not valid.

Example:

```

Dim strNumber As String
Dim nLength as Integer
Dim nValue as Integer
Dim nChecksum as Integer
Dim nIndex as Integer

strNumber = EditField1.Text
nLength = Len(strNumber)
nChecksum = 0

For nIndex = 0 To nLength - 2
nValue = Val(Mid(strNumber, nLength - (nIndex + 1), 1)) * (2 - (nIndex Mod 2))
If nValue <10 Then
nChecksum = nChecksum + nValue
Else
nChecksum = nChecksum + (nValue - 9)
End If
Next

If Val(Mid(strNumber, Len(strNumber), 1)) = (10 - (nChecksum Mod 10)) Mod 10 Then
MsgBox("The credit card number looks valid")
Else
MsgBox("The credit card number is invalid")
End IF

```

Notes: Here's some code that will validate the checksum for a credit card. It works for Visa, MasterCard, American Express and Discover. Not sure about others, but I imagine they use the same basic algorithm. Of course, this doesn't actually mean that the credit card is valid, it's only useful for helping the user catch typos.

The above code doesn't have any error checking and it expects that the credit card number will be entered without spaces, dashes or any other non-numeric characters. Addressing those issues will be an exercise left to the reader. :)

(From Mike Stefanik)

20.0.31 Do you have plugins for X-Rite EyeOne, eXact or i1Pro?

Plugin Version: all.

Answer: Our EyeOne plugin is available on request for licensees of the X-Rite SDKs.

Notes: Please first go to X-Rite and get a SDK license.

Then we can talk about the plugin.

20.0.32 Does SQL Plugin handle stored procedures with multiple result sets?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Yes, the plugin can work with multiple recordsets.

Notes: You need to use SQLCommandMBS class. When you get back results, you use FetchNext to walk over all records in the first result set. Then you simply start again with FetchNext to get the second record set.

Even the RecordSet functions should work, just use them twice to get all records from both record sets.

20.0.33 Does the plugin home home?

Plugin Version: all, Platform: macOS.

Answer: Yes, we like to know who is using the plugin, so the plugin may contact our server.

Example:

none.

Notes: Please note that this does not affect your users as the plugin will only do this in the IDE and the relevant plugin part is never included in your applications.

The plugin if used for some hours, does contact our server to provide statistical data about Xojo version and OS versions. This way we know what versions are used. We can return the version number of the current plugin which may be visible in future versions somehow. And we transmit partial licenses data so we can track use of illegal license keys.

If you do not like to have this, you can block Xojo IDE from contacting our website via your Firewall.

Blocking the transfer will not disable the plugin or change the features.

Or contact us for a plugin version which explicitly does not contain this feature.

20.0.34 folderitem.absolutePath is limited to 255 chars. How can I get longer ones?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Paths on a Mac are not unique, so use them only to display them to the user.

Example:

```
Function AbsolutePath(f as FolderItem) As String
Dim s as string
Dim nf as FolderItem
nf = f
s = ""
while nf<>nil
s = nf.name + "." + s
nf = nf.parent
wend
Return s
End Function
```

20.0.35 Has anyone played round with using CoreImage to do things like add dissolve transitions say when changing from one tab to another within a window?

Platform: macOS.

Answer: This code implements animations for a tabpanel change:

Example:

// in a tabpanel.change event:

```
dim r as CGSTransitionRequestMBS
dim co as new CGSConnectionMBS
dim cw as CGSWindowMBS
dim ct as CGSTransitionMBS
static OldTab as Integer

cw=co.CGSWindow(window1)
If cw = Nil Then
return // 10.3...
End If
r=new CGSTransitionRequestMBS
r.TransitionType=r.CGSFlip
r.HasBackGround=false
r.HasBackColor=false
r.Win=cw
```

```

// watch the value of the clicked tab versus the last tab
if tabpanel1.Value=0 or tabpanel1.Value <OldTab then
r.TransitionOption=r.CGSLeft
ct=co.NewTransition(r)
if ct<>Nil then
Refresh
ct.Invoke(1)
ct.Wait(1)
ct.Release
else
MsgBox "Error creating the transition."
end if
else
r.TransitionOption=r.CGSRight
ct=co.NewTransition(r)
if ct<>Nil then
Refresh
ct.Invoke(1)
ct.Wait(1)
ct.Release
else
MsgBox "Error creating the transition."
end if
end if
// Keep track of the last tab clicked
OldTab = tabpanel1.Value

```

Notes: See CGS* classes for more details.

20.0.36 How about Plugin support for older OS X?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We support in general Mac OS X 10.5 and newer.

Notes: All the 64-bit plugins on Mac require OS X 10.7.

Intel 32-bit plugins on Mac require OS X 10.5 or newer.

Currently the ChartDirector 6, GraphicsMagick and GameKit plugins requires Mac OS X 10.6. Also for SQL Plugin the built in SQLite library requires 10.6.

20.0.37 How can I detect whether an Intel CPU is a 64bit CPU?

Plugin Version: all.

Answer: Look on the CPU family returned by sysctl:

Example:

Function is64bit() As Boolean

```
#if TargetLittleEndian
```

```
dim m as MemoryBlock = NewMemoryBlock(8)
```

```
dim family as Integer
```

```
dim s as string
```

```
m=SystemControlNameToMIBMBS("hw.cpufamily")
```

```
m=SystemControlMBS(m)
```

```
if m<>nil then
```

```
m.LittleEndian=True
```

```
family=m.Long(0)
```

```
const CPUFAMILY_INTEL_6_14 = &h73d67300 /* "Intel Core Solo" and "Intel Core Duo" (32-bit Pentium-M with SSE3) */
```

```
const CPUFAMILY_INTEL_6_15 = &h426f69ef /* "Intel Core 2 Duo" */
```

```
const CPUFAMILY_INTEL_6_23 = &h78ea4fbc /* Penryn */
```

```
const CPUFAMILY_INTEL_6_26 = &h6b5a4cd2 /* Nehalem */
```

```
Select case family
```

```
case CPUFAMILY_INTEL_6_14
```

```
Return false
```

```
case CPUFAMILY_INTEL_6_15
```

```
Return true
```

```
case CPUFAMILY_INTEL_6_23
```

```
Return true
```

```
case CPUFAMILY_INTEL_6_26
```

```
Return true
```

```
// newer CPUs may be missing here
```

```
end Select
```

```
end if
```

```
#endif
```

```
Return false
```

```
Exception
```

```
Return false
```

```
End Function
```

Notes: This code is written for Mac OS X where you only have a limited number of possible CPUs.

20.0.38 How can I disable the close box of a window on Windows?

Plugin Version: all, Platform: Windows.

Answer: The following code will remove the close item from the system menu of the window.

Example:

```
#if TargetWin32 then
Declare Function GetSystemMenu Lib "user32" (hwnd as Integer, bRevert as Integer) as Integer
Declare Function RemoveMenu Lib "user32" (hMenu as Integer, nPosition as Integer, wFlags as Integer) as Integer
Dim hSysMenu as Integer
hSysMenu = GetSystemMenu(me.WinHWND, 0)
RemoveMenu hSysMenu, &HF060, &H0
#endif
```

Notes: The window may not be updated directly.

20.0.39 How can I get all the environment variables from Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetWin32
declare function GetEnvironmentStrings Lib "kernel32" () as ptr
dim m as memoryBlock
dim n as Integer

m=GetEnvironmentStrings()

n=0
do
msgBox m.cstring(n)
while m.byte(n)<>0
n=n+1
wend
n=n+1
```

```
loop until m.byte(n)=0
#endif
```

Notes: The MBS Plugin has an EnvironmentMBS class for this.

20.0.40 How can i get similar behavior to Roxio Toast or iTunes where clicking a 'burn' button allows the next inserted blank CD-R to bypass the Finder and be accepted by my application?

Plugin Version: all, Platform: macOS.

Answer: You need to get a media reservation.

Example:

```
dim d as DRDeviceMBS // get a device
d.AcquireMediaReservation
```

Notes: Use the plugin function AcquireMediaReservation and later release it using ReleaseMediaReservation.

See plugin examples on how to use it and check Apples DiscRecording framework documentation for more details.

20.0.41 How can I get text from a PDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Crossplatform you can use DynaPDF Pro.

Notes: On Mac OS X you can also use PDFKit for the same job.

While DynaPDF Pro gives you each bit of text with rotation, font information and encoding details, PDFKit gives you only the text string for a PDF page.

20.0.42 How can I get text from a Word Document?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: to get the text string from a doc file, use the NSAttributedStringMBS class.

Notes: The NSAttributedStringMBS class is Mac OS X only and we have currently no solution for Windows or Linux.

Use the `NSAttributedStringMBS.initWithDocFormat(data as string)` as boolean method.

20.0.43 How can I get the item string for a given file creator?

Plugin Version: all.

Answer: Try this function:

Example:

```
Sub pullNativeDocs(aCREA As string)
Dim result as Integer
Dim m, k as memoryBlock
Dim f as folderItem
Dim newType as string
Dim anIcon As picture
Dim ofs as Integer
```

```
Declare Function GetFileTypesThatAppCanNativelyOpen Lib "Carbon" (appVRefNumHint as Short, appSignature as OSType, nativeTypes as Ptr) as Short Inline68K("701CABFC")
Declare Function GetDocumentKindString Lib "Carbon" (docVRefNum as Short, docType as OSType, docCreator as OSType, kindString as ptr) as Short Inline68K("7016ABFC")
```

```
listBox1.deleteAllRows
```

```
m = newMemoryBlock(1024)
result = GetFileTypesThatAppCanNativelyOpen(Volume(0).MacVRefNum, aCREA, m)
if result <> 0 then
listBox1.addRow "<Not found.>"
return
end if
```

```
do
if m.byte(ofs*4) = 0 then
exit
else
newType = m.OSTypeMBS(ofs*4)
listBox1.addRow newType
k = newMemoryBlock(64)
result = GetDocumentKindString(Volume(0).MacVRefNum, newType, aCREA, k)
if result = 0 then
listBox1.cell(ofs,1) = k.pString(0)
ofs = ofs + 1
else
listBox1.cell(ofs,1) = "(unknown)"
end if
end if
```

loop

End Sub

Notes: Change "Translation" to "CarbonLib" for Mac OS X.

20.0.44 How can I launch an app using it's creator code?

Plugin Version: all, Platform: macOS.

Answer: Send an AppleEvent "odoc" with the creator code to the Finder ("MACS"):

Example:

```
Function LaunchByCreator(C As String) As Boolean
Dim A As AppleEvent
A = NewAppleEvent("aevt","odoc","MACS")
A.ObjectSpecifierParam("—") = GetUniqueIDObjectDescriptor("appf",nil,C)
return A.Send
End Function
```

20.0.45 How can I learn what shared libraries are required by a plugin on Linux?

Plugin Version: all, Platform: macOS.

Answer: Please use the ldd command in the terminal.

Notes: You build an app on any platform, but for Linux.

For the resulting .so files in the libs folder, you can run the ldd command with the library path as parameter. It shows you references lib files and you can make sure you have those installed.

This is a sample run of our graphicsmagick plugin:

```
cs@Ubuntu32:
textasciitilde /MeinProgramm/MeinProgramm Libs$ ldd libMBSGraphicsMagickPlugin17744.so
linux-gate.so.1 =>(0xb76ee000)
libdl.so.2 =>/lib/i386-linux-gnu/libdl.so.2 (0xb6f0e000)
libgtk-x11-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgtk-x11-2.0.so.0 (0xb6aa6000)
libpthread.so.0 =>/lib/i386-linux-gnu/libpthread.so.0 (0xb6a8a000)
libstdc++.so.6 =>/usr/lib/i386-linux-gnu/libstdc++.so.6 (0xb69a5000)
libm.so.6 =>/lib/i386-linux-gnu/libm.so.6 (0xb6979000)
libgcc_s.so.1 =>/lib/i386-linux-gnu/libgcc_s.so.1 (0xb695b000)
libc.so.6 =>/lib/i386-linux-gnu/libc.so.6 (0xb67b1000)
```

```

/lib/ld-linux.so.2 (0xb76ef000)
libgdk-x11-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgdk-x11-2.0.so.0 (0xb6701000)
libpangocairo-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpangocairo-1.0.so.0 (0xb66f4000)
libX11.so.6 =>/usr/lib/i386-linux-gnu/libX11.so.6 (0xb65c0000)
libXfixes.so.3 =>/usr/lib/i386-linux-gnu/libXfixes.so.3 (0xb65ba000)
libatk-1.0.so.0 =>/usr/lib/i386-linux-gnu/libatk-1.0.so.0 (0xb659a000)
libcairo.so.2 =>/usr/lib/i386-linux-gnu/libcairo.so.2 (0xb64ce000)
libgdk_pixbuf-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgdk_pixbuf-2.0.so.0 (0xb64ad000)
libgio-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgio-2.0.so.0 (0xb6356000)
libpangoft2-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpangoft2-1.0.so.0 (0xb632a000)
libpango-1.0.so.0 =>/usr/lib/i386-linux-gnu/libpango-1.0.so.0 (0xb62e0000)
libfontconfig.so.1 =>/usr/lib/i386-linux-gnu/libfontconfig.so.1 (0xb62ab000)
libgobject-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgobject-2.0.so.0 (0xb625c000)
libglib-2.0.so.0 =>/lib/i386-linux-gnu/libglib-2.0.so.0 (0xb6163000)
libXext.so.6 =>/usr/lib/i386-linux-gnu/libXext.so.6 (0xb6151000)
libXrender.so.1 =>/usr/lib/i386-linux-gnu/libXrender.so.1 (0xb6147000)
libXinerama.so.1 =>/usr/lib/i386-linux-gnu/libXinerama.so.1 (0xb6142000)
libXi.so.6 =>/usr/lib/i386-linux-gnu/libXi.so.6 (0xb6132000)
libXrandr.so.2 =>/usr/lib/i386-linux-gnu/libXrandr.so.2 (0xb6129000)
libXcursor.so.1 =>/usr/lib/i386-linux-gnu/libXcursor.so.1 (0xb611e000)
libXcomposite.so.1 =>/usr/lib/i386-linux-gnu/libXcomposite.so.1 (0xb611a000)
libXdamage.so.1 =>/usr/lib/i386-linux-gnu/libXdamage.so.1 (0xb6115000)
libfreetype.so.6 =>/usr/lib/i386-linux-gnu/libfreetype.so.6 (0xb607b000)
libxcb.so.1 =>/usr/lib/i386-linux-gnu/libxcb.so.1 (0xb605a000)
libpixman-1.so.0 =>/usr/lib/i386-linux-gnu/libpixman-1.so.0 (0xb5fc2000)
libpng12.so.0 =>/lib/i386-linux-gnu/libpng12.so.0 (0xb5f98000)
libxcb-shm.so.0 =>/usr/lib/i386-linux-gnu/libxcb-shm.so.0 (0xb5f93000)
libxcb-render.so.0 =>/usr/lib/i386-linux-gnu/libxcb-render.so.0 (0xb5f89000)
libz.so.1 =>/lib/i386-linux-gnu/libz.so.1 (0xb5f73000)
libgmodule-2.0.so.0 =>/usr/lib/i386-linux-gnu/libgmodule-2.0.so.0 (0xb5f6e000)
libselinux.so.1 =>/lib/i386-linux-gnu/libselinux.so.1 (0xb5f4f000)
libresolv.so.2 =>/lib/i386-linux-gnu/libresolv.so.2 (0xb5f36000)
libexpat.so.1 =>/lib/i386-linux-gnu/libexpat.so.1 (0xb5f0c000)
libffi.so.6 =>/usr/lib/i386-linux-gnu/libffi.so.6 (0xb5f05000)
libpcre.so.3 =>/lib/i386-linux-gnu/libpcre.so.3 (0xb5ec9000)
librt.so.1 =>/lib/i386-linux-gnu/librt.so.1 (0xb5ec0000)
libXau.so.6 =>/usr/lib/i386-linux-gnu/libXau.so.6 (0xb5ebb000)
libXdmcp.so.6 =>/usr/lib/i386-linux-gnu/libXdmcp.so.6 (0xb5eb4000)
cs@Ubuntu32:
textasciitilde /MeinProgramm/MeinProgramm Libs$

```

As you see all library have been found and their load address is printed behind the name. If a library is missing, you usually see the address missing there or being zero.


```

while theRegexMatch <>nil
theStart = theRegexMatch.subExpressionStartB(0) + len(theRegexMatch.subExpressionString(0))

result = result + theRegexMatch.subExpressionString(1)
infoCharset = theRegexMatch.subExpressionString(2)
encodedPart = theRegexMatch.subExpressionString(4)
if theRegexMatch.subExpressionString(3) = "B" then
encodedPart = DecodeBase64(encodedPart)
elseif theRegexMatch.subExpressionString(3) = "Q" then
encodedPart = DecodeQuotedPrintable(encodedPart)
end if
if right(result, 1) = " " then
result = mid(result, 1, len(result)-1)
end if
encodedPart = encodedPart.DefineEncoding(GetInternetTextEncoding(infoCharset))
result = result + encodedPart

theRegex.SearchStartPosition = theStart
theRegexMatch = theRegex.search()
wend

result = result + mid(src, theStart+1)

else
result = src
end if
// theRegexMatch = theRegex.search

msgbox result

```

Notes: May not look nice depending on the controls used.
This is no longer needed when using MimeEmailMBS class which decodes for you.

20.0.48 How do I enable/disable a single tab in a tabpanel?

Plugin Version: all, Platform: macOS.

Answer: Use the TabpanelEnabledMBS method.

Example:

```
TabpanelEnabledMBS(tabpanel1, 1, false)
```

Notes: Use Carbon for MachO and CarbonLib for Mac Carbon and AppearanceLib for Mac OS Classic as

library.

For Cocoa, please use enabled property of NSTabViewItemMBS class.

20.0.49 How do I find the root volume for a file?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this function:

Example:

```
Function GetRootVolume(f as FolderItem) as FolderItem
dim root, dum as folderItem
if f <> nil then
root = f // f might be the volume
do
dum = root.parent
if dum <> nil then
root = dum
end if
loop until dum = nil
return root
end if
End Function
```

20.0.50 How do I get the current languages list?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
dim p as new CFPREFERENCESMBS
dim a as CFArrayMBS
dim s as CFStringMBS
dim o as CFOBJECTMBS
dim sa(-1) as string

o=p.CopyAppValue("AppleLanguages", ".GlobalPreferences")

if o<>Nil then
a=CFArrayMBS(o)

dim i,c as Integer
```

```
c=a.Count-1
for i=0 to c
o=a.Item(i)

if o isa CFStringMBS then
s=CFStringMBS(o)
sa.Append s.str
end if
next
end if

MsgBox Join(sa,EndOfLine)
```

Notes: On Mac OS X you can get the list of current languages like this list:

```
de
en
ja
fr
es
it
pt
pt-PT
nl
sv
nb
da
fi
ru
pl
zh-Hans
zh-Hant
ko
```

Which has German (de) on the top for a German user.
This code has been tested on Mac OS X 10.5 only.

20.0.51 How do I get the Mac OS Version?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

dim i as Integer
if system.gestalt("sysv", i) then
//do this in an 'If' in case you don't get any value back at all and system.gestalt returns boolean
if i = &h750 then //If OS is 7.5
//do stuff
elseif i = &h761 then //If OS is 7.6.1
//do stuff
end if
end if

```

Notes: The MBS Plugin has a function SystemInformationMBS.OSVersionString for this.

20.0.52 How do I get the printer name?

Plugin Version: all.

Answer: For Mac OS Classic see the code below and for Mac OS X use the Carbon Print Manager Classes from the MBS Plugin.

Example:

```

dim s as String
dim i as Integer

s=app.ResourceFork.GetResource("STR ",-8192)
if s<>"" then
i=ascb(leftb(s,1))
s=mid(s,2,i)

MsgBox s
end if

```

Notes: A note from Craig Hoyt:

After looking at your example I had a little deja-vu experience. Several years ago I played around with this same code if FutureBasic. I discovered that it did not and still doesn't provide the 'Printer Name', it does return the print driver name. If it returns 'LaserWriter 8' as the print driver you can look into this file and get the 'PAPA' resource #-8192 to get the actual Printer Name. Unfortunately this does not hold true for other printers. My Epson and HP Printers (the Epson has an Ethernet Card and the HP is USB) do not provide this info in their drivers. As far as I can tell it only returns the name by polling the printer itself.

20.0.53 How do I make a metal window if RB does not allow me this?

Plugin Version: all, Platform: macOS.

Answer: The following declare turns any window on Mac OS X 10.2 or newer into a metal one.

Example:

```
declare sub ChangeWindowAttributes lib "Carbon" (win as windowptr, a as Integer, b as Integer)
```

```
ChangeWindowAttributes window1,256,0
```

Notes: May not look nice depending on the controls used.

20.0.54 How do I make a smooth color transition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

I'd like to show in a report some bars, which start with color A and end with color B.

The color change should be very smooth.

My problem: If I would start from 255,0,0 and end by 0,0,0, I would have 255 different colors. If the bars are longer than 255 pixels, would this look nice?

Example:

```
// Window.Paint:
Sub Paint(g As Graphics)
dim w,w1,x,p as Integer
dim c1,c2,c as color
dim p1,p2 as Double

c1=rgb(255,0,0) // start color
c2=rgb(0,255,0) // end color

w=g.Width
w1=w-1

for x=0 to w1
p1=x/w1
p2=1.0-p1
```

```

c=rgb(c1.red*p1+c2.red*p2, c1.green*p1+c2.green*p2, c1.blue*p1+c2.blue*p2)

g.ForeColor=c
g.DrawLine x,0,x,g.Height

next
End Sub

```

Notes:

Try the code above in a window paint event handler.

20.0.55 How do I read the applications in the dock app?

Plugin Version: all, Platform: macOS.

Answer: Use CFPreferencesMBS class like in this example:

Example:

```

// Reads file names from persistent dock applications and puts them into the list

dim pref as new CFPreferencesMBS

dim persistentapps as CFStringMBS = NewCFStringMBS("persistent-apps")
dim ApplicationID as CFStringMBS = NewCFStringMBS("com.apple.dock")
dim tiledata as CFStringMBS = NewCFStringMBS("tile-data")
dim filelabel as CFStringMBS = NewCFStringMBS("file-label")

// get the array of persistent applications from dock preferences
dim o as CObjectMBS = pref.CopyValue(persistentapps, ApplicationID, pref.kCFPreferencesCurrentUser,
pref.kCFPreferencesAnyHost)

if o isa CFArrayMBS then
dim a as CFArrayMBS = CFArrayMBS(o)

// walk over all items in array
dim c as Integer = a.Count-1
for i as Integer = 0 to c

// get dictionary describing item
o = a.Item(i)

if o isa CFDictionaryMBS then
dim d as CFDictionaryMBS = CFDictionaryMBS(o)

```

```

// and pick tile data dictionary
o = d.Value(tiledata)
if o isa CFDictionaryMBS then
d = CFDictionaryMBS(o)

// and pick there the file label
o = d.Value(filelabel)
if o isa CFStringMBS then
// and display it
dim name as string = CFStringMBS(o).str
List.AddRow name
end if
end if
end if

next

else
MsgBox "Failed to read dock preferences."
end if

```

Notes: You can use the `CFPreferencesMBS.SetValue` to change a value and `CFPreferencesMBS.Synchronize` to write the values to disc. You may need to restart the `Dock.app` if you modified things.

20.0.56 How do I truncate a file?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In a `binarystream` you can set the `length` property to truncate.

20.0.57 How do update a Finder's windows after changing some files?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

dim f as folderitem // some file
dim ae as appleevent
ae=newappleevent("fndr","fupd","MACS")
ae.folderitemparam("—")=f
if not ae.send then
//something went wrong

```

end if

Notes: The `folderitem.finderupdate` from the MBS Plugin does something like this.

20.0.58 How to access a USB device directly?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: First, it depends on the device.

Notes: Some devices can be talked directly from user mode code, but some require a kernel driver.

For some devices you can use plugins to access them like:

- Audio and Video sources using the `QTGrabberClassMBS`
- Mass storage devices using the `folderitem` class.
- Serial devices using the `System.SerialPort` function.
- HID USB devices can be used with `MacHIDMBS`, `WinHIDMBS` or `LinuxHIDInterface` class.
- Any USB device may be used with `MacUSBMBS` or `WinUSBMBS` classes.

In general it is always the best to take the most high level access to have others do the work for the details.

20.0.59 How to add icon to file on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use `FolderItem.AddCustomIcon` or `NSWorkspaceMBS.setIcon` functions.

Notes: Please close any open stream for the file you want to add an icon.

20.0.60 How to ask the Mac for the Name of the Machine?

Plugin Version: all, Platform: macOS.

Answer: Using Apple Events you can use this code:

Example:

Function `Computername()` *As string*

```

dim theEvent as AppleEvent
dim err as boolean

theEvent = newAppleEvent("mchn","getd","MACS")

err = theEvent.send

return theevent.ReplyString

End Function

```

Notes: Code above is for Mac OS 9!

Also the MBS Plugin has a function for this which may be faster and work also on Macs without Filesharing (which handles this event).

20.0.61 How to automatically enable retina in my apps?

Plugin Version: all, Platform: macOS.

Answer: You can run a build script on each build with this code:

Example:

```

Dim App As String = CurrentBuildLocation + "/" + CurrentBuildAppName + ".app"
Call DoShellCommand("/usr/bin/defaults write " + App + "/Contents/Info ""NSHighResolutionCapable""
YES")

```

Notes: This will set the NSHighResolutionCapable flag to YES.

20.0.62 How to avoid leaks with Cocoa functions?

Plugin Version: all, Platform: macOS.

Answer: You can try this code on Mac OS X:

Example:

```

// in a Timer Action event:
Sub Action()
static LastPool as NSAutoreleasePoolMBS = nil
static CurrentPool as NSAutoreleasePoolMBS = nil

```

```

LastPool = CurrentPool
CurrentPool = new NSAutoreleasePoolMBS

```

End Sub

Notes: With Xojo 2009r4 the code above should not be needed as Xojo runtime does automatically handle the `NSAutoreleasePools` for you. For older Xojo versions you need to use code with a timer with the action event above to avoid memory leaks.

Please do not use Xojo 2009r4 and newer with plugins before version 9.5. You can get crashes there which typically show a line with a `objc_msgSend` call.

20.0.63 How to avoid trouble connecting to oracle database with SQL Plugin?

Plugin Version: all, Platform: macOS.

Answer: For oracle the most important thing is to point the plugin to the libraries from oracle.

Notes: In environment variables, the paths like `ORACLE_HOME` must be defined.

On Mac OS X you also need to define `DYLD_LIBRARY_PATH` to point to the dylib files from oracle.

For that you need to modify `/etc/launchd.conf` for Mac OS X 10.8 and newer.

In older versions those variables in `.MacOSX/environment.plist` file in user's home.

Another way for the case you bundle things inside your app is to use the `LSEnvironment` key in `info.plist`. In `info.plist` it looks like this:

```
<key>LSEnvironment</key>
<dict>
<key>test</key>
<string>Hello World</string>
</dict>
```

20.0.64 How to avoid `___NSAutoreleaseNoPool` console messages in threads?

Plugin Version: all, Platform: macOS.

Answer: You need to use your own `NSAutoreleasePool` on a thread like this:

Example:

```
sub MyThread.run
dim pool as new NSAutoreleasePoolMBS
// do work here

pool=nil
```

end sub

Notes: For more details read here:

http://developer.apple.com/mac/library/documentation/Cocoa/Reference/Foundation/Classes/NSAutoreleasePool_Class/Reference/Reference.html

20.0.65 How to bring app to front?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac you can use this code:

Example:

// First way:

```
app.FrontMostMBS = true
```

// second way:

```
dim p as new ProcessMBS
p.GetCurrentProcess
p.FrontProcess = true
```

// third way:

```
NSApplicationMBS.sharedApplication.activateIgnoringOtherApps(true)
```

// for Windows:

```
RemoteControlMBS.WinBringWindowToTop
```

Notes: This will bring a Mac app to the front layer.

20.0.66 How to bring my application to front?

Plugin Version: all, Platform: macOS.

Answer: This makes SimpleText (Code txtxt) to the frontmost application:

Example:

```
Dim A As AppleEvent
A = NewAppleEvent("misc", "actv", "")
If Not A.Send then
  Beep
end if
```

Notes: (Code is Mac only)

20.0.67 How to catch Control-C on Mac or Linux in a console app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use SignalHandlerMBS class for this.

Example:

```
// watch for Control-C on Mac
call SignalHandlerMBS.SetFlagHandler(2)

dim ende as boolean = false
do
if SignalHandlerMBS.IsFlagSet(2) then
Print "Flag 2 set. Existing..."
ende = true
end if

DoEvents 1
loop until ende
```

Notes: The signal is caught, a flag is set and you can ask later in your normal application flow for the result.

20.0.68 How to change name of application menu?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Use this code to change the application menu name on Mac OS X:

Example:

```
dim mb as new MenubarMBS
dim m as MenuMBS = mb.item(1) // 1 is in my tests the app menu
if m<>Nil then
m.MenuTitle = "Hello World"
end if
```

Notes: This code is for Carbon only.

20.0.69 How to change the name in the menubar of my app on Mac OS X?

Plugin Version: all, Platform: macOS.

Answer:

You mean it screws up if the file name of the bundle itself is different than the name of the executable file in the MacOS folder within the bundle? If so, you should find something like this within your Info.plist file (or the 'plst' resource that the RB IDE builds for you):

```
<key>CFBundleExecutable</key>
<string>Executable file name here</string>
```

Just make sure that file name matches.

However, if your question involves how you can change the name of the app that appears in the menu and the dock, that's different. You can make this name different from the file name by changing the CFBundleName key:

```
<key>CFBundleName</key>
<string>Name for menu here</string>
```

Note that if you use my free AppBundler program, this second part is taken care of for you – just fill in a custom name in the right field. You can find AppBundler (from Thomas Reed) at <http://www.bitjuggler.com/products/appbundler/>.

20.0.70 How to check if a folder/directory has subfolders?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this to check all items in a folder:

Example:

```
Function HasSubFolder(folder as FolderItem) As Boolean
dim c as Integer = folder.Count
```

```
for i as Integer = 1 to c
dim item as FolderItem = folder.TrueItem(i)
```

```
if item<>Nil and item.Directory then
Return true
end if
```

next

End Function

Notes: We use trueitem() here to avoid resolving alias/link files.
Also we check for nil as we may not have permission to see all items.
And if one is a directory, we return without checking the rest.

20.0.71 How to check if Macbook runs on battery or AC power?

Plugin Version: all, Platform: macOS.

Answer: Please use our IOPowerSourcesMBS class like this:

Example:

```
Function PowerSourceState() as Integer
dim p as new IOPowerSourcesMBS

// check all power sources
dim u as Integer = p.Count-1
for i as Integer = 0 to u
dim d as CFDictionaryMBS = p.Item(i)
if d<>nil then
// check if they have a power source state key:
dim o as CFObjectMBS = d.Value(NewCFStringMBS("Power Source State"))
if o isa CFStringMBS then
dim s as string = CFStringMBS(o).str

'MsgBox s

if s = "AC Power" then
Return 1
elseif s = "Battery Power" then
Return 2
end if
end if
end if
next
Return 0 // unknown
End Function
```

Notes: If you want to check the CFDictionaryMBS content, simply use a line like "dim x as dictionary = d.dictionary" and check the contents in the debugger.

20.0.72 How to check if Microsoft Outlook is installed?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you need Outlook for Scripting, you should simply check registry for the required Outlook.Application class:

Example:

```
Function OutlookInstalled() As Boolean
    #if TargetWin32 then

    try
    dim r as new RegistryItem("HKEY_CLASSES_ROOT\Outlook.Application\CLSID", false)

    Return true

    catch r as RegistryAccessErrorException
    // not installed
    Return false

    end try

    #else

    // Windows only, so false on other platforms
    Return false

    #endif

End Function
```

20.0.73 How to check on Mac OS which country or language is currently selected?

Plugin Version: all, Platform: macOS.

Answer: The code below returns a country value.

Example:

```
dim result as Integer

IF TargetMacOS THEN
```

```

CONST smScriptLang = 28
CONST smSystemScript = -1

DECLARE FUNCTION GetScriptManagerVariable LIB "Carbon" ( selector as Integer) as Integer
DECLARE FUNCTION GetScriptVariable LIB "Carbon" ( script as Integer, selector as Integer) as Integer

result=GetScriptVariable(smSystemScript, smScriptLang)

END IF

```

Notes: Returns values like:

For more values, check "Script.h" in the frameworks.

20.0.74 How to code sign my app with plugins?

Plugin Version: all, Platform: macOS.

Answer: When you try to code sign the application with plugin dylibs on Mac OS X, you may see error message that there is actually a signature included.

Notes: Please use the -f command line parameter with codesign utility to overwrite our MBS signature. We sign our plugins for MacOS, iOS and Windows to make sure they have not been modified.

In terminal, you do like this:

```

cd <Path to folder of app>

xattr -cr <Appname>.app
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app/Contents/Frameworks/*.dylib
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app/Contents/Frameworks/*.framework
codesign -f -s "Developer ID Application: <Your Name>" <Appname>.app

```

Please use the name of your certificate (See keychain), the name of your app and the path to the app folder. If you have helper apps you need to sign them first. You can use a build step to automatically sign your app on build.

20.0.75 How to collapse a window?

Plugin Version: all, Platform: macOS.

Answer: Use this function (Mac only):

Example:

```
Sub CollapseRBwindow(w as window, CollapseStatus as boolean)
dim state, err as Integer
dim wh as MemoryBlock
```

```
Declare Function CollapseWindow Lib "Carbon" (window as Integer, collapse as Integer) as Integer
```

```
IF CollapseStatus THEN
state = 1
ELSE
state = 0
END IF
```

```
err = CollapseWindow(w.MacWindowPtr, state)
```

```
End Sub
```

Notes: Also the MBS Plugin has a window.collapsedmbs property you can set. For Windows the MBS Plugin has a window.isiconicmbs property.

20.0.76 How to compare two pictures?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can try this code:

Example:

```
Function ComparePictures(p as picture,q as picture) as Integer
dim r,u as RGBSurface
dim x,y,n,m,h,w as Integer
dim w1,w2,h1,h2,d1,d2 as Integer
dim c1,c2 as color
```

```
h1=p.Height
h2=q.Height
w1=p.Width
w2=q.Width
d1=p.Depth
d2=q.Depth
```

```
if d1<>d2 then
Return 1
elseif w1<>w2 then
```

```
return 2
elseif h1<>h2 then
Return 3
else
r=p.RGBSurface
u=q.RGBSurface

if r=nil or u=nil then
Return -1
else
h=h1-1
w=w1-1
m=min(w,h)

for n=0 to m
c1=r.Pixel(n,n)
c2=u.Pixel(n,n)
if c1<>c2 then
Return 4
end if
next

for y=0 to h
for x=0 to w
c1=r.Pixel(x,y)
c2=u.Pixel(x,y)
if c1<>c2 then
Return 5
end if
next
next

// 0 for equal
// -1 for error (no RGBsurface)
// 1 for different depth
// 2 for different width
// 3 for different height
// 4 for different pixels (fast test)
// 5 for different pixels (slow test)
end if
end if

Exception
Return -1
End Function
```

Notes: Remember that this only works on bitmap pictures, so the `picture.BitmapMBS` function may be useful.

20.0.77 How to compile PHP library?

Plugin Version: all, Platform: macOS.

Answer: You have to download the source code and compile a static version of the library.

Notes: This instructions were written based on PHP 5.2.6 on Mac OS X:

- Best take a new Mac with current Xcode version installed.
- Download the source code archive. e.g. "php-5.2.6.tar.bz2"
- Expand that archive on your harddisc.
- Open terminal window
- change directory to the php directory. e.g. "cd /php-5.2.6"
- execute this two lines to define the supported CPU types and the minimum Mac OS X version:
- export CFLAGS="-arch ppc -arch i386 -mmacosx-version-min=10.3"
- export CXXFLAGS="-arch ppc -arch i386 -mmacosx-version-min=10.3"
- the command "./configure help" does show the configure options.
- use configure with a line like this:
- ./configure --enable-embed --with-curl --enable-ftp --enable-zip --enable-sockets --enable-static --enable-soap --with-zlib --with-bz2 --enable-exif --enable-bcmath --enable-calendar
- start the compilation with "make all"
- other option is to use "make install" which first does the same as "make all" and than does some installation scripts.
- you may get an error about a duplicate symbole __yytext. Search the file "zend_ini_scanner.c", search a line with "char *yytext;" and change it to "extern char *yytext;".
- On the end you get a lot of error messages, but you have a working library (named libphp5.so) file in the invisible ".libs" folder inside your php source folder.

Possible problems and solutions:

- If the path to your files has spaces, you can get into trouble. e.g. "/RB Plugins/PHP" is bad as files will be searched sometimes in "/RB".

- If you have in /usr/local/lib libraries which conflict with the default libraries, you can get into trouble.
- If you installed some open source tools which compiled their own libraries, you can get into conflicts.
- if you have to reconfigure or after a problem, you may need to use "make clean" before you start "make all" again.

Feel free to install additional libraries and add more packages to the configure line.

20.0.78 How to convert a BrowserType to a String with WebSession.Browser?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetBrowserName(s as WebSession.BrowserType) As string
Select case s
case WebSession.BrowserType.Android
Return "Andriod"
case WebSession.BrowserType.Blackberry
Return "Blackberry"
case WebSession.BrowserType.Chrome
Return "Chrome"
case WebSession.BrowserType.ChromeOS
Return "ChromeOS"
case WebSession.BrowserType.Firefox
Return "Firefox"
case WebSession.BrowserType.InternetExplorer
Return "InternetExplorer"
case WebSession.BrowserType.Opera
Return "Opera"
case WebSession.BrowserType.Safari
Return "Safari"
case WebSession.BrowserType.SafariMobile
Return "SafariMobile"
case WebSession.BrowserType.Unknown
Return "Unknown"
else
Return "Unkown: "+str(integer(s))
end Select

End Function
```

20.0.79 How to convert a EngineType to a String with WebSession.Engine?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetRenderingEngineName(s as WebSession.EngineType) As string
Select case s
case WebSession.EngineType.Gecko
Return "Gecko"
case WebSession.EngineType.Presto
Return "Presto"
case WebSession.EngineType.Trident
Return "Trident"
case WebSession.EngineType.Unknown
Return "Unknown"
case WebSession.EngineType.WebKit
Return "WebKit"
else
Return "Unkown: "+str(integer(s))
end Select

End Function
```

20.0.80 How to convert a PlatformType to a String with WebSession.Platform?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this:

Example:

```
Function GetPlatformName(s as WebSession.PlatformType) As string
Select case s
case WebSession.PlatformType.Blackberry
Return "Blackberry"
case WebSession.PlatformType.iPad
Return "iPad"
case WebSession.PlatformType.iPhone
Return "iPhone"
case WebSession.PlatformType.iPodTouch
Return "iPodTouch"
case WebSession.PlatformType.Linux
Return "Linux"
case WebSession.PlatformType.Macintosh
Return "Macintosh"
```

```

case WebSession.PlatformType.PS3
Return "PS3"
case WebSession.PlatformType.Unknown
Return "Unknown"
case WebSession.PlatformType.WebOS
Return "WebOS"
case WebSession.PlatformType.Wii
Return "Wii"
case WebSession.PlatformType.Windows
Return "Windows"
else
Return "Unkown: "+str(integer(s))
end Select

```

End Function

20.0.81 How to convert a text to iso-8859-1 using the TextEncoder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

This code can help you although it's not perfect.
You need to set lc to the current color you use.

Example:

```

dim outstring as string
dim theMac, thePC as textencoding
dim Mac2PC as textconverter

theMac = getTextEncoding(0) // MacRoman
thePC = getTextEncoding(&h0201) // ISOLatin1

Mac2PC = getTextConverter(theMac, thePC)
// if you wanted to do the opposite just create a converter
// PC2Mac = getTextConverter(thePC, theMac)

outstring = Mac2PC.convert("Bj√rn, this text should be converted")
Mac2PC.clear

```

Notes:

You have to call Mac2PC.clear after every conversion to reset the encoding engine.
See also newer TextConverterMBS class.

20.0.82 How to convert ChartTime back to Xojo date?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have this example code:

Example:

```
Function ChartTimeToDate(ChartTime as Double) As date
static diff as Double = 0.0
```

```
if diff = 0.0 then
dim d2 as Double = CDBaseChartMBS.chartTime(2015, 1, 1)
dim da as new date(2015, 1, 1)
dim ts as Double = da.TotalSeconds
```

```
diff = ts - d2
end if
```

```
dim d as new date
d.TotalSeconds = diff + ChartTime
```

```
Return d
End Function
```

Notes: As you see we calculate the difference in base date from Date and ChartTime and later use difference to convert.

20.0.83 How to convert line endings in text files?

Plugin Version: all, Platform: macOS.

Answer: You can simply read file with TextInputStream and write with new line endings using TextOutputStream class.

Example:

```
dim inputfile as FolderItem = SpecialFolder.Desktop.Child("test.txt")
dim outputfile as FolderItem = SpecialFolder.Desktop.Child("output.txt")
dim it as TextInputStream = TextInputStream.Open(inputfile)
dim ot as TextOutputStream = TextOutputStream.Create(outputfile)
```

```
ot.Delimiter = EndOfLine.Windows // new line ending
while not it.EOF
ot.WriteLine it.ReadLine
wend
```

Notes: `TextInputStream` will read any input line endings and with `delimiter` property in `TextOutputStream` you can easily define your new delimiter.

20.0.84 How to convert picture to string and back?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use this plugin functions:

Notes: JPEG:

`JPEGStringToPictureMBS(buf as string)` as picture
`JPEGStringToPictureMBS(buf as string,allowdamaged as Boolean)` as picture
`PictureToJPEGStringMBS(pic as picture,quality as Integer)` as string

PNG:

`PictureToPNGStringMBS(pic as picture, gamma as single)` as string
`PictureToPNGStringMBS(pic as picture, mask as picture, gamma as single)` as string
`PictureToPNGStringMBS(pic as picture, gamma as single, Interlace as Boolean, FilterType as Integer)` as string
`PictureToPNGStringMBS(pic as picture, mask as picture, gamma as single, Interlace as Boolean, FilterType as Integer)` as string
`PNGStringToPictureMBS(data as string, gamma as single)` as picture
`PNGStringToPNGPictureMBS(data as string, gamma as single)` as PNGpictureMBS

Tiff:

`TIFFStringToPictureMBS(data as string)` as picture
`TIFFStringToTiffPictureMBS(data as string)` as TiffPictureMBS

BMP:

`BMPStringtoPictureMBS(data as string)` as picture
`Picture.BMPDataMBS(ResolutionValueDPI as Integer=72)` as string

GIF:

`GifStringToGifMBS(data as string)` as GIFMBS
`GifStringToPictureMBS(data as string)` as Picture

20.0.85 How to copy an array?

Plugin Version: all, Platform: macOS.

Answer: You can use a function like this to copy an array:

Example:

```
Function CopyArray(a() as Double) as Double()
dim r() as Double
for each v as Double in a
r.Append v
next
Return r
End Function
```

Notes: If needed make several copies of this method with different data types, not just double.
For a deep copy of an array of objects, you need to change code to also make a copy of those objects.

20.0.86 How to copy an dictionary?

Plugin Version: all, Platform: macOS.

Answer: You can use a function like this to copy a dictionary:

Example:

```
Function CopyDictionary(d as Dictionary) As Dictionary
dim r as new Dictionary
for each key as Variant in d.keys
r.Value(key) = d.Value(key)
next
Return r
End Function
```

Notes: If needed make several copies of this method with different data types, not just double.
For a deep copy of an dictionary of objects, you need to change code to also make a copy of those objects.

20.0.87 How to copy parts of a movie to another one?

Plugin Version: all, Platforms: macOS, Windows.

Answer: The code below copies ten seconds of the snowman movie to the dummy movie starting at the 5th second.

Example:

```

dim f as FolderItem
dim md as EditableMovie
dim ms as EditableMovie

f=SpecialFolder.Desktop.Child("Our First Snowman.mov")
ms=f.OpenEditableMovie

ms.SelectionStartMBS=5
ms.SelectionLengthMBS=10

f=SpecialFolder.Desktop.Child("dummy.mov")
md=f.CreateMovie

msgbox str(md.AddMovieSelectionMBS(ms))

```

Notes: If result is not 0, the method fails.

20.0.88 How to create a birthday like calendar event?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

// start a connection to the calendar database
dim s as new CalCalendarStoreMBS

// needed for the error details
dim e as NSErrorMBS

dim r as CalRecurrenceRuleMBS = CalRecurrenceRuleMBS.initYearlyRecurrence(1, nil) // repeat every
year without end

dim a as new CalAlarmMBS // add alarm
a.action = a.CalAlarmActionDisplay
a.relativeTrigger = -3600*24 // 24 Hours before

// create a new calendar
dim c as new CalEventMBS

dim d as new date(2011, 04, 20) // the date

dim calendars() as CalCalendarMBS = s.calendars

```

```

// set properties
c.Title="Test Birthday"
c.startDate=d
c.recurrenceRule = r
c.calendar=calendars(0) // add to first calendar
c.addAlarm(a)
c.endDate = d
c.isAllDay = true

// save event
call s.saveEvent(c,s.CalSpanAllEvents, e)
if e<>nil then
MsgBox e.localizedDescription
else
MsgBox "New event was created."
end if

```

Notes: This adds an event to iCal for the given date with alarm to remember you and repeats it every year.

20.0.89 How to create a GUID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the UUIDMBS class for this.

20.0.90 How to create a Mac picture clip file?

Plugin Version: all, Platform: Windows.

Answer: You can use code like this one.

Example:

```

dim f As FolderItem
dim p As Picture

f=SpecialFolder.Desktop.Child("Test.pictClipping")
if f=nil then Return

p=new Picture(300,200,32) 'Make a sample picture
p.Graphics.ForeColor=RGB(0,255,255)
p.Graphics.FillOval 0,0,99,99

```

```
p.Graphics.ForeColor=RGB(255,0,0)
p.Graphics.DrawOval 0,0,99,99
```

```
dim r As ResourceFork 'ResourceFork is needed for a clip file
```

```
// Please define a file type Any
r=f.CreateResourceFork("Any")
```

```
// get PICT data using plugin function
dim pictdata as string = p.PicHandleDataMBS
r.AddResource(pictdata,"PICT",256,"Picture")
```

```
dim m as new MemoryBlock(8)
```

```
m.LittleEndian = false
m.Int16Value(0) = 0
m.Int16Value(2) = 0
m.Int16Value(4) = p.Width
m.Int16Value(6) = p.Height
```

```
r.AddResource(m,"RECT",256,"")
```

```
'Values taken from a sample file and irrelevant to the problem
```

```
dim data as string = DecodeBase64("AQAAAAAAAAAAAAAAAAACAFRDRVIAAAABAAAAAAAAAAABUQ0IQAAAAA")
r.AddResource(data,"drag",128,"") 'ditto
r.Close
```

Notes: In general Apple has deprecated this, but a few application still support clippings.

20.0.91 How to create a PDF file in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Check our DynaPDF plugin and the examples.

Notes: An alternative can be to use the CoreGraphics and Cocoa functions on Mac OS X. For Windows, we can only suggest our DynaPDF plugin.

20.0.92 How to create EmailAttachment for PDF Data in memory?

Plugin Version: all, Platform: macOS.

Answer: You can use code like the one below:

Example:

Function EmailAttachmentFromPDFData(PDFData as string, filename as string) As EmailAttachment
 dim a as new EmailAttachment

```
a.data = EncodeBase64(PDFData, 76)
a.ContentEncoding = "base64"
a.MIMETYPE = "application/pdf"
a.MacType = "PDF "
a.MacCreator = "prvw"
a.Name = filename
```

Return a

End Function

Notes: Compared to sample code from Xojo documentation, we set the mime type correct for PDF. The MacType/MacCreator codes are deprecated, but you can still include them for older Mac email clients. "prvw" is the creator code for Apple's preview app.

20.0.93 How to create PDF for image files?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use DynaPDF like this:

Example:

```
Function CreatePrintPDF(jpgFiles() as folderitem, pdfFile as FolderItem, PageWidth as Integer, PageHeight as Integer) As Boolean
  // have files?
  If pdfFile = Nil Then Return False
  If jpgFiles = Nil Then Return False

  If jpgFiles.Ubound < 0 Then Return False

  // new DynaPDF
  Dim pdf As New MyDynapdfMBS

  // page width/height in MilliMeter
  Dim pdfWidth as Integer = PageWidth * 72 / 25.4
  Dim pdfHeight as Integer = PageHeight * 72 / 25.4

  // put your license here
  Call pdf.SetLicenseKey "Starter"

  // create pdf
  Call pdf.CreateNewPDF pdfFile
```

```

// set a couple of options
Call pdf.SetPageCoords(MyDynaPDFMBS.kpcTopDown)
Call pdf.SetResolution(300)
Call pdf.SetUseTransparency(False)
Call pdf.SetSaveNewImageFormat(False)
Call pdf.SetGStateFlags(MyDynaPDFMBS.kgfUseImageColorSpace, False)
Call pdf.SetJPEGQuality(100)

// set page size
Call pdf.SetBBox(MyDynaPDFMBS.kpbMediaBox, 0, 0, pdfWidth, pdfHeight)
Call pdf.SetPageWidth(pdfWidth)
Call pdf.SetPageHeight(pdfHeight)

// append pages with one image per page
For i as Integer = 0 To jpgFiles.Ubound
Call pdf.Append
Call pdf.InsertImageEx(0, 0, pdfWidth, pdfHeight, jpgFiles(i), 1)
Call pdf.EndPage
Next

// close
Call pdf.CloseFile

Return True
End Function

```

Notes: This is to join image files in paper size to a new PDF.
e.g. scans in A4 into an A4 PDF.

20.0.94 How to CURL Options translate to Plugin Calls?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Below a few tips on how to translate command line CURL calls to plugin calls.

Notes: `curl -vX PUT http://localhost:5984/appserials/78569238475/DocumentRegister.docx?rev=3-25634563456 -data-binary @DocumentRegister.docx -H "Content-Type: application/msword"`

- The option `-v` means verbose. You can use `OptionVerbose` and listen for messages in the `DebugMessage` event.
- The option `-X PUT` means we want to do a HTTP PUT Request. So set `OptionPut` to true. Also you will want to set `OptionUpload` to true as you upload data.
- We have the URL which you put into `OptionURL` property.

- The `-data-binary` option tells CURL to pass the given data. With the `@` before the data, it is interpreted as a file name, so the data is read from the given file. You'll need to open this file and pass data with the Read event as needed. (See CURLS ftp file upload example project)
- The last option `-H` specifies an additional header for the upload. Pass this additional header with the `SetOptionHTTPHeader` method.

```
curl -X PUT http://127.0.0.1:5984/appserials/f2f4e540bf8bb60f61cfc4328001c59 -d '{ "type": "Product", "description": "Application Serial", "acronym": "AppSerial", "dateAdded": "2011-03-21 14:57:36" } '
```

- Option `-X PUT` like above.
- Pass the URL again in `OptionURL`
- This time data is passed in command line for CURL. You'd put this data in the quotes into a string and make it available in the Read event. (See CURLS ftp upload example project)

20.0.95 How to delete file with ftp and curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can set post/pre quotes to have ftp commands executed before or after the download/upload.

Example:

```
dim d as CURLMBS // your curl object
```

```
// delete file
```

```
dim ws() As String
```

```
ws.Append "DELE Temp.txt"
```

```
d.SetOptionPostQuote(ws)
```

Notes: Use `SetOptionPostQuote`, `SetOptionPreQuote` or `SetOptionQuote`.

The ftp commands you pass here are native ftp commands and not the commands you use with ftp applications. To delete use `DELE` and the file path.

20.0.96 How to detect display resolution changed?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac OS X simply listen for display changed notifications.

Notes: Use the "Distribution Notification Center.rbp" example project as a base and use it to listen to notifications with the name "O3DeviceChanged".

20.0.97 How to detect retina?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use `Window.BackingScaleFactorMBS` to query the factor.

Example:

```
msgbox str(window1.BackingScaleFactorMBS)
```

20.0.98 How to disable force quit?

Plugin Version: all, Platform: macOS.

Answer:

Please visit this website and get the control panel for Mac OS 9 there:

<http://www3.sk.sympatico.ca/tinyjohn/DFQ.html>

For Mac OS X use the MBS Plugin with the `SetSystemUIModeMBS` method.

Notes:

Please use `presentationOptions` in `NSApplicationMBS` for Cocoa applications.

20.0.99 How to disable the error dialogs from Internet Explorer on javascript errors?

Plugin Version: all, Platform: Windows.

Answer: You can use this code in the `htmlviewer` open event:

Example:

```
if targetwin32 then
htmlviewer1._ole.Content.value("Silent") = True
end if
```

Notes: This disables the error dialogs from Internet Explorer.

20.0.100 How to display a PDF file in Xojo?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac OS X you can use CoreGraphics or PDFKit to display a PDF.

Notes: An alternative can be to load the PDF into a htmlviewer so the PDF plugin can display it. On Windows you may need to use the Acrobat ActiveX control from Adobe or launch Acrobat Reader.

20.0.101 How to do a lottery in RB?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this function:

Example:

```
Sub Lotto(max as Integer,count as Integer,z() as Integer)
// Lotto count numbers of max put into the array z beginning at index 0
dim n(0) as Integer ' all the numbers
dim m as Integer ' the highest field in the current array
dim i,a,b,d as Integer ' working variables

'fill the array with the numbers
m=max-1
redim n(m)

for i=0 to m
n(i)=i+1
next

' unsort them by exchanging random ones
m=max*10
for i=1 to m
a=rnd*max
b=rnd*max

d=n(a)
n(a)=n(b)
n(b)=d
next

' get the first count to the dest array
m=count-1
redim z(m)
for i=0 to m
z(i)=n(i)
next

'sort the result
z.sort
End Sub
```

```

Sub Open()
// Test it

dim za(0) as Integer ' the array of the numbers

lotto 49,6,za ' 6 of 49 in Germany

' and display them
staticText1.text=str(za(0))+chr(13)+str(za(1))+chr(13)+str(za(2))+chr(13)+str(za(3))+chr(13)+str(za(4))+chr(13)+str(za(5))+chr(13)+str(za(6))+chr(13)+str(za(7))+chr(13)+str(za(8))+chr(13)+str(za(9))+chr(13)+str(za(10))+chr(13)+str(za(11))+chr(13)+str(za(12))+chr(13)+str(za(13))+chr(13)+str(za(14))+chr(13)+str(za(15))+chr(13)+str(za(16))+chr(13)+str(za(17))+chr(13)+str(za(18))+chr(13)+str(za(19))+chr(13)+str(za(20))+chr(13)+str(za(21))+chr(13)+str(za(22))+chr(13)+str(za(23))+chr(13)+str(za(24))+chr(13)+str(za(25))+chr(13)+str(za(26))+chr(13)+str(za(27))+chr(13)+str(za(28))+chr(13)+str(za(29))+chr(13)+str(za(30))+chr(13)+str(za(31))+chr(13)+str(za(32))+chr(13)+str(za(33))+chr(13)+str(za(34))+chr(13)+str(za(35))+chr(13)+str(za(36))+chr(13)+str(za(37))+chr(13)+str(za(38))+chr(13)+str(za(39))+chr(13)+str(za(40))+chr(13)+str(za(41))+chr(13)+str(za(42))+chr(13)+str(za(43))+chr(13)+str(za(44))+chr(13)+str(za(45))+chr(13)+str(za(46))+chr(13)+str(za(47))+chr(13)+str(za(48))+chr(13)+str(za(49))+chr(13)+str(za(50))+chr(13)+str(za(51))+chr(13)+str(za(52))+chr(13)+str(za(53))+chr(13)+str(za(54))+chr(13)+str(za(55))+chr(13)+str(za(56))+chr(13)+str(za(57))+chr(13)+str(za(58))+chr(13)+str(za(59))+chr(13)+str(za(60))+chr(13)+str(za(61))+chr(13)+str(za(62))+chr(13)+str(za(63))+chr(13)+str(za(64))+chr(13)+str(za(65))+chr(13)+str(za(66))+chr(13)+str(za(67))+chr(13)+str(za(68))+chr(13)+str(za(69))+chr(13)+str(za(70))+chr(13)+str(za(71))+chr(13)+str(za(72))+chr(13)+str(za(73))+chr(13)+str(za(74))+chr(13)+str(za(75))+chr(13)+str(za(76))+chr(13)+str(za(77))+chr(13)+str(za(78))+chr(13)+str(za(79))+chr(13)+str(za(80))+chr(13)+str(za(81))+chr(13)+str(za(82))+chr(13)+str(za(83))+chr(13)+str(za(84))+chr(13)+str(za(85))+chr(13)+str(za(86))+chr(13)+str(za(87))+chr(13)+str(za(88))+chr(13)+str(za(89))+chr(13)+str(za(90))+chr(13)+str(za(91))+chr(13)+str(za(92))+chr(13)+str(za(93))+chr(13)+str(za(94))+chr(13)+str(za(95))+chr(13)+str(za(96))+chr(13)+str(za(97))+chr(13)+str(za(98))+chr(13)+str(za(99))
End Sub

```

20.0.102 How to do an asycron DNS lookup?

Plugin Version: all, Platform: Windows.

Answer: use CFHostMBS class (Mac OS X only).

Notes: Xojo internal functions and plugin DNS functions are sycronized.

You can use DNSLookupThreadMBS class for doing them asycron.

20.0.103 How to draw a dushed pattern line?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can try this code:

Example:

```
// call like this: DrawDushedPatternLine g,0,0,width,height,10
```

```

Sub DrawDushedPatternLine(g as graphics,x1 as Integer,y1 as Integer,x2 as Integer,y2 as Integer, partlen
as Integer)
dim x,y,ox,oy as Double
dim dx,dy as Double
dim w,h,d as Double
dim b as Boolean

w=x2-x1
h=y2-y1

d=sqrt(w*w+h*h)

dx=w/d*partlen
dy=h/d*partlen

```

```

b=true
x=x1
while (x<x2) and (y<y2)
  ox=x
  oy=y

  x=x+dx
  y=y+dy

  if b then
    g.DrawLine ox,oy,x,y
  end if

  b=not b
wend

```

End Sub

Notes: It would be possible to add this to the plugin, but I think it's better if you do it in plain Xojo code, so it even works on Windows.

20.0.104 How to draw a nice antialiased line?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

This code can help you although it's not perfect.
You need to set lc to the current color you use.

Example:

```

Sub drawLine(xs as Integer, ys as Integer, xe as Integer, ye as Integer, face as RGBSurface, lineColor as
color)
  dim intX, intY, count, n, xDiff, yDiff as Integer
  dim v, v1, floatX, floatY, xx, yy, xStep, yStep as Double
  dim c as color

  const st=1.0

  xDiff=xe-xs
  yDiff=ye-ys
  count=max(abs(xDiff), abs(yDiff))
  xStep=xDiff/count
  yStep=yDiff/count

```

```

xx=xs
yy=ys
for n=1 to count
intX=xx
intY=yy
floatX=xx-intX
floatY=yy-intY

v=(1-floatX)*(1-floatY)*st
v1=1-v
c=face.pixel(intX, intY)
face.pixel(intX, intY)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=floatX*(1-floatY)*st
v1=1-v
c=face.pixel(intX+1, intY)
face.pixel(intX+1, intY)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=(1-floatX)*floatY*st
v1=1-v
c=face.pixel(intX, intY+1)
face.pixel(intX, intY+1)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)
v=floatX*floatY*st
v1=1-v
c=face.pixel(intX+1, intY+1)
face.pixel(intX+1, intY+1)=rgb(v*lineColor.red+v1*c.red, v*lineColor.green+v1*c.green, v*lineColor.blue+v1*c.blue)

xx=xx+xStep
yy=yy+yStep
next

End Sub

```

Notes:

PS: st should be 1 and face should be a RGBSurface or a Graphics object.

20.0.105 How to dump java class interface?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In terminal you can use "javap -s <classname>" to display the class with the method names and parameters.

Notes: For example show ResultSet class: javap -s java.sql.ResultSet

20.0.106 How to duplicate a picture with mask or alpha channel?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this function:

Example:

```
Function Duplicate(extends p as Picture) As Picture
#if RBVersion >= 2011.04 then
if p.HasAlphaChannel then

// create nw picture and copy content:
dim q as new Picture(p.Width, p.Height)
q.Graphics.DrawPicture p,0,0

Return q

end if
#endif

// create new picture
dim q as new Picture(p.Width, p.Height, 32)

// get mask
dim oldMask as Picture = p.mask(false)
if oldMask = nil then
// no mask, so simple copy
q.Graphics.DrawPicture p,0,0
Return q
end if

// remove mask
p.mask = nil

// copy picture and mask
q.Graphics.DrawPicture p, 0, 0
q.mask.Graphics.DrawPicture oldMask,0,0

// restore mask
p.mask = oldmask

Return q
End Function
```

Notes: Simply copy it to a module and call it like this: `q = p.duplicate`.

The code above works with old Xojo versions because of the `#if` even if your RS version does not support alpha channel pictures. This way it's future proof.

20.0.107 How to enable assistive devices?

Plugin Version: all, Platform: macOS.

Answer: You can use AppleScript code like below:

Notes: tell application "System Events"
activate

```
set UI elements enabled to true
```

```
return UI elements enabled
end tell
```

You can run this with AppleScriptMBS class.

20.0.108 How to encrypt a file with Blowfish?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
dim fi as FolderItem = SpecialFolder.Desktop.Child("test.xojo_binary_project")
dim fo as FolderItem = SpecialFolder.Desktop.Child("test.encrypted")
```

```
// read input
```

```
dim bi as BinaryStream = BinaryStream.Open(fi)
```

```
dim si as string = bi.Read(bi.Length)
```

```
bi.Close
```

```
// encrypt
```

```
dim so as string = BlowfishMBS.Encrypt("MyKey",si)
```

```
// write output
```

```
dim bo as BinaryStream = BinaryStream.Create(fo)
```

```
bo.Write so
```

```
bo.Close
```

Notes: Of course you can decrypt same way, just use Decrypt function and of course swap files.

20.0.109 How to extract text from HTML?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use both RemoveHTMLTagsMBS and DecodingFromHTMLMBS like this:

Example:

```
dim html as string = "<p><B>Gr&uuml;&szlig;e</B></P>"
dim htmltext as string = RemoveHTMLTagsMBS(html)
dim text as string = DecodingFromHTMLMBS(htmltext)
```

MsgBox text // shows: Gr√üë

Notes: You can use it together with RemoveHTMLTagsMBS to remove html tags. What you get will be the text without tags.

DecodingFromHTMLMBS turns HTML escapes back to unicode characters. Like ä to √.

20.0.110 How to find empty folders in a folder?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Try this code:

Example:

```
dim folder as folderitem // your folder

dim c as Integer = folder.count
for i as Integer = 1 to c
dim item as folderitem = folder.trueitem(i)
if item = nil then
// ignore
elseif item.directory then
// folder
if item.count = 0 then
// found empty folder
end if
end if
next
```

20.0.111 How to find iTunes on a Mac OS X machine fast?

Plugin Version: all, Platform: macOS.

Answer: You can try Launch Services.

Example:

```
dim f as FolderItem

f=LaunchServicesFindApplicationForInfoMBS("hook","com.apple.iTunes","iTunes.app")

MsgBox f.NativePath
```

20.0.112 How to find network interface for a socket by it's name?

Plugin Version: all, Platform: macOS.

Answer: You can use our plugin to build a lookup table.

Example:

```
Function FindNetworkInterface(name as string) As NetworkInterface
name = name.trim

if name.len = 0 then Return nil

// search by IP/MAC
dim u as Integer = System.NetworkInterfaceCount-1
for i as Integer = 0 to u
dim n as NetworkInterface = System.GetNetworkInterface(i)
if n.IPAddress = name or n.MACAddress = name then
Return n
end if
next

// use MBS Plugin to build a mapping
dim interfaces() as NetworkInterfaceMBS = NetworkInterfaceMBS.AllInterfaces
dim map as new Dictionary

for each n as NetworkInterfaceMBS in interfaces
dim IPv4s() as string = n.IPv4s
dim IPv6s() as string = n.IPv6s

for each IPv4 as string in IPv4s
map.Value(IPv4) = n.Name
next
for each IPv6 as string in IPv6s
map.Value(IPv6) = n.Name
next
if n.MAC<>>" then
map.Value(n.MAC) = n.Name
```

```

end if
next

// now search interfaces by name, IPv4 or IPv6
for i as Integer = 0 to u
dim n as NetworkInterface = System.GetNetworkInterface(i)
if map.Lookup(n.IPAddress, "") = name then
Return n
end if

if map.Lookup(n.MACAddress, "") = name then
Return n
end if
next

End Function

```

Notes: The code above uses a lookup table build using NetworkInterfaceMBS class to find the network interface by name.

20.0.113 How to find version of Microsoft Word?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```

// find Word
dim f as FolderItem = LaunchServicesFindApplicationForInfoMBS("", "com.microsoft.Word", "")

// open bundle
dim c as new NSBundleMBS(f)

// read info
dim d as Dictionary = c.infoDictionary

// show version
MsgBox d.Lookup("CFBundleVersion", "")

```

Notes: Older versions of Word can be found with creator code "MSWD".

20.0.114 How to fix CURL error 60/53 on connecting to server?

Plugin Version: all, Platform: macOS.

Answer: You probably connect with SSL and you have no valid certificate.

Example:

```
dim d as new CURLSMBS

// Disable SSL verification
d.OptionSSLVerifyHost = 0 // don't verify server
d.OptionSSLVerifyPeer = 0 // don't proofs certificate is authentic

// With SSL Verification:
dim cacert as FolderItem = Getfolderitem("cacert.pem")
d.OptionCAInfo = cacert.NativePath
d.OptionSSLVerifyHost = 2 // verify server
d.OptionSSLVerifyPeer = 1 // proofs certificate is authentic
```

Notes: You can either use the code above to disable the SSL verification and have no security. Or you use the cacert file and enable the verification. Than you only get a connection if the server has a valid certificate.

see also:

<http://curl.haxx.se/ca/>

20.0.115 How to format double with n digits?

Plugin Version: all, Platform: macOS.

Answer: You can use the FormatMBS function for this.

Example:

```
dim d as Double = 123.4567890
listbox1.AddRow FormatMBS("%f", d)
listbox1.AddRow FormatMBS("%e", d)
listbox1.AddRow FormatMBS("%g", d)

listbox1.AddRow FormatMBS("%5.5f", d)
listbox1.AddRow FormatMBS("%5.5e", d)
listbox1.AddRow FormatMBS("%5.5g", d)

d = 0.000000123456
listbox1.AddRow FormatMBS("%f", d)
listbox1.AddRow FormatMBS("%e", d)
```

```
listbox1.AddRow FormatMBS("%g", d)

listbox1.AddRow FormatMBS("%5.5f", d)
listbox1.AddRow FormatMBS("%5.5e", d)
listbox1.AddRow FormatMBS("%5.5g", d)
```

Notes: see FormatMBS for details.

In general %f is normal style, %e is scientific and %g is whichever gives best result for given space.

20.0.116 How to get a time converted to user time zone in a web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the WebSession.GMTOffset property.

Example:

```
Sub Open()
// current date on server
dim d as new date
dim s as string = d.LongTime

// adjust to client GMT offset
d.GMTOffset = d.GMTOffset + Session.GMTOffset

dim t as string = D.LongTime

MsgBox s+EndOfLine+t
End Sub
```

20.0.117 How to get an handle to the frontmost window on Windows?

Plugin Version: all, Platform: Windows.

Answer: This function returns a handle for the frontmost window:

Example:

```
Function GetForegroundWindowHandle() as Integer
#if targetwin32 then
declare function GetForegroundWindow Lib "user32.dll" as Integer
Return GetForegroundWindow()
#endif
End Function
```

20.0.118 How to get CFAbsoluteTime from date?

Plugin Version: all, Platforms: macOS, Windows.

Answer: Use code like this:

Example:

```
dim d as new date
dim t as CFTimeZoneMBS = SystemCFTimeZoneMBS
dim g as new CFGregorianCalendarMBS
g.Day = d.Day
g.Month = d.Month
g.Year = d.Year
g.Minute = d.Minute
g.Hour = d.Hour
g.Second = d.Second

dim at as CFAbsoluteTimeMBS = g.AbsoluteTime(t)
dim x as Double = at.Value
```

```
MsgBox str(x)
```

Notes: As you see we need a timezone and put the date values in a gregorian date record. Now we can query absolute time for the given timezone.

20.0.119 How to get client IP address on web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the WebSession.RemoteAddress property.

Example:

```
Sub Open()
Title = Session.RemoteAddress
End Sub
```

20.0.120 How to get fonts to load in charts on Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use the `SetFontSearchPath` method in the `CDBaseChartMBS` class to specify where your fonts are.

Example:

```

if TargetLinux then
CDBaseChartMBS.SetFontSearchPath "/usr/share/fonts/truetype;/usr/share/fonts/truetype/msttcorefonts"
else
// on Mac and Windows we use system fonts.
end if

// also you can later switch default fonts:

dim Chart as CDBaseChartMBS // your chart

#If TargetARM And TargetLinux Then
// use specific fonts on Linux on Raspberry Pi
Call Chart.setDefaultFonts("/usr/share/fonts/truetype/piboto/PibotoLt-Regular.ttf", "/usr/share/fonts/truetype/piboto/Pi
#EndIf

```

Notes: On macOS, iOS and Windows, the fonts are loaded from the system's font folder.

e.g. if you use ubuntu, you can install the `ttf-mscorefonts-installer` package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

20.0.121 How to get fonts to load in DynaPDF on Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use the `AddFontSearchPath` method in the `DynaPDFMBS` class to specify where your fonts are.

Example:

```

dim d as new DynaPDFMBS
if TargetLinux then
call d.AddFontSearchPath "/usr/share/fonts/truetype", true
else
// on Mac and Windows we use system fonts.
end if

```

Notes: On Mac OS X and Windows, the fonts are loaded from the system's font folder.

e.g. if you use ubuntu, you can install the `ttf-mscorefonts-installer` package and call this method with `"/usr/share/fonts/truetype/msttcorefonts"` as the path. No backslash on the end of a path, please.

20.0.122 How to get GMT time and back?

Plugin Version: all, Platform: macOS.

Answer: You can use the date class and the GMTOffset property.

Example:

```
// now
dim d as new date

// now in GMT
dim e as new date
e.GMTOffset = 0

// show
MsgBox str(d.TotalSeconds,"0.0")+ " " +str(e.TotalSeconds, "0.0")

dim GMTTimeStamp as Double = e.TotalSeconds

// restore
dim f as new date

// add GMT offset here
f.TotalSeconds = GMTTimeStamp + f.GMTOffset*3600
// because here it's removed
f.GMTOffset = f.GMTOffset

MsgBox d.ShortTime+" (" +str(d.GMTOffset)+") " +str(d.TotalSeconds,"0.0")+EndOfLine+_
e.ShortTime+" (" +str(e.GMTOffset)+") " +str(e.TotalSeconds,"0.0")+EndOfLine+_
f.ShortTime+" (" +str(f.GMTOffset)+") " +str(f.TotalSeconds,"0.0")
```

Notes: It's sometimes a bit tricky with the date class as setting one property often changes the others.

20.0.123 How to get good crash reports?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Check this website from the webkit website:

Notes: <http://webkit.org/quality/crashlogs.html>

20.0.124 How to get list of all threads?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the runtime module like in this function:

Example:

```
Function Threads() As Thread()
#pragma DisableBackgroundTasks
dim t() as Thread

Dim o as Runtime.ObjectIterator=Runtime.IterateObjects
While o.MoveNext
if o.Current isa Thread then
t.Append thread(o.current)
end if
Wend

Return t
End Function
```

Notes: This returns an array of all thread objects currently in memory.

The pragma is important here as it avoids thread switches which may cause a thread to be created or deleted.

20.0.125 How to get parameters from webpage URL in Xojo Web Edition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the Webpage.ParametersReceived event.

Example:

```
Sub ParametersReceived(Variables As Dictionary)
for each key as Variant in Variables.keys
MsgBox key+" ->" +Variables.Value(key)
next
End Sub
```

Notes: The text encodings of this strings is not defined in Xojo 2010r5. Please use DefineEncoding.

20.0.126 How to get the color for disabled textcolor?

Plugin Version: all, Platform: macOS.

Answer: Ask the appearance manager:

Example:

```
Function GetThemeTextColor(inColor as Integer, inDepth as Integer, inColorDev as Boolean) As Color
declare function GetThemeTextColor lib "Carbon" (inColor as Integer, inDepth as Integer, inColorDev as
Boolean, outColor as Ptr) as Integer
```

```
dim i as Integer
```

```
dim col as MemoryBlock
```

```
col = newMemoryBlock(6)
```

```
i = GetThemeTextColor(inColor, inDepth, inColorDev, col)
```

```
return RGB(col.UShort(0)\256, col.UShort(2)\256, col.UShort(4)\256)
```

```
End Function
```

Notes: The color for this is:

```
const kThemeTextColorDialogInactive = 2.
```

```
c = GetThemeTextColor(kThemeTextColorDialogInactive, Screen(0).Depth, true)
```

For Mac OS X you should use "CarbonLib" instead of "AppearanceLib" ...

20.0.127 How to get the current free stack space?

Plugin Version: all, Platform: macOS.

Answer: You can something like the code below:

Example:

```
Sub ShowStackSize()
```

```
dim threadid as Integer
```

```
dim size as Integer
```

```
declare function GetCurrentThread lib "Carbon" (byref threadid as Integer) as short
```

```
declare function ThreadCurrentStackSize lib "Carbon" (threadid as Integer, byref size as Integer) as short
```

```
if GetCurrentThread(threadid)=0 then
```

```
if 0=ThreadCurrentStackSize(threadid,size) then
```

```
MsgBox str(size)
```

```
end if
```

```
end if
```

End Sub

Notes: For Mac OS 9, use "ThreadLib" instead of "CarbonLib". You can use #if if you like for that.

20.0.128 How to get the current timezone?

Plugin Version: all, Platforms: macOS, Windows.

Answer:

You can use the TimeZoneMBS class or the CFTimeZoneMBS class.
Or code like below:

Example:

```
Function GMTOffsetInMinutes() as Integer
// Returns the offset of the current time to GMT in minutes.
// supports Mac OS and Windows, but not Linux yet (let me know if
// you have code for that, please)
//
// Note that the offset is not always an even multiple of 60, but
// there are also half hour offsets, even one 5:45h offset

// This version by Thomas Tempelmann (rb@tempel.org) on 25 Nov 2005
// with a fix that should also make it work with future Intel Mac targets.
//
// Using code from various authors found on the RB NUG mailing list

dim result, bias, dayLightbias as Integer
dim info as memoryBlock
dim offset as Integer

#if targetMacOS then

Declare Sub ReadLocation lib "Carbon" (location As ptr)

info = NewMemoryBlock(12)
ReadLocation info
if false then
// bad, because it does not work on Intel Macs:
'offset = info.short(9) * 256 + info.byte(11)
else
offset = BitwiseAnd (info.long(8), &hFFFFFF)
end

offset = info.short(9) * 256 + info.byte(11)
```

```

offset = offset \60
return offset

#endif

#if targetWin32 then

Declare Function GetTimeZoneInformation Lib "Kernel32" ( tzInfoPointer as Ptr ) as Integer
// returns one of
// TIME_ZONE_ID_UNKNOWN 0
// - Note: e.g. New Delhi (GMT+5:30) and Newfoundland (-3:30) return this value 0
// TIME_ZONE_ID_STANDARD 1
// TIME_ZONE_ID_DAYLIGHT 2

info = new MemoryBlock(172)
result = GetTimeZoneInformation(info)

bias = info.Long(0)
// note: the original code I found in the NUG archives used Long(84) and switched to Long(0)
// only for result=1 and result=2, but my tests found that Long(0) is also the right value for result=0

if result = 2 then
daylightBias = info.long(168)
end if
offset = - (bias + dayLightbias)
return offset

#endif

End Function

```

20.0.129 How to get the current window title?

Plugin Version: all, Platform: macOS.

Answer: The code below returns the current window title for the frontmost window on Mac OS X if Accessibility services are

Example:

```

Function CurrentWindowTitle() As string
// your application needs permissions for accessibility to make this work!

dim SystemWideElement,FocusedApplicationElement,FocusedWindowElement as AXUIElementMBS
dim FocusedApplication,FocusedWindow,Title as AXValueMBS
dim s as String
dim cs as CFStringMBS

```

```

SystemWideElement=AccessibilityMBS.SystemWideAXUIElement
if SystemWideElement<>nil then
FocusedApplication=SystemWideElement.AttributeValue(AccessibilityMBS.kAXFocusedApplicationAttribute)
if FocusedApplication.Type=AccessibilityMBS.kAXUIElementMBSTypeID then
FocusedApplicationElement=new AXUIElementMBS
FocusedApplicationElement.Handle=FocusedApplication.Handle
FocusedApplicationElement.RetainObject

FocusedWindow=FocusedApplicationElement.AttributeValue(AccessibilityMBS.kAXFocusedWindowAttribute)

if FocusedWindow<>nil and AccessibilityMBS.kAXUIElementMBSTypeID=FocusedWindow.Type then

FocusedWindowElement=new AXUIElementMBS
FocusedWindowElement.Handle=FocusedWindow.Handle
FocusedWindowElement.RetainObject

Title=FocusedWindowElement.AttributeValue(AccessibilityMBS.kAXTitleAttribute)
if Title<>nil and Title.Type=kCFStringMBSTypeID then
cs=new CFStringMBS
cs.handle=Title.Handle
cs.RetainObject
Return cs.str
end if
end if
end if
end if
End Function

```

20.0.130 How to get the cursor blink interval time?

Plugin Version: all, Platform: macOS.

Answer: On Mac OS you can use GetCaretTime from the toolbox.

Example:

```
declare function GetCaretTime lib "Carbon" () as Integer
```

```
MsgBox str(GetCaretTime()+ " ticks")
```

Notes: 60 ticks make one second.

20.0.131 How to get the list of the current selected files in the Finder?

Plugin Version: all, Platform: macOS.

Answer:

Use the AppleScript like this one:

```
tell application "finder"
return selection
end tell
```

Which translates into this AppleEvent:

```
Process("Finder").SendAE "core,getd,'—':obj { form:prop, want:type(prop), seld:type(sele), from:'null'() }
"
```

and as Xojo code it looks like this:

Example:

```
dim ae as appleEvent
dim o1 as appleEventObjectSpecifier
dim f as folderItem
dim alist as appleEventDescList
dim i as Integer
dim dateiname as string

// setup the AppleEvent
o1=getpropertyObjectDescriptor( nil, "sele")
ae= newappleEvent("core", "getd", "MACS")
ae.objectSpecifierParam("—")=o1

// send it
if ae.send then
// got the list
alist=ae.replyDescList

// now show the list of filename into an editfield:

for i=1 to alist.count
f=alist.folderItem(i)

dateiname=f.name
// editfield1 with property "multiline=true"!
editfield1.text=editfield1.text + dateiname + chr(13)
next
```

end if

20.0.132 How to get the Mac OS system version?

Plugin Version: all, Platform: macOS.

Answer: The following code queries the value and displays the version number:

Example:

```

dim first as Integer
dim second as Integer
dim third as Integer
dim l as Integer

if System.Gestalt("sysv",l) then

Third=Bitwiseand(l,15)
second=Bitwiseand(l\16,15)
first=Bitwiseand(l\256,15)+10*Bitwiseand(l\256\16,15)
end if

if First>=10 then
msgbox "Mac OS X "+str(First)+" "+str(Second)+" "+str(third)
else
msgbox "Mac OS "+str(First)+" "+str(Second)+" "+str(third)
end if

```

20.0.133 How to get the Mac OS Version using System.Gestalt?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```

Dim s As String
Dim b As Boolean
Dim i, resp as Integer

// Systemversion
b = System.Gestalt("sysv", resp)
If b then
s = Hex(resp)

```

```

For i =Len(s)-1 DownTo 1
s=Left(s,i)+””+Mid(s,i+1)
Next
MsgBox ”Systemversion: Mac OS ” + s
end if

```

Notes: The MBS Plugin has a SystemInformationMBS.OSVersionString function for this.

20.0.134 How to get the screensize excluding the task bar?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Notes: Use the Screen class with the available* properties.

20.0.135 How to get the size of the frontmost window on Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Notes: Make yourself a class for the WindowRect with four properties:

```

Bottom as Integer
Left as Integer
Right as Integer
Top as Integer

```

Add the following method to your class:

```

Sub GetWindowRect(windowhandle as Integer)
dim err as Integer
dim mem as memoryBlock
#if targetwin32 then
Declare Function GetWindowRect Lib ”user32.dll” (hwnd as Integer, ipRect As Ptr) as Integer

mem = newmemoryBlock(16)
err = GetWindowRect(windowhandle, mem)
Left = mem.long(0)
Top = mem.Long(4)
Right = mem.Long(8)
Bottom = mem.Long(12)

```

```
#endif  
End Sub
```

Good to use for the MDI Master Window!

20.0.136 How to get the source code of a HTMLViewer?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
// for Windows:
```

```
msgbox HTMLViewer1.IEHTMLTextMBS
```

```
// for MacOS with WebKit 2.x:
```

```
msgbox HTMLViewer1.WKWebViewMBS.HTMLText
```

20.0.137 How to get Xojo apps running Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You need to install some require packages.

Notes: You need CUPS as well as GTK packages. On 64 bit systems also the ia32-libs package.

Please note that you need a x86 compatible Linux. So no PPC, Power, ARM or other CPUs.

20.0.138 How to handle really huge images with GraphicsMagick or ImageMagick?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Sometimes it may be better to use an extra application to process images.

Notes: A typical 32 bit app made with Xojo can use around 1.8 GB on Windows and 3 GB on Mac OS X. Some images may be huge, so that processing them causes several copies of the image to be in memory. With a 500 MB image in memory, doing a scale or rotation may require a temp image. So with source, temp and dest images with each 500 MB plus your normal app memory usage, you may hit the limit of Windows with 1.8 GB.

In that case it may be worth running a tool like gm in the shell class. gm is the command line version of GraphicsMagick. There you can run the 64 bit version which is not limited in memory like your own application. Also you can monitor progress and keep your app responsive.

20.0.139 How to handle tab key for editable cells in listbox?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this function:

Example:

```
Function HandleTabInList(list as listbox, row as Integer, column as Integer, key as String) As Boolean
// Handle tab character in Listbox.CellKeyDown event
```

```
Select case asc(key)
case 9
if Keyboard.AsyncShiftKey then
// back

// look for column left
for i as Integer = column-1 downto 0
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next

// not found, so look in row before
row = row - 1
if row >= 0 then
for i as Integer = list.ColumnCount-1 downto 0
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
end if
else
// forward

// look for column right
for i as Integer = column+1 to list.ColumnCount-1
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
```

```

// not found, so look in row below
row = row + 1
if row <list.ListCount then
for i as Integer = 0 to list.ColumnCount-1
if list.ColumnType(i) >= list.TypeEditable then
list.EditCell(row, i)
Return true
end if
next
end if
end if
end Select
End Function

```

Notes: You call it from CellKeyDown event like this:

```

EventHandler Function CellKeyDown(row as Integer, column as Integer, key as String) As Boolean
if HandleTabInList(me, row, column, key) then Return true
End EventHandler

```

As you see in the code, we handle tab and shift + tab for moving back and forward. Also we wrap to previous/next row if needed. Feel free to extend this to wrap from last to first row or create a new row for editing.

20.0.140 How to hard link MapKit framework?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Our MapKit classes weak link the framework. If you need hard linking it for the App Store, you can add this method to a class:

Example:

```

Sub ReferenceMapKit()
// just put this in window or app class

#if TargetMachO and Target64Bit then
Declare sub testing Lib "MapKit" Selector "test" (id as ptr)
testing(nil)
#endif

End Sub

```

Notes: No need to call the method.

Just having it in a window or app, will cause the compiler to hard link the framework.

20.0.141 How to have a PDF downloaded to the user in a web application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use a WebHTMLViewer control and load the PDF file with the PDF plugin from the browser.

Example:

```
dim CurrentFile as WebFile // a property of the WebPage

// define the PDF file
CurrentFile = new WebFile
CurrentFile.Filename = "test.pdf"
CurrentFile.MIMEType = "application/pdf"
CurrentFile.Data = "some pdf data" // MyDynaPDF.GetBuffer
CurrentFile.ForceDownload = true

// start the download
showurl(CurrentFile.url)
```

Notes: See our Create PDF example for the Xojo Web Edition.

20.0.142 How to hide all applications except mine?

Platform: macOS.

Answer: The code below will on Mac OS hide all applications except your one:

Example:

```
dim p as new ProcessMBS

p.GetFirstProcess
do
if not p.FrontProcess then
p.Visible=false
end if
loop until not p.GetNextProcess
```

20.0.143 How to hide script errors in HTMLViewer on Windows?

Plugin Version: all, Platform: Windows.

Answer: Set Internet Explorer to silent mode with code like this:

Example:

```
htmlviewer1._ole.Content.value("Silent") = True
```

Notes: Simply put this code in the open event of your htmlviewer control (using me instead of htmlviewer1).

20.0.144 How to hide the grid/background/border in ChartDirector?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: If you want to hide something in a chart, simply assign the kTransparent constant as color.

20.0.145 How to hide the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub HideCursor Lib "Carbon" () Inline68K("A852")
```

```
HideCursor
```

Notes: The MBS Plugin has this function and supports it on Windows, too.

20.0.146 How to insert image to NSTextView or TextArea?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With NSTextViewMBS you can use this code to insert file:

Example:

```
// insert a file to textview
```

```
Public Sub InsertFile(textview as NSTextViewMBS, f as FolderItem)
```

```
// read to file
```

```

dim b as BinaryStream = BinaryStream.Open(f)
dim s as string = b.Read(b.Length)

// build wrapper
dim fileWrapper as NSFileWrapperMBS = NSFileWrapperMBS.initRegularFileWithContents(s)
fileWrapper.preferredFilename = f.name

// make attachment
dim fileAttachment as new NSTextAttachmentMBS(fileWrapper)
dim attributedString as NSAttributedStringMBS = NSAttributedStringMBS.attributedStringWithAttachment(fileAttachment)

// add to a NSTextViewMBS
textview.insertText attributedString

End Sub

```

Notes: For TextArea you can query the underlying NSTextViewMBS object via TextArea.NSTextViewMBS method.

20.0.147 How to jump to an anchor in a htmlviewer?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You can use javascript to change the current window's location.

Example:

```

// load website
htmlviewer1.LoadURL "http://www.monkeybreadsoftware.net/addressbook-abpersonmbs.shtml"

// later jump to anchor named "16":

if TargetWin32 then
call HTMLViewer1.IERunJavaScriptMBS "window.location = ""#16""
end if

```

20.0.148 How to keep a movieplayer unclickable?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: To keep the user away from clicking on a playing Movie you can just drop a Canvas in front of the Movieplayer and take the clicks there.

Example:

```
Function Canvas1.MouseDown(X as Integer, Y as Integer) as boolean
return true // take it and do nothing
End Function
```

20.0.149 How to keep my web app from using 100% CPU time?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Linux and MacOS you can use renice command in the terminal. On Windows use the task manager to reduce priority.

Notes: If you launch your app with nohup on Linux or Mac OS X like this from the terminal or a script:

```
nohup /webapps/MyApp/MyApp &
```

you can simply have a second line saying this:

```
renice 20 $ !
```

which tells the system to lower priority to lowest value for the latest background process.

20.0.150 How to kill a process by name?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can kill a process (or application) by name if you loop over all the processes and kill the one you need.

Example:

```
dim p as new ProcessMBS
p.GetfirstProcess ' get first
do
if p.name = "TextEdit" then
call p.KillProcess
Return
end if
loop until not p.GetNextProcess
```

Notes: You may want to check the result of killProcess function. Not every user is allowed to kill every application.

20.0.151 How to know how many CPUs are present?

Plugin Version: all, Platform: macOS.

Answer: Try this function:

Example:

```
Function GetCPUCount() as Integer
Declare Function MPProcessors Lib "Carbon" () as Integer
```

```
Return MPProcessors()
End Function
```

Notes: Your app will than need that library to launch on Classic. To avoid this the MBS plugin checks if this library is available and return 1 if it's not available.

20.0.152 How to know the calling function?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: On Mac you can use a helper function like this this code:

Example:

```
Public Function CallingFunction() as string
// Query name of calling function of a function
```

```
#Pragma BreakOnExceptions false
```

```
try
```

```
// raise a dummy exception
dim r as new NilObjectException
raise r
```

```
catch x as NilObjectException
```

```
// get stack
dim stack() as string = x.Stack
```

```
// pick function name and return
dim name as string = stack(2)
Return name
```

```
end try
End Function
```

Notes: You need to include function names in your application.

20.0.153 How to launch an app using it's creator code?

Plugin Version: all, Platform: macOS.

Answer: Send an AppleEvent "oapp" with the creator code to the Finder ("MACS"):

Example:

```
Dim a as AppleEvent
dim creator as string

creator = "MSIE" ' here the Internet Explorer

a = NewAppleEvent("aevt", "odoc", "MACS")
a.Timeout = -1

a.ObjectSpecifierParam("—") = GetUniqueIDObjectDescriptor("appf", nil, creator)

if not a.send then
msgBox "An error has occured"
else

end if
```

20.0.154 How to launch disc utility?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use this code:

Example:

```
dim f as FolderItem = LaunchServicesFindApplicationForInfoMBS("", "com.apple.DiskUtility", "")

if f<>Nil then
f.Launch
end if
```

Notes: This works even if people renamed the disc utility or moved it to another folder.

20.0.155 How to make a lot of changes to a REAL SQL Database faster?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You may try to embed your changes to the database between two transaction calls.

Example:

```
dim db as Database // some database

db.SQLiteExecute "BEGIN TRANSACTION"
// Do some Stuff
db.SQLiteExecute "END TRANSACTION"
```

Notes: This can increase speed by some factors.

20.0.156 How to make a NSImage object for my retina enabled app?

Plugin Version: all, Platform: macOS.

Answer: You can use code like this:

Example:

```
Function NewRetinaImage(pic as Picture, mask as Picture = nil) As NSImageMBS
// first make a NSImageMBS from it
dim n as new NSImageMBS(pic, mask)

// now set to half the size, so we have 2x pixels for the image
n.size = new NSSizeMBS(n.width/2, n.height/2)

// and return
Return n
End Function
```

Notes: The thing to do is to have 2x the pixels, but assign a size to the image which gives it the right size in points.

You can pass the NSImageMBS from here to NSMenuItemMBS. For Retina displays, the full resolution is used. For others it will be reduced.

20.0.157 How to make a window borderless on Windows?

Plugin Version: all, Platform: Windows.

Answer: Try this declares:

Example:

```
// Sets window to borderless popup type, and sets its initial dimensions.
// Call this method, then Win32SetBorderlessPos, and then RB's Show
// method. Use RB Frame type 7 (Global Floating Window).
```

```
Const SWP_NOMOVE = &H2
Const SWP_FRAMECHANGED = &H20
Const HWND_TOPMOST = -1
Const GWL_STYLE = -16
Const WS_POPUPWINDOW = &H8080000
```

```
Dim styleFlags as Integer
```

```
#If TargetWin32 Then
```

```
Declare Function SetWindowLong Lib "user32" Alias "SetWindowLongA" (hwnd as Integer, nIndex as Integer, dwNewLong as Integer) as Integer
```

```
Declare Function SetWindowPos Lib "user32" (hwnd as Integer, hWndInstertAfter as Integer, x as Integer, y as Integer, cx as Integer, cy as Integer, flags as Integer) as Integer
```

```
styleFlags = SetWindowLong( w.WinHWND, GWL_STYLE, WS_POPUPWINDOW )
styleFlags = BitwiseOr( SWP_FRAMECHANGED, SWP_NOMOVE )
styleFlags = SetWindowPos( w.WinHWND, HWND_TOPMOST, 0, 0, wd, ht, styleFlags )
```

```
#EndIf
```

20.0.158 How to make an alias using AppleEvents?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
Sub MakeAlias(folder as folderitem, target as folderitem, aliasname as string)
```

```
dim ev as AppleEvent
```

```
dim myResult as boolean
```

```
dim properties as AppleEventRecord
```

```
ev = NewAppleEvent("core", "crel", "MACS")
```

```
ev.MacTypeParam("kocl") = "alis"
```

```
ev.FolderItemParam("to ") = target
```

```
ev.FolderItemParam("insh") = folder
```

```
properties=new AppleEventRecord
```

```

properties.StringParam("pnam")=aliasname

ev.RecordParam("prdt")=properties

myResult = ev.send
// true on success, false on error
End Sub

```

Notes: Call it like this:

```
MakeAlias SpecialFolder.Desktop, SpecialFolder.Desktop.Child("Gif Copy.rb"), "test.rb alias"
```

Seems to not work on Mac OS X 10.6

20.0.159 How to make AppleScripts much faster?

Plugin Version: all, Platform: macOS.

Answer: use "ignoring application responses" like in this example:

```

Notes: on run { fn,fpx,fpy }
ignoring application responses
tell app "Finder" to set the position of folder fn to fpx,fpy
end ignoring
end run

```

20.0.160 How to make double clicks on a canvas?

Plugin Version: all, Platform: macOS.

Answer:

Update: Newer Xojo versions support DoubleClick event, so you don't need this code.

Here's my tip from the tips list on how to add a double-click event to the Canvas control. The technique could easily be used for a window or any Rectcontrol:

Because of its built-in drawing methods, the Canvas control is often used to create custom interface controls. But while the Canvas control has event handlers for most mouse events, it doesn't have an event handler for DoubleClick events. Fortunately, you can add a double-click event handler to a Canvas control easily. Basically, you're going to create a new class based on Canvas and add a double-click event to that. You can then use the new class anytime you need a Canvas with a double-click event.

To create a new Canvas class with a DoubleClick event handler, do this:

1. Add a new class to your project.
2. Set the Super property of the new class to "Canvas".
3. Change the name of this new class to "DoubleClickCanvas".

A double-click occurs when two clicks occur within the users double-click time (set in the Mouse control panel on both Macintosh and Windows) and within five pixels of each other. So, you'll need a few properties to store when and where the last click occurred.

4. Add a new property with the following declaration and mark it as private: lastClickTicks as Integer
5. Add a new property with the following declaration and mark it as private: lastClickX as Integer
6. Add a new property with the following declaration and mark it as private: lastClickY as Integer

Since the Canvas control doesn't have a DoubleClick event, you will need to add one.

7. Add a new event to your class by choosing New Event from the Edit menu and enter "DoubleClick" as the event name.

Double-clicks occur on MouseUp. In order for the mouseUp event to fire, you must return True in the MouseDown event.

8. In the MouseDown event, add the following code:
Return True

In the MouseUp event, you will need to determine what the users double-click time is. This value is represented on both the Mac and Windows in ticks. A tick is 1/60th of a second. Since there isn't a built-in function for this, you'll need to make a toolbox call. The mouseUp event code below makes the appropriate toolbox call for both Macintosh and Windows. It then compares the time of the users last click to the time of the current click and compares the location of the users last click to the location of the current click.

9. Add the following code to the MouseUp event:

```
dim doubleClickTime, currentClickTicks as Integer

#if targetMacOS then
Declare Function GetDbtTime Lib "Carbon" () as Integer
doubleClickTime = GetDbtTime()
#endif

#if targetWin32 then
Declare Function GetDoubleClickTime Lib "User32.DLL" () as Integer
```

```

doubleClickTime = GetDoubleClickTime()/60 // convert to ticks from milliseconds
#endif

currentClickTicks = ticks
//if the two clicks happened close enough together in time
if (currentClickTicks - lastClickTicks) <= doubleClickTime then
//if the two clicks occurred close enough together in space
if abs(X - lastClickX) <= 5 and abs(Y - LastClickY) <= 5 then
DoubleClick //a double click has occurred so call the event
end if
end if
lastClickTicks = currentClickTicks
lastClickX = X
lastClickY = Y

```

10. Now to test out your new DoubleClickCanvas, drag the class from the Project window to a window in your project to create an instance of it.

11. Double-click on the canvas you just added to your window to open the Code Editor. Notice that the canvas has a DoubleClick event handler. In this event handler, add the following code:
BEEP

20.0.161 How to make my Mac not sleeping?

Plugin Version: all, Platform: macOS.

Answer: Just inform the Mac OS about some system activity with code like this:

Example:

```

Sub UpdateSystemActivity()

#if TargetCarbon
declare function myUpdateSystemActivity lib "Carbon" alias "UpdateSystemActivity" (activity as Integer)
as short

const OverallAct = 0 // Delays idle sleep by small amount */
const UsrActivity = 1 // Delays idle sleep and dimming by timeout time */
const NetActivity = 2 // Delays idle sleep and power cycling by small amount */
const HDAActivity = 3 // Delays hard drive spindown and idle sleep by small amount */
const IdleActivity = 4 // Delays idle sleep by timeout time */

dim e as Integer

e=myUpdateSystemActivity(UsrActivity)

```

```
// you may react on an error if e is not 0 after the call.
```

```
#endif
End Sub
```

Notes: You may use another constant if you prefer some different behavior. Call it maybe every second.

20.0.162 How to make my own registration code scheme?

Plugin Version: all, Platform: Windows.

Answer: There are excellent articles about how to make a registration code scheme, but you can also simply use our RegistrationEngineMBS class.

Notes: If you need a license text, why not use the one from Xojo as a starting point?

20.0.163 How to make small controls on Mac OS X?

Plugin Version: all, Platform: macOS.

Answer: You can try this code on Mac OS X:

Example:

```

'/*
** Use the control's default drawing variant. This does not apply to
** Scroll Bars, for which Normal is Large.
**/
const kControlSizeNormal = 0

'/*
** Use the control's small drawing variant. Currently supported by
** the Check Box, Combo Box, Radio Button, Scroll Bar, Slider and Tab
** controls.
**/
const kControlSizeSmall = 1

'/*
** Use the control's small drawing variant. Currently supported by
** the Indeterminate Progress Bar, Progress Bar and Round Button
** controls.
**/
const kControlSizeLarge = 2

```

```

'/*
' * Control drawing variant determined by the control's bounds. This
' * ControlSize is only available with Scroll Bars to support their
' * legacy behavior of drawing differently within different bounds.
' */
const kControlSizeAuto = &hFFFF

const kControlSizeTag = "size"

declare function SetControlData lib "Carbon" (controlhandle as Integer, part as short, tagname as OS-
Type, size as Integer, data as ptr) as short

dim m as MemoryBlock

m=NewMemoryBlock(2)
m.UShort(0)=kControlSizeSmall

Title=str(SetControlData(CheckBox1.Handle, 0, kControlSizeTag, 2, m))

```

20.0.164 How to mark my Mac app as background only?

Plugin Version: all, Platform: macOS.

Answer: You can run a build script on each build with this code:

Example:

```

Dim App As String = CurrentBuildLocation + "/" + CurrentBuildAppName + ".app"
Call DoShellCommand("/usr/bin/defaults write " + App + "/Contents/Info ""NSUIElement"" YES")

```

Notes: This will set the NSUIElement flag to YES.

20.0.165 How to move a file or folder to trash?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like below:

Example:

```

Function MoveToTrash(f as FolderItem) As Boolean
#if TargetMacOS then
dim r as FolderItem
dim e as Integer = MacFileOperationMBS.MoveObjectToTrashSync(f, r, MacFileOperationMBS.kFSFile-
OperationDefaultOptions)

```

```

if e = 0 then
Return true // Ok
end if

#elseif TargetWin32 then
dim w as new WindowsFileCopyMBS

dim flags as Integer = w.FileOperationAllowUndo + w.FileOperationNoErrorUI + w.FileOperationSilent
+ w.FileOperationNoConfirmation
if w.FileOperationDelete(f, flags) then
Return true // OK
end if

flags = w.FileOperationNoErrorUI + w.FileOperationSilent + w.FileOperationNoConfirmation
if w.FileOperationDelete(f, flags) then
Return true // OK
end if
#else
// Target not supported
break
Return false
#endif
End Function

```

Notes: If you want to move a file to trash, you could use `f.movefileto f.trashfolder`, but that will overwrite existing files in the trash. You can use our `MacFileOperationMBS` class to move a file on Mac to the trash. And it uses the same code as the Finder, so files are renamed when the same name is already in use in the trash:

On Windows we use `WindowsFileCopyMBS` class.
Requires Mac OS X 10.5.

20.0.166 How to move an application to the front using the creator code?

Plugin Version: all, Platform: macOS.

Answer: This makes SimpleText (Code ttxt) to the frontmost application:

Example:

```

dim a as appleevent

a=newappleEvent("misc","actv","ttxt")

```

```
if a.send then
end if
```

Notes: (Code is Mac only)

20.0.167 How to move file with ftp and curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can set post/pre quotes to have ftp commands executed before or after the download/upload.

Example:

```
dim d as CURLMBS // your curl object

// rename/move file
dim ws() As String
ws.Append "RNFR Temp.txt"
ws.append "RNT0 MyFile.txt"

d.SetOptionPostQuote(ws)
```

Notes: Use SetOptionPostQuote, SetOptionPreQuote or SetOptionQuote.

The ftp commands you pass here are native ftp commands and not the commands you use with ftp applications. So rename is two commands. First RNFR to tell where to rename from and second RNT0 with the new file name. To delete use DELE and the file path.

20.0.168 How to normalize string on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like below:

Example:

```
Function Normalize(t as string) As string
const kCFStringNormalizationFormD = 0 // Canonical Decomposition
const kCFStringNormalizationFormKD = 1 // Compatibility Decomposition
const kCFStringNormalizationFormC = 2 // Canonical Decomposition followed by Canonical Composition
const kCFStringNormalizationFormKC = 3 // Compatibility Decomposition followed by Canonical Composition

dim s as CFStringMBS = NewCFStringMBS(t)
dim m as CFMutableStringMBS = s.Normalize(kCFStringNormalizationFormD)
```

```
Return m.str  
End Function
```

Notes: This uses Apple's CFString functions to normalize unicode variants.

20.0.169 How to obscure the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub ObscureCursor Lib "Carbon" ()
```

```
ObscureCursor
```

Notes: The MBS Plugin has this function, but it's not supported for Windows.

20.0.170 How to open icon file on Mac?

Plugin Version: all, Platform: macOS.

Answer: Use the NSImageMBS class like this:

Example:

```
dim f as FolderItem = SpecialFolder.Desktop.Child("test.ico")
```

```
dim n as new NSImageMBS(f)
```

```
window1.Backdrop = n.CopyPictureWithMask
```

20.0.171 How to open PDF in acrobat reader?

Plugin Version: all, Platform: macOS.

Answer: Try this code:

Example:

```
dim pdf as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
```

```

// open PDF in Acrobat Reader on Mac:

// find app
dim bundleID as string = "com.adobe.Reader"
dim app as FolderItem = LaunchServicesFindApplicationForInfoMBS("", bundleID, "")

if app<>nil then

// launch app with parameters

dim docs() as FolderItem
docs.Append pdf

dim param as new LaunchServicesLaunchParameterMBS
param.Defaults = true
param.Application = app

dim x as FolderItem = LaunchServicesOpenXMBS(docs, param)

// on failure, simply launch it
if x = nil then
pdf.Launch(true)
end if

else
pdf.Launch(true)
end if

```

Notes: On Windows, simply use pdf.launch or WindowsShellExecuteMBS.

20.0.172 How to open printer preferences on Mac?

Plugin Version: all, Platform: macOS.

Answer: You can use our OpenMacOSXPreferencesPaneMBS function like this:

Example:

```

dim e as Integer = OpenMacOSXPreferencesPaneMBS("PrintAndFax")
if 0 = e then
MsgBox "OK"
elseif e = -43 then
MsgBox "File not found."
else
MsgBox "Error: "+str(e)
end if

```

20.0.173 How to open special characters panel on Mac?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have functions for that in Cocoa and Carbon.

Example:

```
dim a as new NSApplicationMBS
a.orderFrontCharacterPalette
```

Notes: For Cocoa, you can use `orderFrontCharacterPalette` method in `NSApplicationMBS` class.

Or simply for Carbon and Cocoa the `ShowCharacterPaletteMBS` method.

20.0.174 How to optimize picture loading in Web Edition?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the `WebPicture` class.

Notes: Take your picture and create a `WebPicture` object. Store this `WebPicture` in a property of the `WebPage`, `Session` or `app` (as global as possible). On the first time you use this picture on an user session, the browser will load it. Second time you use it, the browser will most likely pick it from the cache.

Having pictures in `App` or some module reuses the same picture for all sessions which reduces memory footprint.

This does not work well with pictures you change very often or use only for one webpage on one user.

If you like to see an example, check our `Map` example.

20.0.175 How to parse XML?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use code like this:

Example:

```
dim s as string = "<test><test /></test>"
```

```
try
```

```

dim x as new XmlDocument(s)
MsgBox "OK"
catch xe as XmlException
MsgBox "invalid XML"
end try

```

Notes: If you got an exception, you have a parse error.

20.0.176 How to play audio in a web app?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the HTML5 audio tag and control it with javascript.

Notes: This is just another example app I made today. It plays a christmas song. The audio file is provided by the application to the server, so no external web server is needed and this application can run stand alone. To compile and run you need Xojo 2010r5.

In the open event we search the audio files and open them as binarystreams. We create the two webfile objects. Those webfiles are part of the app class, so we have them globally. There we set the data with the content of our streams. We also define file names and mime types. They are needed so browser know what we have here:

```

audioFileM4V = new WebFile
audioFileM4V.Data = bM.Read(BM.Length)
audioFileM4V.Filename = "music.m4a"
audioFileM4V.MIMEType = "audio/m4a"

```

```

audioFileOGG = new WebFile
audioFileOGG.Data = bO.Read(BO.Length)
audioFileOGG.Filename = "music.ogg"
audioFileOGG.MIMEType = "audio/ogg"

```

Next in the open event of the webpage we have a PageSource control. The location is set to be before content. In the open event we define the html code for this. First we pick the URLs for the audio files. Than we build the html to use the audio tag. As you see, we give it an ID for later use and have it preload automatically. If you add an autoplay tag, you can have the audio play right away. Inside the audio tag we have two sources so we provide audio for both Firefox (OGG) and Safari (MPEG4). Finally we have a text to display if HTML5 audio tag is not supported.

You can set the source in the EditSource event:

```
dim urlO as string = app.audioFileOGG.URL
dim urlM as string = app.audioFileM4V.URL
me.Source = "<audio id=""mymusic"" preload=""auto""><source src="""+urlO+""" type=""audio/ogg""
/><source src="""+urlM+""" type=""audio/mpeg"" />Your browser does not support the audio ele-
ment.</audio>"
```

Next in the Play button we execute code to play the audio. This is a short javascript code which searches in the html document for the element with the ID "mymusic" which is the ID of our audio tag above. Once we got the object, we call it's play method to start playback.

```
me.ExecuteJavaScript("document.getElementById('mymusic').play();")
```

same for pause:

```
me.ExecuteJavaScript("document.getElementById('mymusic').pause();")
```

and finally for changing volume:

```
me.ExecuteJavaScript("document.getElementById('mymusic').volume="+str(me.Value/100.0)+"");")
```

20.0.177 How to pretty print xml?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use the XML Transform method with the right XLS.

Notes: Learn more here:

<http://docs.xojo.com/index.php/XMLDocument.Transform>

20.0.178 How to print to PDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: This code below shows how to redirect printing to a PDF file on Mac OS X.

Example:

```
// get Xojo printer setup
dim p as new PrinterSetup

// now put it into NSPrintInfo to manipulate
dim n as new NSPrintInfoMBS
n.SetupString = p.SetupString
```

```

// change destination to file
dim f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
n.SetSaveDestination(f)

// move back
p.SetupString = n.SetupString

// and print as usual
dim g as Graphics = OpenPrinter(p)
g.DrawString "Hello World", 20, 20

```

Notes: And you can use normal graphics class for that.

20.0.179 How to query Spotlight's Last Open Date for a file?

Plugin Version: all, Platform: macOS.

Answer: You can use a MDItemMBS objec to query this value:

Example:

```

Function LastOpenedDate(Extends F As FolderItem, DefaultOtherDates As Boolean = True) As Date
#If TargetMacOS Then
Dim xMDItem as New MDItemMBS(F)
Dim xDate as Variant

If xMDItem <>Nil Then
xDate = xMDItem.GetAttribute(xMDItem.kMDItemLastUsedDate).DateValue
If xDate IsA Date Then Return xDate
Else
If xDate <>Nil Then Break
End If
#EndIf

If DefaultOtherDates Then
If F.ModificationDate <>Nil Then Return F.ModificationDate
If F.CreationDate <>Nil Then Return F.CreationDate
End If
End Function

```

Notes: Thanks for Josh Hoggan for this example code.

20.0.180 How to quit windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetwin32 then
dim i1,i2,r as Integer
declare function ExitWindowsEx lib "user32" (uFlags as Integer, dwReserved as Integer) as Integer
i1 = 2
i2 = 0
r = ExitWindowsEx(i1,i2)
if r<>0 then
' Error()
end if

#endif
```

Notes: uFlags parameters:

```
'4 = EWX_Force
'0 = EWX_Logoff
'2 = EWX_Reboot
'1 = EWX_shutdown, should shut down computer
```

Also check the ExitWindowsMBS method.

20.0.181 How to read a CSV file correctly?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With all the rules for quotes and delimiters, you can simply use the SplitCommaSeparatedValuesMBS method in our plugins like this:

Example:

```
dim f as FolderItem = SpecialFolder.Desktop.Child("test.csv")
dim t as TextInputStream = f.OpenAsTextFile

while not t.EOF
dim s as string = t.ReadLine(encodings.ASCII)

dim items() as string = SplitCommaSeparatedValuesMBS(s, ";", """")
```

```
List.AddRow """
dim u as Integer = UBound(items)
for i as Integer = 0 to u
List.Cell(List.LastIndex,i) = items(i)
next

wend
```

Notes: Please make sure you choose the right text encoding.

20.0.182 How to read the command line on windows?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
#if targetwin32 then
dim line as string
Dim mem as MemoryBlock

Declare Function GetCommandLineA Lib "kernel32" () As Ptr

mem=GetCommandLineA()
s=mem.cstring(0)

#endif
```

Notes: Newer Xojo versions have a system.commandline property.

20.0.183 How to render PDF pages with PDF Kit?

Plugin Version: all, Platform: Windows.

Answer: Try this code:

Example:

```
// choose a file
dim f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")

// open it as PDF Document
dim sourceFile as New PDFDocumentMBS(f)
```

```

if sourceFile.handle <>0 then // it is a PDF file

// get upper bound of pages
dim c as Integer = sourceFile.pageCount-1

// from first to last page
for n as Integer = 0 to c

// pick that page
dim page as PDFPageMBS = sourceFile.pageAtIndex(n)

// render to image
dim p as NSImageMBS = page.Render

// and convert to RB picture and display
Backdrop = p.CopyPictureWithMask

next

end if

```

Notes: PDFKit works only on Mac OS X.

20.0.184 How to restart a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```

dim ae as appleevent
ae=newappleEvent("FNDR","rest","MACS")
if not ae.send then
msgBox "The computer couldn't be restarted."
end if

```

20.0.185 How to resume ftp upload with curl plugin?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: CURL supports that and you simply need to set the right options.

Notes: First of course OptionUpload must be true. Second OptionFTPAppend must be true so the OptionResumeFrom is used. Store there (or in OptionResumeFromLarge) your start value. Don't forget to implement the read event and return data there as requested.

20.0.186 How to rotate a PDF page with CoreGraphics?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: This code opens a PDF and draws the first page into a new PDF with 90° rotation.

Example:

```
// Rotate a PDF page

// our files
dim sourcefile as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
dim destfile as FolderItem = SpecialFolder.Desktop.Child("rotated.pdf")

// open PDF
dim pdf as CGPDFDocumentMBS = sourcefile.OpenAsCGPDFDocumentMBS

// query media size of first page
dim r as CGRectMBS = pdf.MediaBox(1)

// create new PDF
dim c as CGContextMBS = destfile.NewCGPDFDocumentMBS(r,"title","Author","Creator")

// create rotated rectangle
dim nr as new CGRectMBS(0,0,r.Height,r.Width)

// create new page
c.BeginPage nr
c.SaveGState

const pi = 3.14159265

// rotate by 90°
c.RotateCTM pi*1.5

// fix origin
c.TranslateCTM -r.width,0

// draw PDF
c.DrawCGPDFDocument pdf,r,1

// cleanup
c.RestoreGState
c.EndPage
```

```
c = nil

// show in PDF viewer
destfile.Launch
```

Notes: This code is Mac only as it needs CoreGraphics.

20.0.187 How to rotate image with CoreImage?

Plugin Version: all, Platform: macOS.

Answer: Use the code like the one below:

Example:

```
// Rotate image with CoreImage

// load image
dim f as FolderItem = SpecialFolder.Desktop.Child("test.png")
dim image as new CIImageMBS(f)

// rotate 45 degree
dim n as new NSAffineTransformMBS
n.rotateByDegrees(45)

dim TransformFilter as new CIFilterAffineTransformMBS
TransformFilter.inputImage = image
TransformFilter.inputTransform = n

// get result
dim resultImage as CIImageMBS = TransformFilter.outputImage

// for saving to file
dim outputImage as NSImageMBS = resultImage.RenderNSImage(false)

f = SpecialFolder.Desktop.Child("output.png")
dim b as BinaryStream = BinaryStream.Create(f, true)
b.Write outputImage.PNGRepresentation

// as Xojo picture object for display
dim pic as Picture = outputImage.CopyPictureWithMask

Backdrop = pic
```

20.0.188 How to run a 32 bit application on a 64 bit Linux?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Install 32 bit compatibility libraries.

Notes: The package is called ia32-libs for ubuntu (and others).

Some applications need to be run on a 32 bit system as they need some hardware related libraries. Like libUSB or libHID for USB devices.

20.0.189 How to save HTMLViewer to PDF with landscape orientation?

Plugin Version: all, Platform: macOS.

Answer: You can use NSPrintInfoMBS to change the options for PrintToPDFFile function.

Example:

```
// make it landscape
dim n as NSPrintInfoMBS = NSPrintInfoMBS.sharedPrintInfo
n.orientation = n.NSLandscapeOrientation

// save html to file
dim f as FolderItem = SpecialFolder.Desktop.Child("test.pdf")
call HTMLViewer1.PrintToPDFFileMBS(f,10,30,10,30)
```

Notes: You may want to reset options later.
This code is only for Mac OS X.

20.0.190 How to save RTFD?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: With NSTextViewMBS you can use this code to save to RTFD:

Example:

```
// save text as RTFD including image attachments
dim f as FolderItem = GetSaveFolderItem(FileTypes1.ApplicationRtfd, "test.rtf")

if f = nil then Return

dim a as NSAttributedStringMBS = textView.textStorage
dim w as NSFFileWrapperMBS = a.RTFDFileWrapperFromRange(0, a.length, DocumentAttributes)

dim e as NSErrorMBS
if w.writeToFile(f, e) then
```

```

else
MsgBox e.LocalizedDescription
end if

```

Notes: For TextArea you can query the underlying NSTextViewMBS object via TextArea.NSTextViewMBS method.

20.0.191 How to save RTFD?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: How to load PDF to htmlviewer on desktop?

Example:

```

Public Sub LoadPDFData(viewer as HTMLViewer, PDFData as string)
Dim base64string As String = EncodeBase64(PDFData)

// remove line endings to make it a big line
base64string = ReplaceLineEndings(base64string, "")

// build data URL
// https://en.wikipedia.org/wiki/Data_URI_scheme
Dim dataURL As String = "data:application/pdf;base64," + base64string

// show in webviewer
HTMLViewer1.LoadURL(dataURL)

// may not work everywhere due to URL length limit
// for Web projects, use WebFile instead!
End Sub

```

Notes: This avoids a temporary file, which may also work.
For Web Apps, please use WebFile.

20.0.192 How to scale a picture proportionally with mask?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: For a proportional scaling, we calculate the new picture size relative to the target maximum size.

Example:

```

Function ProportionalScaledWithMask(extends pic as Picture, Width as Integer, Height as Integer) As Picture
// Calculate scale factor

dim faktor as Double = min( Height / Pic.Height, Width / Pic.Width)

// Calculate new size
dim w as Integer = Pic.Width * faktor
dim h as Integer = Pic.Height * faktor

// create new picture
dim NewPic as new Picture(w,h,32)

// check if we have a mask and clear it
dim m as picture = pic.mask(False)
pic.mask = nil

// draw picture in the new size
NewPic.Graphics.DrawPicture Pic, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height

if m <>nil then
// restore mask and scale it
pic.mask = m
NewPic.mask.Graphics.DrawPicture m, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height
end if

// return result
Return NewPic
End Function

```

Notes: This version handles mask. As you see we actually have to remove mask in order to copy the picture part correctly.

20.0.193 How to scale a picture proportionally?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: For a proportional scaling, we calculate the new picture size relative to the target maximum size.

Example:

```

Function ProportionalScaled(extends pic as Picture, Width as Integer, Height as Integer) As Picture
// Calculate scale factor

dim faktor as Double = min( Height / Pic.Height, Width / Pic.Width)

```

```

// Calculate new size
dim w as Integer = Pic.Width * faktor
dim h as Integer = Pic.Height * faktor

// create new picture
dim NewPic as new Picture(w,h,32)

// draw picture in the new size
NewPic.Graphics.DrawPicture Pic, 0, 0, w, h, 0, 0, Pic.Width, Pic.Height

// return result
Return NewPic
End Function

```

Notes: This does not handle mask, but you can scale the mask the same way and assign it to the new picture.
(see other FAQ entry with mask)

20.0.194 How to scale/resize a CIIImageMBS?

Plugin Version: all, Platform: Windows.

Answer: Use the CIFilterLanczosScaleTransform filter to scale down a picture to a specific size.

Example:

```

Dim pic As Picture = LogoMBS(500)
Dim image As CIIImageMBS = CIIImageMBS.imageWithPicture(pic)

Dim filter As New CIFilterLanczosScaleTransformMBS

Const targetWidth = 600.0
Const targetHeight = 400.0

Dim scale As Double = targetHeight / image.Extent.Height
Dim aspect As Double = targetWidth / (image.Extent.Width * scale)

filter.inputImage = image
filter.inputScale = scale
filter.inputAspectRatio = aspect

Dim result As Picture = filter.outputImage.RenderPicture

Backdrop = result

```

Notes: This is same code as our scaleTo convenience method.

20.0.195 How to scale/resize a picture?

Plugin Version: all, Platform: Windows.

Answer: There are several ways to scale or resize a picture. The easiest way may be the ScaleMBS function in the Picture class.

Example:

```
dim Original,Scaled as Picture
```

```
Original=LogoMBS(500)
Scaled=Original.ScaleMBS(100,100,true)
```

Notes: The plugin ways:

- GraphicsMagick can scale/resize.
- CoreImage scale filter may result in the fastest and best images on Mac OS X 10.4.
- NSImageMBS can scale, but is Mac OS X only.
- CGImageMBS can scale, but is Mac OS X only.
- CIImageMBS can scale, but is Mac OS X only.
- QuickTime Graphics exporter and importer can be connected to scale. (this was used more often a few years ago)
- ImageMagick can scale very nice and crossplatform. But the ImageMagick libraries are big.
- The picture.ScaleMBS function is self written and results in equal output on Mac, Windows and Linux without any additional libraries installed.
- Picture.ScalingMBS does crossplatform scaling with several modes.

with pure Xojo:

- make a new picture and draw the old one with new size inside.

20.0.196 How to search with regex and use unicode codepoints?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can specify unicode characters in search string with backslash x and digits.

Example:

```
dim r as RegExMbs
dim s as string
dim c as Integer
```

```

s="123 √√√° ABC 456"

r=new RegExMBS
if r.Compile("√.") then
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if

r=new RegExMBS
if r.Compile("\xF6.") then // finds √ using Unicode codepoint
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if

```

20.0.197 How to see if a file is invisible for Mac OS X?

Plugin Version: all, Platform: macOS.

Answer: Try this function:

Example:

```

Function Invisible(F As FolderItem) As Boolean
Dim TIS As TextInputStream
Dim S,All As String
Dim I as Integer
dim g as folderitem

If Left(F.Name,1)="." or not f.visible Then
Return True
End If

g=F.Parent.Child(".hidden")
If g.Exists Then
TIS=g.OpenAsTextFile
if tis<>Nil then
All=TIS.ReadAll
For I=1 to CountFields(All,Chr(11))
S=NthField(All, Chr(11), I)

```

```

If S=F.name Then
Return True
End If
Next
end if
End if
End Function

```

20.0.198 How to set cache size for SQLite or REALSQLDatabase?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You use the pragma cache_size command on the database.

Example:

```

// set cache size to 20000 pages which is about 20 MB for default page size
dim db as REALSQLDatabase
db.SQLExecute "PRAGMA cache_size = 20000"

```

Notes: Default cache size is 2000 pages which is not much.

You get best performance if whole database fits in memory.

At least you should try to have a cache big enough so you can do queries in memory.

You only need to call this pragma command once after you opened the database.

20.0.199 How to set the modified dot in the window?

Plugin Version: all, Platform: macOS.

Answer: Try this declares:

Example:

```

window1.ModifiedMBS=true

```

20.0.200 How to show a PDF file to the user in a Web Application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use a WebHTMLViewer control and load the

Example:

```

dim CurrentFile as WebFile // a property of the WebPage

// define the PDF file
CurrentFile = new WebFile
CurrentFile.Filename = "test.pdf"
CurrentFile.MIMEType = "application/pdf"
CurrentFile.Data = "some pdf data" // MyDynaPDF.GetBuffer

// load into html viewer
HTMLViewer1.URL = CurrentFile.URL

```

Notes: See our Create PDF example for the Xojo Web Edition.

20.0.201 How to show Keyboard Viewer programmatically?

Platform: macOS.

Answer: Use Xojo or AppleScript to launch the KeyboardViewerServer.app.

Example:

```

dim a as new AppleScriptMBS
dim text as string
dim lines(-1) as string

lines.append "set theApplication to ""KeyboardViewerServer""
lines.append "set thePath to ""/System/Library/Components/KeyboardViewer.component/Contents/Shared-
Support/KeyboardViewerServer.app""
lines.append ""
lines.append "set POSIXPath to ((POSIX file thePath) as string)"
lines.append "tell application ""System Events"" to set isRunning to 0 <(count (application processes whose
name is theApplication))"
lines.append "if isRunning then tell application POSIXPath to quit"
lines.append "delay 0.15"
lines.append ""
lines.append "ignoring application responses"
lines.append " tell application POSIXPath to run"
lines.append "end ignoring"

text=join(lines,EndOfLine.macintosh)

a.Compile text
a.Execute

```

Notes: AppleScript code:

```
set theApplication to "KeyboardViewerServer"
set thePath to "/System/Library/Components/KeyboardViewer.component/Contents/SharedSupport/KeyboardViewerServer.app"
```

```
set POSIXPath to ((POSIX file thePath) as string)
tell application "System Events" to set isRunning to 0 <(count (application processes whose name is theApplication))
if isRunning then tell application POSIXPath to quit
delay 0.15
```

```
ignoring application responses
tell application POSIXPath to run
end ignoring
```

20.0.202 How to show the mouse cursor on Mac?

Plugin Version: all, Platform: macOS.

Answer: Try this declare:

Example:

```
Declare Sub ShowCursor Lib "Carbon" ()
```

```
ShowCursor
```

Notes: The MBS Plugin has this function and supports it on Windows, too.

20.0.203 How to shutdown a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
dim ae as appleevent
ae=newappleEvent("FNDR", "shut", "MACS")
if not ae.send then
msgBox "The computer couldn't be shutdown."
end if
```

Notes: Or toolbox call (Attention: This method will stop the computer immediatly: No document asked to be saved, all applications quitting without knowing).

```
Declare Sub ShutDwnPower Lib "Carbon" ()
ShutDwnPower
```

20.0.204 How to sleep a Mac?

Plugin Version: all, Platform: macOS.

Answer: Ask the Finder via Apple Events:

Example:

```
dim ae as appleevent
ae=newappleEvent("FNDR","slep","MACS")
if not ae.send then
msgBox "The computer doesn't want to sleep."
end if
```

20.0.205 How to speed up rasterizer for displaying PDFs with DynaPDF?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Here a few speed tips:

Notes:

- Use the DynaPDFRasterizerMBS function instead of our render functions.
- Reuse DynaPDFRasterizerMBS as long as the target picture size doesn't change.
- Import only the PDF pages you want to display.
- Let DynaPDF do zooming, rotating or other effects instead of you change it.

20.0.206 How to use PDFLib in my RB application?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The PDFlib plugin was discontinued in favor of our DynaPDF plugin.

Notes: If you need help to move, please contact us.

20.0.207 How to use quotes in a string?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Just double them.

Example:

```
msgbox "This String contains ""quotes"""
```

20.0.208 How to use Sybase in Web App?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use our MBS Xojo SQL Plugin to connect to a Sybase Database in your web application.

Notes: If you see db.Connect giving the error message "cs_ctx_alloc ->CS_MEM_ERROR", than some things are not setup right for Sybase.

The Apache process may not have all the SYBASE environment variables being set when the CGI was launched.

Adding these lines to /etc/httpd/conf/httpd.conf stopped the faux memory errors for us:

```
SetEnv LD_LIBRARY_PATH /opt/sybase/OCS-15_0/lib:/opt/sybase/OCS-15_0/lib3p64:/opt/sybase/OCS-15_0/lib3p:
SetEnv SYBROOT /opt/sybase
SetEnv SYBASE_OCS /opt/sybase
SetEnv SYBASE /opt/sybase
```

20.0.209 How to use the Application Support folder?

Plugin Version: all, Platform: macOS.

Answer:

I was saving a registration code for an app to the Preference folder. People on the list have suggested that it would be better in the ApplicationSupportFolder. How do I save the file called CWWPrefs into that folder using MBS?

I have checked for examples and the docs but can't see how to apply it

```
//f = SpecialFolder.Preferences.child("CWWPrefs")
f = ApplicationSupportFolderMBS(-32768)
```

Example:

```

dim folder,file as FolderItem

folder = createApplicationSupportFolderMBS(-32763)

if folder=nil then
// Some very old Mac OS Versions may not support it
// or the plugin may fail for any reason
folder=SpecialFolder.Preferences
end if

file=folder.Child("CWWPrefs")

MsgBox file.NativePath

```

Notes:

You may not be able to write there with a normal user account!

20.0.210 How to use the IOPMCopyScheduledPowerEvents function in Xojo?

Plugin Version: all, Platform: macOS.

Answer: You can use the following code which does this using the SoftDeclareMBS class.

Example:

```

Sub Open()
dim c as CFDateMBS
dim t as CFAbsoluteTimeMBS

// get current date
c=NewCFDateMBS

// in absolute time (seconds since x)
t=c.AbsoluteTime

// add 600 seconds (= 10 Minutes)
t.Value=t.Value+600

// Make a Date from it
c=t.Date

// Schedule the event
// 0 on success
// E00002C1 for missing root rights

```

```

Title=hex(schedulePowerEvent(c, "wake"))

// Just for information, display the scheduled stuff
CFShowMBS CopyScheduledPowerEvents
End Sub

Function CopyScheduledPowerEvents() As carrayMBS
dim s as SoftDeclareMBS
dim m as MemoryBlock

s=new SoftDeclareMBS

if s.LoadLibrary("IOKit.framework") then
if s.LoadFunction("IOPMCopyScheduledPowerEvents") then
if s.CallFunction(0,nil) then
Return NewCFArrayMBSHandle(s.Result,true)
else
MsgBox "Failed to Call IOPMCopyScheduledPowerEvents."
end if
else
MsgBox "Failed to load IOPMCopyScheduledPowerEvents."
end if
else
MsgBox "Failed to load IOKit."
end if

Return nil
End Function

Function SchedulePowerEvent(time_to_wake as CFDateMBS, Type as CFStringMBS) as Integer
dim s as SoftDeclareMBS
dim m as MemoryBlock

'/*
'* Types of power event
'* These are potential arguments to IOPMSchedulePowerEvent().
'* These are all potential values of the kIOPMPowerEventTypeKey in the CFDictionaryes
'* returned by IOPMCopyScheduledPowerEvents().
'*/
'/*!
'@define kIOPMAutoWake
'@abstract Value for scheduled wake from sleep.
'*/
'#define kIOPMAutoWake "wake"
,
'/*!
'@define kIOPMAutoPowerOn
'@abstract Value for scheduled power on from off state.

```

```

*/
#define kIOPMAutoPowerOn "poweron"
,
/*!
#define kIOPMAutoWakeOrPowerOn
@abstract Value for scheduled wake from sleep, or power on. The system will either wake OR
power on, whichever is necessary.
*/
,
#define kIOPMAutoWakeOrPowerOn "wakepoweron"
/*!
#define kIOPMAutoSleep
@abstract Value for scheduled sleep.
*/
,
#define kIOPMAutoSleep "sleep"
/*!
#define kIOPMAutoShutdown
@abstract Value for scheduled shutdown.
*/
,
#define kIOPMAutoShutdown "shutdown"

s=new SoftDeclareMBS

if s.LoadLibrary("IOKit.framework") then
if s.LoadFunction("IOPMSchedulePowerEvent") then

m=NewMemoryBlock(12)
m.Long(0)=time_to_wake.handle
m.Long(4)=0 // nil
m.Long(8)=type.Handle

if s.CallFunction(3,m) then
Return s.Result
end if
end if
end if

End Function

```

Notes: Requires Mac OS X and to execute root rights.

20.0.211 How to validate a GUID?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use this function below which uses a regular expression to verify that the string is a valid UUID/GUID:

Example:

Function IsGUID(guid as string) As Boolean

dim r as new RegEx

```
r.SearchPattern = "^(\{ 0,1 } ([ 0-9a-fA-F ] ) { 8 } - ( [ 0-9a-fA-F ] ) { 4 } - ( [ 0-9a-fA-F ] ) { 4 } - ( [ 0-9a-fA-F ] ) { 4 } - ( [ 0-9a-fA-F ] ) { 12 } \} { 0,1 } )$ "
```

Return r.Search(guid)<>nil

End Function

Notes: Simply parsing the GUID with CFUUIDMBS does not give the same result as CFUUIDMBS will also take a string like "DDDD".

20.0.212 How to walk a folder hierarchie non recursively?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Use code like this one:

Example:

Sub Walk(folder as FolderItem)

dim folders() as FolderItem

folders.Append folder

while UBound(folders)>=0

dim currentFolder as FolderItem = folders.pop

dim c as Integer = currentFolder.Count

for i as Integer = 1 to c

dim item as FolderItem = currentFolder.TrueItem(i)

if item = Nil then

// no permission

elseif item.Visible then // only visible

if item.Directory then

folders.Append item

```
else
// work with file here
end if

end if

next

wend
End Sub
```

Notes: As you see we go with a long loop which runs until we don't have more folders to process. We ignore items we can't access due to permission limits. And we only work visible items. If you like, check `folderitem.isBundleMBS` on item to handle packages and applications better on Mac OS X.

20.0.213 I got this error: PropVal, QDPictMBS.Name (property value), Type mismatch error. Expected CGDataProviderMBS, but got Variant, Name:QDPictMBS

Plugin Version: all, Platform: macOS.

Answer: The plugins MacOSX and MacOSXCF belong together. If you use one part, please also install the other part.

Notes: We splitted the plugin because the Xojo IDE on Windows crashed on compilation.

20.0.214 I registered the MBS Plugins in my application, but later the registration dialog is shown.

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: There are two main reasons.

Notes: 1. you may use the plugin before registering them. This is often the case if you register in a window open event and use the plugin in a control open event.

On the console on Mac OS X or Windows, you may see a message like this "MBS Plugins were used by the application before the RegisterMBSPlugin function was called. Please fix this in your code!".

2. you may have mixed different plugin versions which are not compatible.

In this case you can see a message "Internal plugin registration error." on the console on Mac OS X. Newer plugins may show a message dialog reporting this. Older version simply think they are not registered.

If the installer just merges old and new applications, users may have libraries of older and newer plugin versions in the libs folder. If your application loads the wrong version, the registration fails.

If you use remote debugging, make sure you clear the temporary files there, too. Otherwise you may have old DLLs on your hard disc which may disturb your application.

You can run into issues if you use your registration code on different places of your app. Please register only once in app.open (or app Constructor). If you have several codes, simply call them one after the other.

Also check that you only call RegisterMBSPlugin with valid serial number. If you later call RegisterMBSPlugin with Demo like in example code above, you remove the license.

Next check if you can clear the Xojo caches and that helps. This includes the Xojo Scratch folder and the Plugins & Project caches. Simply locate those folders and delete them. For Windows look in hidden AppData folder in your user folder. For Mac, please check textasciitilde /Library/Caches and your temp folders.

Finally make sure you use the right serial number. Not an older one or a misspelled one.

20.0.215 I want to accept Drag & Drop from iTunes

Plugin Version: all, Platform: macOS.

Answer: You need to accept AcceptMacDataDrop "itun" and Handle the DropObject.

Example:

```
Sub Open()
window1.AcceptMacDataDrop "itun"
End Sub
```

```
Sub DropObject(obj As DragItem)
dim s as string
dim f as folderItem
dim d as CFDictionaryMBS
dim o as CFObjectMBS
dim key as CFStringMBS
dim dl as CFDictionaryListMBS
dim i,c as Integer
dim u as CFURLMBS
dim file as FolderItem
```

```
if obj.MacDataAvailable("itun") then
s = obj.MacData("itun")
```

```
// Parse XML
o=NewCFOBJECTMBSFromXML(NewCFBinaryDataMBS(s))

// Make dictionary
if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)

// get Tracks Dictionary
key=NewCFStringMBS("Tracks")
o=d.Value(key)

if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)
dl=d.List

// Walk over all entries in the Tracks dictionary
c=dl.Count-1
for i=0 to c
o=dl.Value(i)

if o isa CFDictionaryMBS then
d=CFDictionaryMBS(o)

key=NewCFStringMBS("Location")
o=d.Value(key)
if o isa CFStringMBS then
u=NewCFURLMBS(CFStringMBS(o),nil)

file=u.file
if file<>nil then
MsgBox file.NativePath
end if
end if
end if
next
end if
end if
end if
End Sub
```

Notes: The code above inside a window on Xojo 5.5 with MBS Plugin 5.3 will do it nice and show the paths.

20.0.216 I'm drawing into a listbox but don't see something.

Plugin Version: all.

Answer: If you draw this in a listbox cellbackground, you need to draw on the correct position

Example:

```
Function CellBackgroundPaint(g As Graphics, row as Integer, column as Integer) As Boolean
dim f as FolderItem
f=SpecialFolder.Desktop
f.DrawWideIconMBS(g,listbox1.left,listbox1.top+row*20,16)
Return true
End Function
```

Notes: Try this in a listbox. The Graphics object there has a clipping and an offset which the plugin doesn't know about.

20.0.217 I'm searching for a method or so to move a window from position x.y to somewhere else on the screen.

Platform: macOS.

Answer:

The code I produced in RB isn't smooth enough. Is there a call in MBS, if not, can it be done? The speed of it has to be like the show of a DrawerWindow.

Try the declare below for Carbon. With WindowLib it will work on Mac OS 8.5 and newer.

Notes:

See Window.Transition functions.

20.0.218 If I use one of your plug-ins under windows, would this then impose the use of dll after compilation or my would my compiled soft still be a stand-alone single file software?

Platforms: macOS, Linux, Windows.

Answer: Stand alone.

Notes: Xojo compiles all used plugins into the application binary.

Some plugin parts need external dlls but you will find that in the documentation. (e.g. pdflib for some classes)

20.0.219 Is the fn key on a powerbook keyboard down?

Plugin Version: all, Platform: macOS.

Answer: I am unable to figure out how or if it is possible to detect if the fn key is down on a powerbook keyboard. Is it possible?

Example:

' Window.Open Event of a blank project:

```
dim i as Integer

for i=0 to 127
if keyboard.asynckeydown(i) then
title=str(i) // found
return
end if
next
title="" // not found
```

Notes: This test application shows the keycode (decimal) 63 for the fn key.

20.0.220 Is there a case sensitive Dictionary?

Plugin Version: all.

Answer: The MBS Plugin has several classes which can work as a replacement.

Notes: First you could use VariantToVariantHashMapMBS or VariantToVariantOrderedMapMBS.

If you know that all keys are Strings or Integers only, you can use the specialized classes which are a little bit faster due to avoiding variants:

```
IntegerToIntegerHashMapMBS class
IntegerToIntegerOrderedMapMBS class
IntegerToStringHashMapMBS class
IntegerToStringOrderedMapMBS class
IntegerToVariantHashMapMBS class
IntegerToVariantOrderedMapMBS class
StringToStringHashMapMBS class
StringToStringOrderedMapMBS class
StringToVariantHashMapMBS class
StringToVariantOrderedMapMBS class
```

20.0.221 Is there a way to use the MBS plugin to get only the visible item and folder count on a volume?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can use the DirectorySizeMBS class for this as in the example below:

Example:

```
dim d as DirectorySizeMBS

d=new DirectorySizeMBS

// volume(1) as my boot volume is very full
if d.update(volume(1),true,0) then
MsgBox str(d.VisibleItemCount)+" visible items, "+str(d.HiddenItemCount)+" invisible items."
end if
```

Notes: Complete Question: Is there a way to use the MBS plugin to get only the visible item and folder count on a volume? The FileCount and FolderCount properties of VolumeInformationMBS seem to provide the total # of items including invisible items such as .DS_Store and more importantly .Trashes which is causing me a great amount of difficulty during a recursive scan of a volume. I've got a progress bar which uses the total of the filecount and foldercount properties as the maximum value, but my routine needs to filter out all invisible items, as it is creating a catalog of a volume for archiving purposes. Any thoughts how I could get accurate number.

20.0.222 Is there an easy way I can launch the Displays preferences panel?

Plugin Version: all, Platform: macOS.

Answer: Use the code below:

Example:

```
dim error as Integer

error=OpenMacOSXPreferencesPaneMBS("Displays")
if error<>0 then
MsgBox "Failed to launch QuickTime System Preferences panel."
end if
```

20.0.223 List of Windows Error codes?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We have a list of windows error codes on our website.

Notes: <http://www.monkeybreadsoftware.de/xojo/winerror.shtml>

20.0.224 Midi latency on Windows problem?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: The issue is system related, not a problem with RB or the plugin.

Notes: Two things will adversely affect the timing:

(1) latency of the software synthesizer output driver. The default Windows wavetable synthesizer has considerable latency. I don't know how many milliseconds, but it is noticeable.

(2) latency of the digital audio output driver. Different systems have different drivers for different audio hardware. My Dell laptop has a minimum 15ms latency in the audio driver.

These two things put together were causing a very sluggish MIDI response. I was able to verify these as the culprits by routing MIDI directly out of RB into a sample player, which only introduces the latency of (2) and does not include latency of (1).

I don't know how widely known are these facts, if not then you may want to add this information to the documentation, since Windows programmers using the MIDI plugin may not know those problems, and might mistakenly blame your plugin, as I did :) Sorry about that!

(From Aaron Andrew Hunt)

20.0.225 My Xojo Web App does not launch. Why?

Plugin Version: all, Platform: macOS.

Answer: Here is a list of checks to do for linux apache installations with Xojo or Xojo Web applications:

Notes: Just a list of checks to do for linux apache installations:

- You have 64bit linux? Than you need 32 bit compatibility libraries.
- The folder of your app is writable? Set permissions to 777.
- The cgi script is executable? Set permissions to 755.

- The app file itself is executable? Set permissions to 755.
- You uploaded cgi file as text, so it has unix line endings? (this often gives error "Premature end of script headers" in apache log)
- You uploaded config.cfg file and made it writable? Set permissions to 666.
- Your apache allows execution of cgi scripts? You enabled cgi for apache and uncommented addhandler command for CGI on a new apache installation?
- You uploaded the app file and libraries as binary files? Upload as text breaks them.
- You did upload the libs folder?
- You don't have code in app.open, session.open and other events which crashes app right at launch?
- You don't have a print command in your app.open event? (see feedback case 23817)
- You allowed htaccess file to overwrite permissions?

20.0.226 SQLiteDatabase not initialized error?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Before you can use SQLiteDatabaseMBS, it must be initialized.

Example:

```
dim d as new SQLiteDatabaseMBS
```

Notes: This happens normally when you use "new SQLiteDatabaseMBS".

But if you just have a SQLConnectionMBS and get a recordset there, the initialization may not have happened, yet.

So please simply add a line "dim d as new SQLiteDatabaseMBS" to your app.open code after registration, so the plugin part can initialize and late provide recordsets.

20.0.227 Textconverter returns only the first x characters. Why?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

Some older Xojo versions limit the Textconverter to around 1024 characters in input and output. This should be fixed with RB5.

Notes:

Xojo seems not to support Textconverters at all on Windows.

20.0.228 The type translation between CoreFoundation/Foundation and Xojo data types.

Plugin Version: all, Platform: macOS.

Answer: The plugin does conversion between Cocoa/Carbon data types and native Xojo data types. The following list help you knowing what the current plugins support:

Notes: Cocoa NSObject to Variant:

```

nil ->nil
NSDictionary ->Dictionary
NSData ->MemoryBlock
NSString ->String
NSAttributedString ->NSAttributedStringMBS
NSDate ->Date
NSNumber ->double/integer/Int64/UInt64/UInt32/Boolean
NSURL ->String
NSValue with NSRect ->NSRectMBS
NSValue with NSPoint ->NSPointMBS
NSValue with NSSize ->NSSizeMBS
NSValue with NSRange ->NSRangeMBS
NSValue with QTTime ->QTTimeMBS
NSValue with QTTimeRange ->QTTimeRangeMBS
NSArray ->Array of Variant
QuartzFilter ->QuartzFilterMBS

```

- ->*MBS

Variant to Cocoa NSObject:

```

nil ->nil
Dictionary ->NSDictionary
Boolean ->NSNumber
Integer ->NSNumber
Color ->NSColor
Int64 ->NSNumber
Single ->NSNumber
Double ->NSNumber
Date ->NSDate
MemoryBlock ->NSData
String ->NSString
NSImageMBS ->NSImage
NSAttributedStringMBS ->NSAttributedString
NSColorMBS ->NSColor
NSRectMBS ->NSValue with NSRect
NSSizeMBS ->NSValue with NSSize

```

NSPointMBS ->NSValue with NSPoint
 NSRangeMBS ->NSValue with NSRange
 NSBurnMBS ->NSBurn
 NSViewMBS ->NSView
 NSFontMBS ->NSFont
 NSParagraphStyleMBS ->NSParagraphStyle
 NSAttributedStringMBS ->NSAttributedString
 WebPolicyDelegateMBS ->WebPolicyDelegate
 WebUIDelegateMBS ->WebUIDelegate
 WebFrameLoadDelegateMBS ->WebFrameLoadDelegate
 WebResourceLoadDelegateMBS ->WebResourceLoadDelegate
 NSIndexSetMBS ->NSIndexSet
 QTTimeMBS ->QTTime
 QTTimeRangeMBS ->QTTimeRange
 Array of Variant ->NSArray
 Array of String ->NSArray
 CFStringMBS ->NSString
 CFNumberMBS ->NSNumber
 CFDataMBS ->NSData
 CFURLMBS ->NSURL
 CFArrayMBS ->NSArray
 CFDictionaryMBS ->NSDictionary
 CFBinaryDataMBS ->NSData

Carbon CTypeRef to Variant:

CFDictionaryRef ->Dictionary
 CFStringRef ->String
 CFDataRef ->String
 CFURL ->String
 CFNumber ->Integer/Double/Int64
 CFArray ->Array
 CFDate ->date
 nil ->nil
 CGColorSpace ->CGColorSpaceMBS
 CGColor ->CGColorMBS
 CGImage ->CGImageMBS
 CF* ->CF*MBS

Variant to Carbon CTypeRef:

Dictionary ->CFDictionaryRef
 Boolean ->CFBooleanRef
 Color ->CFNumberRef
 Integer ->CFNumberRef

Int64 ->CFNumberRef
Single ->CFNumberRef
Double ->CFNumberRef
String ->CFStringRef
Color ->CGColorRef
Date ->CFDateRef
nil ->nil
Memoryblock ->CFDataRef
FolderItem ->CFURLRef
Dictionary ->CFDictionaryRef
Array of Variant/String/Date/Double/Single/Int64/Integer ->CFArray
CGRectMBS ->CGRect as CFDataRef
CGSizeMBS ->CGSize as CFDataRef
CGPointMBS ->CGPoint as CFDataRef
CGColorMBS ->CGColor
CGColorSpaceMBS ->CGColorSpace
CGImageMBS ->CGImage
CGDataConsumerMBS ->CGDataConsumer
CGDataProviderMBS ->CGDataProvider
CF*MBS ->CF*

Strings without encodings should be put into dictionaries as memoryblocks.

20.0.229 Uploaded my web app with FTP, but it does not run on the server!

Plugin Version: all, Platform: Windows.

Answer: If you see errors like a simple "Segmentation Fault" on Linux or some other wired errors, you may want to check your FTP upload mode. It must be binary for web apps. ASCII mode corrupts the application.

20.0.230 What classes to use for hotkeys?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use CarbonHotKeyMBS class on Mac and WindowsKeyFilterMBS on Windows.

Notes: CarbonHotKeyMBS will also work fine in Cocoa apps.

20.0.231 What do I need for Linux to get picture functions working?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: In order to get our plugins working on Linux systems without GUI, the plugin loads graphics

libraries dynamically.

Notes: To get it working, the plugin tries to load gtk with this paths:

- libgtk-x11-2.0.so”
- libgtk-x11-2.0.so.0”
- /usr/lib/libgtk-x11-2.0.so”
- /usr/lib32/libgtk-x11-2.0.so”
- /usr/lib/libgtk-x11-2.0.so.0”
- /usr/lib32/libgtk-x11-2.0.so.0”

gdk is loaded with this paths:

- libgdk-x11-2.0.so”
- libgdk-x11-2.0.so.0”
- /usr/lib/libgdk-x11-2.0.so”
- /usr/lib32/libgdk-x11-2.0.so”
- /usr/lib/libgdk-x11-2.0.so.0”
- /usr/lib32/libgdk-x11-2.0.so.0”

For the paths without explicit path, the system will search in /lib, /usr/lib and all directories in the LD_LIBRARY_PATH environment variable.

20.0.232 What does the NAN code mean?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

20.0.233 What font is used as a 'small font' in typical Mac OS X apps?

Plugin Version: all, Platform: macOS.

Answer:

Xojo 4.5 has a constant "SmallSystem" to use for a font name.

For older versions try this code:

Example:

```

Sub GetThemeFont(fontType as Integer, ByRef fontName as String, ByRef fontSize as Integer, ByRef
fontName as Integer)
dim err as Integer
dim theFont, theFontSize, theFontStyle as MemoryBlock

const smSystemScript = -1

Declare Function GetThemeFont Lib "Carbon" (inFontID as Integer, inScript as Integer, outFontName
as Ptr, outFontSize as Ptr, outStyle as Ptr) as Integer

theFont = NewMemoryBlock(256) //Str255
theFontSize = NewMemoryBlock(2) //SInt16
theFontStyle = NewMemoryBlock(1) //Style

err = GetThemeFont(fontType, smSystemScript, theFont, theFontSize, theFontStyle)

if err = 0 then
fontName = theFont.PString(0)
fontSize = theFontSize.UShort(0)
fontStyle = theFontStyle.Byte(0)
else
fontName = ""
fontSize = 0
fontStyle = 0
end if
End Sub

```

20.0.234 What is last plugin version to run on Mac OS X 10.4?

Plugin Version: all, Platform: Windows.

Answer: Last Version with 10.4 support is version 15.4.

Notes: With version 15.4 you can build applications for OS X 10.4 and newer.

For Version 16.0 we disabled 10.4 and moved minimum to 10.5. We may be able to enable it again to build a version of 16.x, but may need to charge for this by hour.

20.0.235 What is last plugin version to run on PPC?

Plugin Version: all, Platform: Windows.

Answer: Last Version with PPC is 15.4.

Notes: With version 15.4 you can build PPC applications for OS X 10.4 and newer.

For Version 16.0 we disabled PPC. We may be able to enable it again to build a PPC version of 16.x, but may need to charge for this by hour.

20.0.236 What is last version of the plugins for macOS 32-bit?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use version 23.0 or older.

Notes: We stopped including 32-bit code for macOS in version 23.1. Please use older versions if you use an old Xojo.

Xojo 2017r3 and newer load our 64-bit plugins.

20.0.237 What is the difference between Timer and WebTimer?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Timer is server side and WebTimer client side.

Notes: Timer is the normal timer class in Xojo. It runs on the server. On the client side the WebTimer runs on the client. It triggers a request to the server to perform the action. So a WebTimer is good to keep the connection running and the website updated regularly. A timer on the server is good to make regular jobs like starting a database backup every 24 hours.

20.0.238 What is the list of Excel functions?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Below is a list of function names known by LibXL.

Notes: LibXL parses the functions and writes tokens to the excel file. So even if Excel can do more functions, we can only accept the ones known by LibXL.

ABS, ABSREF, ACOS, ACOSH, ACTIVE.CELL, ADD.BAR, ADD.COMMAND, ADD.MENU, ADD.TOOLBAR, ADDRESS, AND, APP.TITLE, AREAS, ARGUMENT, ASC, ASIN, ASINH, ATAN, ATAN2, ATANH, AVEDEV, AVERAGE, AVERAGEA, BAHTTEXT, BETADIST, BETAINV, BINOMDIST, BREAK, CALL, CALLER, CANCEL.KEY, CEILING, CELL, CHAR, CHECK.COMMAND, CHIDIST, CHIINV, CHITEST, CHOOSE, CLEAN, CODE, COLUMN, COLUMNS, COMBIN, CONCATENATE, CONFIDENCE, CORREL, COS, COSH, COUNT, COUNTA, COUNTBLANK, COUNTIF, COVAR, CREATE.OBJECT, CRITBINOM, CUSTOM.REPEAT, CUSTOM.UNDO, DATE, DATEDIF, DATESTRING, DATEVALUE, DAVERAGE, DAY, DAYS360, DB, DBCS, DCOUNT, DCOUNTA, DDB, DEGREES, DELETE.BAR, DELETE.COMMAND, DELETE.MENU, DELETE.TOOLBAR, DEREf, DEVSQ, DGET, DIALOG.BOX, DIRECTORY, DMAX, DMIN, DOCUMENTS, DOLLAR, DPRODUCT, DSTDEV, DSTDEVP, DSUM, DVAR, DVARP, ECHO, ELSE, ELSE.IF, ENABLE.COMMAND, ENABLE.TOOL, END.IF, ERROR, ERROR.TYPE, EVALUATE, EVEN, EXACT, EXEC, EXECUTE, EXP, EXPONDIST, FACT, FALSE, FCLOSE, FDIST, FILES, FIND, FINDB, FINV, FISHER, FISHERINV, FIXED, FLOOR, FOPEN, FOR, FOR.CELL, FORECAST,

FORMULA.CONVERT, FPOS, FREAD, FREADLN, FREQUENCY, FSIZE, FTEST, FV, FWRITE, FWRITELN, GAMMADIST, GAMMAINV, GAMMALN, GEOMEAN, GET.BAR, GET.CELL, GET.CHART.ITEM, GET.DEF, GET.DOCUMENT, GET.FORMULA, GET.LINK.INFO, GET.MOVIE, GET.NAME, GET.NOTE, GET.OBJECT, GET.PIVOT.FIELD, GET.PIVOT.ITEM, GET.PIVOT.TABLE, GET.TOOL, GET.TOOLBAR, GET.WINDOW, GET.WORKBOOK, GET.WORKSPACE, GETPIVOTDATA, GOTO, GROUP, GROWTH, HALT, HARMEAN, HELP, HLOOKUP, HOUR, HYPERLINK, HYPGEOMDIST, IF, INDEX, INDIRECT, INFO, INITIATE, INPUT, INT, INTERCEPT, IPMT, IRR, ISBLANK, ISERR, ISERROR, ISLOGICAL, ISNA, ISNONTEXT, ISNUMBER, ISPMT, ISREF, ISTEXT, ISTHAIDIGIT, KURT, LARGE, LAST.ERROR, LEFT, LEFTB, LEN, LENB, LINEST, LINKS, LN, LOG, LOG10, LOGEST, LOGINV, LOGNORMDIST, LOOKUP, LOWER, MATCH, MAX, MAXA, MDETERM, MEDIAN, MID, MIDB, MIN, MINA, MINUTE, MINVERSE, MIRR, MMULT, MOD, MODE, MONTH, MOVIE.COMMAND, N, NA, NAMES, NEGBINOMDIST, NEXT, NORMDIST, NORMINV, NORMSDIST, NORMSINV, NOT, NOTE, NOW, NPER, NPV, NUMBERSTRING, ODD, OFFSET, OPEN.DIALOG, OPTIONS.LISTS.GET, OR, PAUSE, PEARSON, PERCENTILE, PERCENTRANK, PERMUT, PHONETIC, PI, PIVOT.ADD.DATA, PMT, POISSON, POKE, POWER, PPMT, PRESS.TOOL, PROB, PRODUCT, PROPER, PV, QUARTILE, RADIANS, RAND, RANK, RATE, REFTEXT, REGISTER, REGISTER.ID, RELREF, RENAME.COMMAND, REPLACE, REPLACEB, REPT, REQUEST, RESET.TOOLBAR, RESTART, RESULT, RESUME, RETURN, RIGHT, RIGHTB, ROMAN, ROUND, ROUNDBAHTDOWN, ROUNDBAHTUP, ROUNDDOWN, ROUNDUP, ROW, ROWS, RSQ, RTD, SAVE.DIALOG, SAVE.TOOLBAR, SCENARIO.GET, SEARCH, SEARCHB, SECOND, SELECTION, SERIES, SET.NAME, SET.VALUE, SHOW.BAR, SIGN, SIN, SINH, SKEW, SLN, SLOPE, SMALL, SPELLING.CHECK, SQRT, STANDARDIZE, STDEV, STDEVA, STDEVP, STDEVPA, STEP, STEYX, SUBSTITUTE, SUBTOTAL, SUM, SUMIF, SUMPRODUCT, SUMSQ, SUMX2MY2, SUMX2PY2, SUMXMY2, SYD, T, TAN, TANH, TDIST, TERMINATE, TEXT, TEXT.BOX, TEXTREF, THAIDAYOFWEEK, THAIDIGIT, THAIMONTHOFYEAR, THAINUMSOUND, THAINUMSTRING, THAISTRINGLENGTH, THAIYEAR, TIME, TIMEVALUE, TINV, TODAY, TRANSPOSE, TREND, TRIM, TRIMMEAN, TRUE, TRUNC, TTEST, TYPE, UNREGISTER, UPPER, USDOLLAR, USERDEFINED, VALUE, VAR, VARA, VARP, VARPA, VDB, VIEW.GET, VLOOKUP, VOLATILE, WEEKDAY, WEIBULL, WHILE, WINDOW.TITLE, WINDOWS, YEAR and ZTEST.

20.0.239 What is the replacement for PluginMBS?

Plugin Version: all, Platform: macOS.

Answer: Use the SoftDeclareMBS class to load libraries dynamically.

20.0.240 What to do on Xojo reporting a conflict?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

I get an error like "This item conflicts with another item of the same name" when using one of the plugin functions.

Xojo just wants to tell you that you dropped something in the plugins folder what is not a plugin.

Notes:

Some users dropped the examples, the documentation or other files into the plugins folder. Don't do it.

20.0.241 What to do with a NSImageCacheException?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You need to add exception handlers for NSExcptionMBS in order to catch this exception.

Notes: You may also add code to write the stack of the exception into a log file for later locating the error source.

A NSImage has several image representations in memory. So basicly you pass in the base image and for whatever size an image is needed, the NSImage class will create a cache image representation of the requested size so on the next query it can use that cache for the same requested size.

20.0.242 What to do with MySQL Error 2014?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: You can get this error on MySQL if you have a recordset open while you create another one.

20.0.243 What to do with SQL Plugin reporting Malformed string as error?

Plugin Version: all, Platform: macOS.

Answer: Please make sure the table and/or database fields have a text encoding set.

Notes: For Firebird our plugin tries to use UTF-8 encoding if possible and to correctly convert between various tables, the tables and their fields need to have a text encoding defined.

e.g. if the text field in the table is windows-1252 and the other ISO 8859-5, then the Firebird database can convert them to UTF-8 and deliver texts to the plugin.

If encoding is set to none, it may get confused for non-ascii text.

20.0.244 Where is CGGetActiveDisplayListMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetActiveDisplayList.

20.0.245 Where is CGGetDisplaysWithPointMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetDisplaysWithPoint.

20.0.246 Where is CGGetDisplaysWithRectMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetDisplaysWithRect.

20.0.247 Where is CGGetOnlineDisplayListMBS?

Plugin Version: all, Platform: Windows.

Answer: This is now CGDisplayMBS.GetOnlineDisplayList.

20.0.248 Where is GetObjectClassNameMBS?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Please use this replacement method:

Example:

```
Function GetObjectClassNameMBS(o as Object) As string
dim t as Introspection.TypeInfo = Introspection.GetType(o)
Return t.FullName
End Function
```

Notes: GetObjectClassNameMBS was removed from the plugins.

20.0.249 Where is NetworkAvailableMBS?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: We removed NetworkAvailableMBS some versions ago. It was not working right and basically it's not useful. If you want to check whether you have a network, than do a DNS resolve:

Example:

```

// two independent domain names
const domain1 = "www.google.com"
const domain2 = "www.macs.w.de"

// resolve IPs
dim ip1 as string = DNSNameToAddressMBS(Domain1)
dim ip2 as string = DNSNameToAddressMBS(Domain2)

// if we got IPs and not the same IPs (error/login pages)
if len(ip1)=0 or len(ip2)=0 or ip1=ip2 then
MsgBox "no connection"
else
MsgBox "have connection"
end if

```

Notes: This way you can detect whether you got something from DNS. And you can make sure that a DNS redirection to a login page won't catch you.

20.0.250 Where is StringHeight function in DynaPDF?

Plugin Version: all, Platform: Windows.

Answer: Use the function GetFTextHeight or GetFTextHeightEx.

Notes: Be aware that GetFTextHeight works with format commands and you may want to escape your text if you don't use them.

20.0.251 Where is XLSDocumentMBS class?

Plugin Version: all, Platform: macOS.

Answer: This class has been removed in favor of XLBookMBS class.

Notes: These classes have been removed: XLSCellMBS, XLSDocumentMBS, XLSFormatRecordMBS, XLSMergedCellsMBS, XLSRowMBS and XLSSheetMBS.

20.0.252 Where to get information about file formats?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

Please visit this web page:

<http://www.wotsit.org>

20.0.253 Where to register creator code for my application?

Plugin Version: all, Platform: macOS.

Answer:

Register at Apple:

<http://developer.apple.com/dev/cftype/information.html>

20.0.254 Which Mac OS X frameworks are 64bit only?

Plugin Version: all, Platform: macOS.

Answer: Some frameworks from Mac OS X do not support 32 bit applications, so we can't provide plugins for Xojo until 64bit target is available.

Notes: For Mac OS X 10.8:

- Accounts
- EventKit
- GLKit
- Social

and in 10.9:

- Accounts
- AVKit
- EventKit
- GameController
- GLKit
- MapKit
- MediaLibrary
- Social
- SpriteKit

In general Apple makes all new frameworks being 64 bit only.

20.0.255 Which plugins are 64bit only?

Plugin Version: all, Platform: macOS.

Answer: Some of our plugins work only in 64 bit modes as operation systems do not provide 32 bit code.

Notes: This effects currently: EventKit, Accounts, Social frameworks from Apple and our matching plugins.

20.0.256 Why application doesn't launch because of a missing ddraw.dll!?

Plugin Version: all, Platform: Windows.

Answer: Some RB versions require that you install DirectX from Microsoft on your Windows.

20.0.257 Why application doesn't launch because of a missing shlwapi.dll!?

Plugin Version: all, Platform: Windows.

Answer: Some RB versions require that you install the Internet Explorer from Microsoft on your Windows.

Notes: This bug is for several older Windows 95 editions.

20.0.258 Why do I hear a beep on keydown?

Plugin Version: all, Platform: Windows.

Answer: When the user presses a key, RB goes through all keydown event handlers till on returns true.

Notes: If no keydown event handler returns true for the key, a beep is performed.

20.0.259 Why does folderitem.item return nil?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer: Because Xojo fails to make a folderitem for you. Reason may be an alias file which can't be resolved or simply that you don't have enough access rights to read the folder content.

Notes: A more rarely reason is that the directory changed and the file with the given index or name does no longer exist.

20.0.260 Why doesn't showurl work?

Plugin Version: all, Platforms: macOS, Linux, Windows.

Answer:

There are three main reasons:

1. showurl is not supported by Xojo in 68k applications.
2. there is now application defined for the protocol (e.g. http) in the Internet Control panel.
3. You don't have Internet Config installed.

You can use the InternetConfigMBS class to check for this stuff.

20.0.261 Why don't the picture functions not work on Linux?

Plugin Version: all, Platform: macOS.

Answer: Please make sure libcairo is installed.

Notes: For accessing pictures on Linux, the MBS Plugin relays on the cairo library.

Please install the package if you don't have it already.

Our plugin looks for library called libcairo.so or libcairo.so.2.

20.0.262 Why have I no values in my chart?

Plugin Version: all, Platforms: macOS, Windows.

Answer: You have no data points visible, there may be several reasons:

Notes: For example one of the data values may be infinite or invalid.

Or the scaling may be out of range, so you simply see nothing.

20.0.263 Will application size increase with using plugins?

Plugin Version: all, Platform: Windows.

Answer: All plugins used by your application will be included in the application.

Notes: If you use no plugins, your application will not change size.

And if you use one class from the plugins, your application size will increase by a few kilobytes.

The documentation of the plugins include a list of all plugin parts and their sizes for the different platforms.

20.0.264 XLS: Custom format string guidelines

Plugin Version: all, Platform: macOS.

Answer: You have to download the source code and compile a static version of the library.

Notes: Up to four sections of format codes can be specified. The format codes, separated by semicolons, define the formats for positive numbers, negative numbers, zero values, and text, in that order. If only two sections are specified, the first is used for positive numbers and zeros, and the second is used for negative numbers. If only one section is specified, it is used for all numbers. Four sections example:

```
#,###.00_); [ Red ] (#,###.00);0.00;"sales "@
```

The following table describes the different symbols that are available for use in custom number formats.

Specify colors

To set the text color for a section of the format, type the name of one of the following eight colors in square brackets in the section. The color code must be the first item in the section.

Instead of using the name of the color, the color index can be used, like this [Color3] for Red. Valid numeric indexes for color range from 1 to 56, which reference by index to the legacy color palette.

Specify conditions

To set number formats that will be applied only if a number meets a specified condition, enclose the condition in square brackets. The condition consists of a comparison operator and a value. Comparison operators include: = Equal to; >Greater than; <Less than; >= Greater than or equal to, <= Less than or equal to, and <>Not equal to. For example, the following format displays numbers that are less than or equal to 100 in a red font and numbers that are greater than 100 in a blue font.

```
[ Red ] [ <=100 ] ; [ Blue ] [ >100 ]
```

If the cell value does not meet any of the criteria, then pound signs ("##") are displayed across the width of the cell.

Dates and times

Examples

20.0.265 Xojo doesn't work with your plugins on Windows 98.

Plugin Version: all, Platform: Windows.

Answer: Please upgrade your Windows version.

20.0.266 Xojo or my RB application itself crashes on launch on Mac OS Classic.
Why?

Plugin Version: all.

Answer:

You may check if the application has enough memory to be loaded.

RB should have on Mac OS Classic more than 20 MB of RAM.

I preferred to use 50 MB and for an application a 10 MB partition is a good way to start.

Parameter	Description
x	The x value of the data point. For an enumerated x-axis (see <code>Axis.setLabels</code> on what is an enumerated axis), the first data point is 0, and the nth data point is (n-1).
xLabel	The bottom x-axis label of the data point.
x2Label	The top x-axis label of the data point.
value	The value of the data point.
accValue	The sum of values of all data points that are in the same x position and same data group as the current data point, and with data set number less than or equal to the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
totalValue	The sum of values of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
percent	The percentage of the data point based on the total value of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
accPercent	The accumulated percentage of the data point based on the total value of all data points that are in the same x position and same data group as the current data point. This is useful for stacked charts, such as stacked bar chart and stacked area chart.
gpercent	The percentage of the data point based on the total value of all data points in a layer.
dataSet	The data set number to which the data point belongs. The first data set is 0. The nth data set is (n-1).
dataSetName	The name of the data set to which the data point belongs.
dataItem	The data point number within the data set. The first data point is 0. The nth data point is (n-1).
dataGroup	The data group number to which the data point belongs. The first data group is 0. The nth data group is (n-1).
dataGroupName	The name of the data group to which the data point belongs.
layerId	The layer number to which the data point belongs. The first layer is 0. The nth layer is (n-1).
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using <code>Layer.addExtraField</code> , <code>Layer.addExtraField2</code> , <code>BaseChart.addExtraField</code> or <code>BaseChart.addExtraField2</code> .

diFieldN	Same as fieldN. See above.
dsFieldN	Similar to fieldN, except that dsFieldN means the extra field is indexed by data set number. The Pth data set corresponds to the Pth element of the extra field.
dsdiFieldN	Similar to fieldN, except that dsdiFieldN means the extra fields are indexed by both the data set number and data point number. The Pth data item of the Qth data set corresponds to the Pth element of the (N + Q)th extra field.

Parameter	Description
zx	The symbol scale in the x dimension. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .
zy	The symbol scale in the y dimension. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .
z	The symbol scale without distinguishing the dimension to use. Applicable for layers with symbol scales set by <code>LineStyle.setSymbolScale</code> .

Parameter	Description
slope	The slope of the trend line.
intercept	The y-intercept of the trend line.
corr	The correlation coefficient in linear regression analysis.
stderr	The standard error in linear regression analysis.

Parameter	Description
top	The value of the top edge of the box-whisker symbol.
bottom	The value of the bottom edge of the box-whisker symbol.
max	The value of the maximum mark of the box-whisker symbol.
min	The value of the minimum mark of the box-whisker symbol.
med	The value of the median mark of the box-whisker symbol.

Parameter	Description
high	The high value.
low	The low value.
open	The open value.
close	The close value.

Parameter	Description
dir	The direction of the vector.
len	The length of the vector.

Parameter	Description
radius	The radial value of the data point.
value	Same as { radius } . See above.
angle	The angular value of the data point.
x	Same as { angle } . See above.
label	The angular label of the data point.
xLabel	Same as { label } . See above.
name	The name of the layer to which the data point belongs.
dataSetName	Same as { name } . See above.
i	The data point number. The first data point is 0. The nth data point is (n-1).
dataItem	Same as { i } . See above.
z	The symbol scale. Applicable for layers with symbol scales set by Polar-Layer.setSymbolScale.
fieldN	The (N + 1)th extra field. For example, { field0 } means the first extra field. An extra field is an array of custom elements added using Layer.addExtraField, Layer.addExtraField2, BaseChart.addExtraField or BaseChart.addExtraField2.
diFieldN	Same as fieldN. See above.
dsFieldN	Similar to fieldN, except that dsFieldN means the extra field is indexed by layer index. The Pth layer corresponds to the Pth element of the extra field.
dsdiFieldN	Similar to fieldN, except that dsdiFieldN means the extra fields are indexed by both the data set number and data point number. The Pth data item of the Qth layer corresponds to the Pth element of the (N + Q)th extra field.
Parameter	Description
dir	The direction of the vector.
len	The length of the vector.
Parameter	Description
value	The axis value at the tick position.
label	The axis label at the tick position.
Parameter	Description
[param]	The name of the parameter
[a]	If this field a number, it specifies the number of decimal places (digits to the right of the decimal point).

[b]

textasciitilde ' for no thousand separator. The default is 'textasciitilde ', which can be modified using `BaseChart.setNumberFormat`.

[c]

The thousand separator. Should be a non-alphanumeric character (not 0-9, A-Z, a-z). Use '.

The decimal point character. The default is '.', which can be modified using `BaseChart.setNumberFormat`.

[d]

textasciitilde ' for no negative sign character. The default is '-', which can be modified using `BaseChart.setNumberFormat`.

The negative sign character. Use '-'

Parameter	Description
yyyy	The year in 4 digits (e.g. 2002)
yyy	The year showing only the least significant 3 digits (e.g. 002 for the year 2002)
yy	The year showing only the least significant 2 digits (e.g. 02 for the year 2002)
y	The year showing only the least significant 1 digits (e.g. 2 for the year 2002)
mmm	The month formatted as its name. The default is to use the first 3 characters of the english month name (Jan, Feb, Mar ...). The names can be configured using <code>BaseChart.setMonthNames</code> .
mm	The month formatted as 2 digits from 01 - 12, adding leading zero if necessary.
m	The month formatted using the minimum number of digits from 1 - 12.
MMM	The first 3 characters of the month name converted to upper case. The names can be configured using <code>BaseChart.setMonthNames</code> .
MM	The first 2 characters of the month name converted to upper case. The names can be configured using <code>BaseChart.setMonthNames</code> .
M	The first character of the month name converted to upper case. The names can be configured using <code>BaseChart.setMonthNames</code> .
dd	The day of month formatted as 2 digits from 01 - 31, adding leading zero if necessary.
d	The day of month formatted using the minimum number of digits from 1 - 31.
w	The name of the day of week. The default is to use the first 3 characters of the english day of week name (Sun, Mon, Tue ...). The names can be configured using <code>BaseChart.setWeekDayNames</code> .
hh	The hour of day formatted as 2 digits, adding leading zero if necessary. The 2 digits will be 00 - 23 if the 'a' option (see below) is not specified, otherwise it will be 01 - 12.
h	The hour of day formatted using the minimum number of digits. The digits will be 0 - 23 if the 'a' option (see below) is not specified, otherwise it will be 01 - 12.
nn	The minute formatted as 2 digits from 00 - 59, adding leading zero if necessary.
n	The minute formatted using the minimum number of digits from 00 - 59.
ss	The second formatted as 2 digits from 00 - 59, adding leading zero if necessary.
s	The second formatted using the minimum number of digits from 00 - 59.
a	Display either 'am' or 'pm', depending on whether the time is in the morning or afternoon. The text 'am' and 'pm' can be modified using <code>BaseChart.setAMPM</code> .

Shape Id	Value	Description
SquareShape	1	Square shape. See (1, 1) above.
DiamondShape	2	Diamond shape. See (2, 1) above.
TriangleShape	3	Triangle shape pointing upwards. See (3, 1) above.
RightTriangleShape	4	Triangle shape pointing rightwards. See (4, 1) above.
LeftTriangleShape	5	Triangle shape pointing leftwards. See (5, 1) above.
InvertedTriangleShape	6	Triangle shape pointing downwards. See (1, 2) above.
CircleShape	7	Circle shape. See (2, 2) above.
StarShape	[Method]	Star shapes of various points. See (2, 3), (2, 4), (2, 5), (3, 1), (3, 2), (3, 3), (3, 4), (3, 5) above for stars with 3 to 10 points.
PolygonShape	[Method]	Polygon shapes symmetrical about a vertical axis with a vertex at the top center position. See (4, 1), (4, 3), (4, 5), (5, 1) for polygons of 5 to 8 sides.
Polygon2Shape	[Method]	Polygon shapes symmetrical about a vertical axis but without any vertex at the top center position. See (4, 2), (4, 4) for polygons of 5 and 6 sides.
CrossShape	[Method]	'+' shapes. See (5, 2), (5, 3), (5, 4), (5, 5), (6, 1), (6, 2), (6, 3) for '+' shape with arm width of 0.1 - 0.7.
Cross2Shape	[Method]	'X' shapes. See (6, 4), (6, 5), (7, 1), (7, 2), (7, 3), (7, 4), (7, 5) for 'X' shapes with arm width of 0.1 - 0.7.

langEnglish	0	Roman script
langFrench	1	Roman script
langGerman	2	Roman script
langItalian	3	Roman script
langDutch	4	Roman script
langSwedish	5	Roman script
langSpanish	6	Roman script
langDanish	7	Roman script
langPortuguese	8	Roman script
langNorwegian	9	Roman script
langHebrew	10	Hebrew script
langJapanese	11	Japanese script
langArabic	12	Arabic script
langFinnish	13	Roman script
langGreek	14	Greek script using smRoman script code
langIcelandic	15	modified smRoman/Icelandic script
langMaltese	16	Roman script
langTurkish	17	modified smRoman/Turkish script
langCroatian	18	modified smRoman/Croatian script
langTradChinese	19	Chinese (Mandarin) in traditional characters
langUrdu	20	Arabic script
langHindi	21	Devanagari script
langThai	22	Thai script
langKorean	23	Korean script

Nan	Meaning
1	Invalid square root (negative number, usually)
2	Invalid addition (indeterminate such as infinity + (-infinity))
4	Invalid division (indeterminate such as 0/0)
8	Invalid multiplication (indeterminate such as 0*infinity)
9	Invalid modulo such as (a mod 0)
17	Try to convert invalid string to a number like val("x7")
33	Invalid argument in a trig function
34	Invalid argument in an inverse trig function
36	Invalid argument in a log function
37	Invalid argument in Pow function
38	Invalid argument in toolbox financial function
40	Invalid argument in hyperbolic function
42	Invalid argument in a gamma function

Symbol	Description and result
0	Digit placeholder. For example, if the value 8.9 is to be displayed as 8.90, use the format #.00
#	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall not display extra zeros when the number typed has fewer digits on either side of the decimal than there are # symbols in the format. For example, if the custom format is #.##, and 8.9 is in the cell, the number 8.9 is displayed.
?	Digit placeholder. This symbol follows the same rules as the 0 symbol. However, the application shall put a space for insignificant zeros on either side of the decimal point so that decimal points are aligned in the column. For example, the custom format 0.0? aligns the decimal points for the numbers 8.9 and 88.99 in a column.
. (period)	Decimal point.
%	Percentage. If the cell contains a number between 0 and 1, and the custom format 0% is used, the application shall multiply the number by 100 and add the percentage symbol in the cell.
, (comma)	Thousands separator. The application shall separate thousands by commas if the format contains a comma that is enclosed by number signs (#) or by zeros. A comma that follows a placeholder scales the number by one thousand. For example, if the format is #.0,, and the cell value is 12,200,000 then the number 12.2 is displayed.
E- E+ e- e+	Scientific format. The application shall display a number to the right of the "E" symbol that corresponds to the number of places that the decimal point was moved. For example, if the format is 0.00E+00, and the value 12,200,000 is in the cell, the number 1.22E+07 is displayed. If the number format is #0.0E+0, then the number 12.2E+6 is displayed.
\$ -+/():space	Displays the symbol. If it is desired to display a character that differs from one of these symbols, precede the character with a backslash (\). Alternatively, enclose the character in quotation marks. For example, if the number format is (000), and the value 12 is in the cell, the number (012) is displayed.
\	Display the next character in the format. The application shall not display the backslash. For example, if the number format is 0\!, and the value 3 is in the cell, the value 3! is displayed.
*	Repeat the next character in the format enough times to fill the column to its current width. There shall not be more than one asterisk in one section of the format. If more than one asterisk appears in one section of the format, all but the last asterisk shall be ignored. For example, if the number format is 0*x, and the value 3 is in the cell, the value 3xxxxxx is displayed. The number of x characters that are displayed in the cell varies based on the width of the column.
_ (underline)	Skip the width of the next character. This is useful for lining up negative and positive values in different cells of the same column. For example, the number format _(0.0_);(0.0) aligns the numbers 2.3 and -4.5 in the column even though the negative number is enclosed by parentheses.
"text"	Display whatever text is inside the quotation marks. For example, the format 0.00 "dollars" displays 1.23 dollars when the value 1.23 is in the cell.
@	Text placeholder. If text is typed in the cell, the text from the cell is placed in the format where the at symbol (@) appears. For example, if the number format is "Bob "@ Smith" (including quotation marks), and the value "John" is in the cell, the value Bob John Smith is displayed.

[Black] [Green] [White] [Blue] [Magenta] [Yellow] [Cyan] [Red]

To display	As	Use this code
Months	1-12	m
Months	01-12	mm
Months	Jan-Dec	mmm
Months	January-December	mmmm
Months	J-D	mmmmm
Days	1-31	d
Days	01-31	dd
Days	Sun-Sat	ddd
Days	Sunday-Saturday	dddd
Years	00-99	yy
Years	1900-9999	yyyy
Hours	0-23	h
Hours	00-23	hh
Minutes	0-59	m
Minutes	00-59	mm
Seconds	0-59	s
Seconds	00-59	ss
Time	4 AM	h AM/PM
Time	4:36 PM	h:mm AM/PM
Time	4:36:03 P	h:mm:ss A/P
Time	4:36:03.75	h:mm:ss.00
Elapsed time	1:02	[h] :mm
Elapsed time	62:16	[mm] :ss
Elapsed time	3735.80	[ss] .00

To display	As	Use this code
1234.59	1234.6	#####.#
8.9	8.900	#.000
.631	0.6	0.#
12	12.0	#.0#
1234.568	1234.57	#.0#
44.398	44.398	???.???
102.65	102.65	???.???
2.8	2.8	???.???
5.25	5 1/4	# ??/??
5.3	5 3/10	# ??/??
12000	12,000	#,####
12000	12	#,
12400000	12.4	0.0,,