



Adobe

FEBRUARY 2008

# Adobe technology platform ActionScript™ Reference Rich Internet Application Development

Adobe® Flash® Player 9 / Adobe® Flex® 3 / Adobe® AIR™

## **Class / Interface**

● Adobe Air

■ Adobe Flex

P: Package

→ Extends

*Implements*

[Property](#)

[Static Property](#)

[Method](#)

[Static Method](#)



Table of Contents

2 About this guide

3 How to use this guide

4 – 5 Adobe technology platform for RIAs

6 – 54 ActionScript Reference Guide

Back Quick URL Reference

About This Guide

This guide provides a reference for all native ActionScript APIs for the Adobe technology platform runtimes, Adobe Flash Player and Adobe AIR, as well as the Flex framework APIs. The bulk of the reference guide contains an alphabetical listing of all of the ActionScript APIs, but also includes a number of additional references and resources.

Before you get started, you should grab the latest version of the free Adobe runtimes and frameworks from:

**Adobe Flash Player**  
<http://www.adobe.com/go/getflashplayer>

**Adobe AIR**  
<http://www.adobe.com/go/air>

**Adobe Flex**  
<http://www.adobe.com/go/flex>

How to Use This Guide

This guide can be used both as an API reference, as well as a tool to quickly learn about the ActionScript apis available within Adobe Flash Player, Adobe AIR, and the Adobe Flex framework.

To use the guide as a reference guide, just follow these steps:

Step 1 : Find the Class

All of the classes are included in alphabetical order, with the alphabetical range of each page shown in the top right of the page. Simply leaf through the guide, using the Class/Interface name as a reference, to find the item you are looking for.

Step 2 : Read the information

Each item entry contains the complete API reference for that item. Each entry is formatted to make it easy to quickly scan the item and view its information.

The entry contains a wealth of information about the item including name, type (class or interface), package name, superclass, interfaces implemented as well as a complete api listing, including static and instance properties and methods.

**Class / Interface**

● Adobe Air

■ Adobe Flex

P: Package

→ Extends

*Implements*

[Property](#)

[Static Property](#)

[Method](#)

[Static Method](#)

Lets look at the ArrayCollection entry:

**ArrayCollection** ■  
P: cd mx.collections  
→ ListCollectionView  
*IExternalizable*  
[source: Array](#)  
[ArrayCollection\(\)](#)

This item is a Class named ArrayCollection, which extends the ListCollectionView class, and implements the IExternalizable interface. It can be found in the mx.collections package, and originates in the Flex Framework and runs in Flash Player and Adobe AIR.

Understanding Where APIs can be used:

The Adobe technology platform contains two primary runtimes. One, Adobe Flash Player is browser based, and the other, Adobe AIR is desktop based. Adobe AIR is built on top of the Flash Player, so Flash Player APIs are available within Adobe AIR. In addition, this means that Adobe AIR apis are not available within the Flash Player. The Flex Framework is built on top of Flash Player APIs, and thus it runs in both the player and Adobe AIR. However, a number of Flex APIs take advantage of AIR apis, and thus only work within Adobe AIR.

Each entry contains icons to indicate where the API can be used. The icon shows where the API originates, and thus in which runtime the API will run.

**No Icon :**

Originates in Flash Player, and is available in the Flash Player, Adobe AIR and the Flex Framework.

**Red Icon Only: ●**

Originates in Adobe AIR, and cannot run in Flash Player.

**Black Icon Only : ■**

Originates in Flex Framework and runs in Flash Player and Adobe AIR.

**Black Icon and Red Icon : ■●**

Flex API that only runs in Adobe AIR.

Examples of classes from this guide:

**FileReference**

Originates in Flash Player and will run in Flash Player and Adobe AIR.

**File ●**

Originates in Adobe AIR, and will not run in Flash Player.

**Fade ■**

Originates in Flex Framework and will run in Flash Player and Adobe AIR.

**FileSystemTree ■●**

Originates in Flex Framework, but uses Adobe AIR Apis. Will run in Adobe AIR, but not in Flash Player.

Step 3 : Find more information

If you still need to find more information about an API, you can either look at the API for the subclass of the item in the reference guide, or look up the API in the online documentation. You can find the links to the online documentation on the back cover of the reference guide.

©2008 Adobe Systems Incorporated. All rights reserved. ActionScript™, Adobe® Acrobat® 9 Pro, Adobe® Flex® 3, Adobe® Media Player, Adobe® AIR™, Adobe® ColdFusion® 8, Adobe® Flash® Media Server 3, Adobe® Flex® Builder™ 3 Professional, Adobe® Dreamweaver® CS3, Adobe® Flash® CS3 Professional, Adobe® Photoshop® CS3, Adobe® Photoshop® Express, Adobe® Illustrator® CS3, Adobe® Premiere® Pro CS3, Adobe® Premiere® Express, Adobe® After Effects® CS3 Professional, Adobe® Fireworks® CS3, Adobe® Buzzword™, Adobe® Developer Connection, Adobe® Acrobat® Connect™ Pro Family, Adobe® Flash® Player 9, Adobe® LiveCycle® Enterprise Suite, Adobe® Scene7® are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.



This work is released under a Creative Commons Attribution-Noncommercial 3.0 Unported License.  
<http://creativecommons.org/licenses/by-nc/3.0/>

## Adobe technology platform for RIAs

Content and Applications

Client Framework

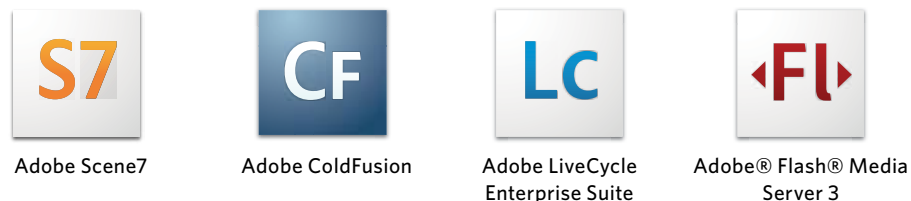
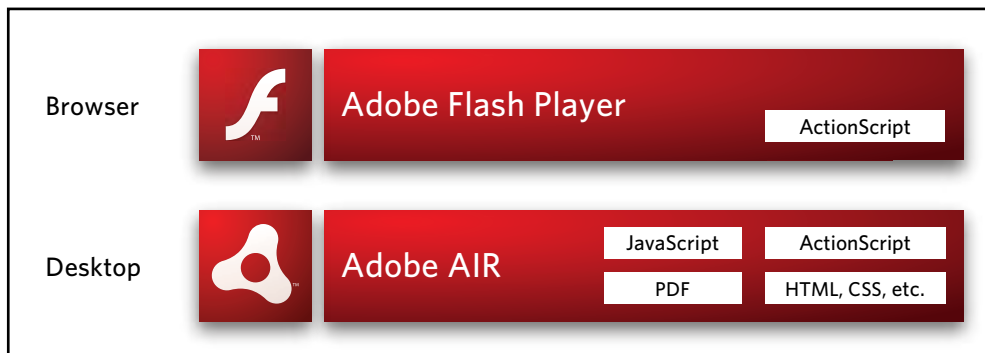
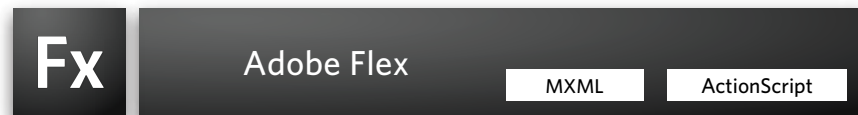
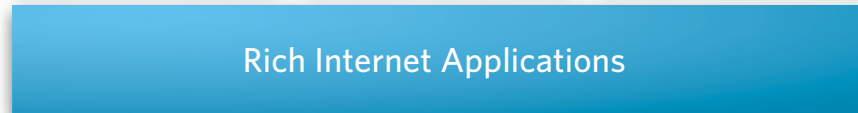
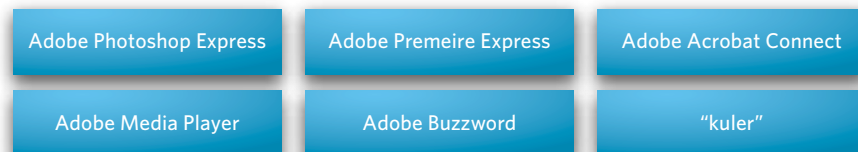
Runtimes

Operating Systems

Client

Server

Server



Tooling



**AbstractEvent** ■

P: mx.rpc.events  
→ MessageEvent  
token: AsyncToken

**AbstractConsumer** ■

P: mx.messaging  
→ MessageAgent  
resubscribeAttempts : int  
resubscribeInterval : int  
subscribed : Boolean  
timestamp : Number  
AbstractConsumer():void  
disconnect():void  
receive():void  
subscribe():void  
unsubscribe():void

**AbstractInvoker** ■

P: mx.rpc  
→ EventDispatcher  
lastResult: Object  
makeObjectsBindable: Boolean  
cancel(): AsyncToken  
clearResult(): void

**AbstractMessage** ■

P: mx.messaging.messages  
→ Object  
IMessage  
body: Object  
clientId: String  
destination: String  
DESTINATION\_CLIENT\_ID\_HEADER: String  
String  
ENDPOINT\_HEADER: String  
FLEX\_CLIENT\_ID\_HEADER: String  
headers: Object  
messageId: String  
REMOTE\_CREDENTIALS\_CHARSET\_HEADER: String  
REMOTE\_CREDENTIALS\_HEADER: String  
REQUEST\_TIMEOUT\_HEADER: String  
timestamp: Number  
timeToLive: Number  
AbstractMessage()  
toString(): String

**AbstractOperation** ■

P: mx.rpc  
→ AbstractInvoker  
arguments: Object  
name: String  
service: AbstractService  
AbstractOperation()  
send(): AsyncToken

**AbstractService** ■

P: mx.rpc  
→ Proxy  
IEventDispatcher  
channelSet: ChannelSet  
destination: String  
operations: Object  
requestTimeout: int  
AbstractService()  
disconnect(): void  
getOperation(): AbstractOperation  
logout(): void  
setCredentials(): void  
setRemoteCredentials(): void

**AbstractTarget** ■

P: mx.logging  
→ Object  
ILoggingTarget  
IMXMLObject  
filters: Array  
id: String  
level: int  
AbstractTarget()  
addLogger(): void  
initialized(): void  
logEvent(): void  
removeLogger(): void

**AbstractWebService** ■

P: mx.rpc.soap  
→ AbstractService  
DEFAULT\_DESTINATION\_HTTP: String  
DEFAULT\_DESTINATION\_HTTPS: String  
String  
description: String  
destination: String  
endpointURI: String  
headers: Array  
httpHeaders: Object  
makeObjectsBindable: Boolean  
port: String  
ready: Boolean  
rootURL: String  
service: String  
useProxy: Boolean  
xmlSpecialCharsFilter: Function  
AbstractWebService()  
addHeader(): void  
addSimpleHeader(): void  
clearHeaders(): void  
getHeader(): SOAPHeader  
removeHeader(): void  
setRemoteCredentials(): void

**Accessibility**

P: flash.accessibility  
→ Object  
active: Boolean  
updateProperties(): void

**AccessibilityProperties**

P: flash.accessibility  
→ Object  
description: String  
forceSimple: Boolean  
name: String  
noAutoLabeling: Boolean  
shortcut: String  
silent: Boolean  
AccessibilityProperties()

**Accordion** ■

P: mx.containers  
→ Container  
IHistoryManagerClient  
IFocusManagerComponent  
contentHeight: Number  
contentWidth: Number  
headerRenderer: IFactory  
historyManagementEnabled: Boolean  
resizeToContent: Boolean  
selectedChild: Container  
selectedIndex: int  
Accordion()  
getHeaderAt(): Button  
loadState(): void  
saveState(): Object

**AccordionAutomationImpl** ■

P: mx.automation.delegates.  
containers  
→ ContainerAutomationImpl  
AccordionAutomationImpl()  
init(): void  
replayAutomatableEvent(): Boolean

**AccordionHeader** ■

P: mx.containers.accordionClasses  
→ Button  
IDataRenderer  
data: Object  
AccordionHeader()

**AccordionHeaderSkin** ■

P: mx.skins.halo  
→ Border

**AcknowledgeMessage** ■

P: mx.messaging.messages  
→ AsyncMessage  
ERROR\_HINT\_HEADER: String  
AcknowledgeMessage()

**ActionEffectInstance** ■

P: mx.effects.effectClasses  
→ EffectInstance  
playedAction: Boolean  
ActionEffectInstance()  
getStartValue(): \*  
saveStartValue(): \*

**ActionScriptVersion**

P: flash.display  
→ Object  
ACTIONSCRIPT2: uint  
ACTIONSCRIPT3: uint

**ActivatorSkin** ■

P: mx.skins.halo  
→ Border

**ActivityEvent**

P: flash.events  
→ Event  
activating: Boolean  
ACTIVITY: String  
ActivityEvent()  
clone(): Event  
toString(): String

**AddChild** ■

P: mx.states  
→ Object  
creationPolicy: String  
position: String  
relativeTo: UIComponent  
target: DisplayObject  
targetFactory: IDeferredInstance  
AddChild()  
apply(): void  
createInstance(): void  
initialize(): void  
remove(): void

**AddChildAction** ■

P: mx.effects  
→ Effect  
IOVERRIDE  
index: int  
position: String  
relativeTo: DisplayObjectContainer  
AddChildAction()

**AddChildActionInstance** ■

P: mx.effects.effectClasses  
→ ActionEffectInstance  
index: int  
position: String  
relativeTo: DisplayObjectContainer  
AddChildActionInstance()

**AddItemAction** ■

P: mx.effects  
→ Effect

**AddItemActionInstance** ■

P: mx.effects.effectClasses  
→ ActionEffectInstance

**AdvancedDataGrid** ■

P: mx.controls  
→ AdvancedDataGridBaseEx  
anchorColumnIndex: int  
caretColumnIndex: int  
cellSelectionMode: Object  
displayDisclosureIcon: Boolean  
displayItemsExpanded: Boolean  
firstVisibleItem: Object  
groupedColumns: Array  
groupIconFunction: Function  
groupItemRenderer: IFactory  
groupLabelFunction: Function  
groupRowHeight: Number  
HEADER\_ICON\_PART: String  
HEADER\_TEXT\_PART: String  
hierarchicalCollectionView: IHierarchicalCollectionView  
highlightColumnIndex: int  
itemIcons: Object  
lockedColumnCount: int  
lockedRowCount: int  
movingSelectionLayer: Sprite  
rendererProviders: Array  
selectedCells: Array  
selectedColumnIndex: int  
sortExpertMode: Boolean  
treeColumn: AdvancedDataGridColumn  
treeColumnIndex: int  
treeColumnIndex: int  
tween: Object  
visibleCellRenderers: Object  
addCellSelectionData(): void  
addIndicatorToSelectionLayer(): void  
AdvancedDataGrid()  
applyCellSelectionEffect(): void  
applyUserStylesForItemRenderer(): void  
atLeastOneProperty(): Boolean  
clearCellSelectionData(): void  
clearIndicators(): void  
clearSelectedCells(): void  
collapseAll(): void  
createChildren(): void  
dragCompleteHandler(): void  
dragDropHandler(): void  
drawVerticalLine(): void  
expandAll(): void  
expandChildrenOf(): void  
expandItem(): void  
finishKeySelection(): void  
getParentItem(): \*  
initListData(): void  
isItemOpen(): Boolean  
moveIndicators(): void  
removeCellSelectionData(): void  
selectCellItem(): Boolean  
selectItem(): Boolean  
setItemIcon(): void  
treeNavigationHandler(): Boolean

**AdvancedDataGridBase** ■

P: mx.controls.advancedData  
→ AdvancedListBaseGridClasses  
columnMap: Object  
currentColNum: int  
currentItemTop: Number  
currentRowHeight: Number  
currentRowNum: int  
headerHeight: Number  
headerInfos: Array  
headerRenderer: IFactory  
headerRowInfo: Array  
headerWordWrap: Boolean  
itemRendererToFactoryMap: Dictionary  
MULTIPLE\_CELLS: String  
MULTIPLE\_ROWS: String  
NONE: String  
selectionMode: String  
showHeaders: Boolean  
SINGLE\_CELL: String  
SINGLE\_ROW: String  
sortItemRenderer: IFactory  
styleFunction: Function  
visibleHeaderInfos: Array  
AdvancedDataGridBase()  
calculateRowHeight(): Number  
clearIndicators(): void  
createHeaders(): void  
createLockedRows(): void  
drawVisibleItem(): void  
getHeaderRenderer(): IListItemRenderer  
getRowHeight(): Number  
isCellSelectionMode(): Boolean  
isRowSelectionMode(): Boolean  
makeListData(): BaseListData  
setVisibleDataItem(): void

**AdvancedDataGridAutomationImpl** ■

P: mx.automation.delegates.  
advancedDataGrid  
→ AdvancedDataGridBaseEx  
AutomationImpl  
automationTabularData: Object  
AdvancedDataGridAutomationImpl()  
init(): void

**AdvancedDataGridBaseExAutomationImpl** ■

P: mx.automation.delegates.  
advancedDataGrid  
→ AdvancedListBaseAutomationImpl  
AdvancedDataGridBaseExAutomationImpl()  
init(): void

**AdvancedDataGridBaseEx** ■

P: mx.controls  
→ AdvancedDataGridBase  
IIMESupport  
columns: Array  
draggableColumns: Boolean  
editable: String  
editedItemPosition: Object  
editedItemRenderer: IListItemRenderer  
headerInfoInitialized: Boolean  
horizontalScrollPosition: Number  
imeMode: String  
isKeyPressed: Boolean  
itemEditorInstance: IListItemRenderer  
lookAheadDuration: Number  
minColumnWidth: Number  
orderedHeadersList: Array

resizableColumns: Boolean  
sortableColumns: Boolean  
absoluteToDisplayColumnIndex(): int  
absoluteToVisibleColumnIndex(): int  
absoluteToVisibleIndices(): Object  
addSortField(): void  
AdvancedDataGridBaseEx()  
clearSeparators(): void  
colNumToIndex(): int  
createHeaderSeparators(): void  
createItemEditor(): void  
destroyItemEditor(): void  
displayToAbsoluteColumnIndex(): int  
drawColumnBackground(): void  
drawHeaderBackground(): void  
drawHorizontalLine(): void  
drawLinesAndColumnBackgrounds(): void  
drawRowBackground(): void  
drawSeparators(): void  
drawVerticalLine(): void  
findSortField(): int  
getFieldSortInfo(): SortInfo  
getSeparator(): UIComponent  
indexToColNum(): int  
isColumnFullyVisible(): Boolean  
isDataEditable(): Boolean  
moveFocusToHeader(): void  
removeSortField(): void  
scrollToViewColumn(): void  
selectColumnHeader(): void  
unselectColumnHeader(): void  
visibleToAbsoluteColumnIndex(): int

**AdvancedDataGridBaseSelectionData** ■

P: mx.controls.advanced  
→ Object  
approximate: Boolean  
columnIndex: int  
data: Object  
rowIndex: int  
AdvancedDataGridBaseSelectionData()

**AdvancedDataGridColumn** ■

P: mx.controls.advancedData  
GridClasses  
→ CSSStyleDeclaration  
IIMESupport  
dataField: String  
dataTipField: String  
dataTipFunction: Function  
editable: Boolean  
editorDataField: String  
editorHeightOffset: Number  
editorUsesEnterKey: Boolean  
editorWidthOffset: Number  
editorXOffset: Number  
editorYOffset: Number  
formatter: Formatter  
headerRenderer: IFactory  
headerText: String  
headerWordWrap: \*  
imeMode: String  
itemEditor: IFactory  
itemRenderer: IFactory  
labelFunction: Function  
minWidth: Number  
renderersEditor: Boolean  
resizable: Boolean  
showDataTips: \*  
sortable: Boolean  
sortCompareFunction: Function  
sortDescending: Boolean  
styleFunction: Function

visible: Boolean  
width: Number  
wordWrap: \*  
AdvancedDataGridColumn()  
clone(): AdvancedDataGridColumn  
itemToDataTip(): String  
itemToLabel(): String

**AdvancedDataGridColumnGroup** ■

P: mx.controls.advanced  
DataGridClasses  
→ AdvancedDataGridColumn  
children: Array  
childrenDragEnabled: Boolean  
AdvancedDataGridColumnGroup()  
itemToData(): \*

**AdvancedDataGridDragProxy** ■

P: mx.controls.advanced  
→ UIComponent  
DataGridClasses

**AdvancedDataGridEvent** ■

P: mx.events  
→ Event  
animate: Boolean  
column: AdvancedDataGridColumn  
columnSpan: int  
depth: int  
headerItem: IListItemRenderer  
index: int  
internalLabelFunction: Function  
parent: AdvancedDataGridHeaderInfo  
visible: Boolean  
visibleChildren: Array  
visibleIndex: int  
AdvancedDataGridHeaderInfo()  
AdvancedDataGridHeader  
Renderer ■  
P: mx.controls.advancedData  
GridClasses  
→ UIComponent  
IDataRenderer  
IDropInListItemRenderer  
IListItemRenderer  
data: Object  
label: IUITextField  
listData: BaseListData  
sortItemRenderer: IFactory  
AdvancedDataGridHeaderRenderer()  
getFieldSortInfo(): SortInfo  
mouseEventToHeaderPart(): String  
toolTipShowHandler(): void

**AdvancedDataGridEventReason** ■

P: mx.events  
→ Object  
CANCELLED: String  
NEW\_COLUMN: String  
NEW\_ROW: String  
OTHER: String

**AdvancedDataGridGroupItemRenderer** ■

P: mx.controls.advancedData  
GridClasses  
→ UIComponent  
IDataRenderer  
IDropInListItemRenderer  
IListItemRenderer  
IFontContextComponent  
data: Object  
disclosureIcon: IFlexDisplayObject  
icon: IFlexDisplayObject  
label: IUITextField

listData: BaseListData  
AdvancedDataGridGroupItemRenderer()

**AdvancedDataGridGroupItemRendererAutomationImpl** ■

P: mx.automation.delegates.  
advancedDataGrid  
→ UIComponentAutomationImpl  
AdvancedDataGridGroupItemRendererAutomationImpl()  
init(): void

**AdvancedDataGridHeaderHorizontalSeparator** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**AdvancedDataGridHeaderInfo** ■

P: mx.controls.advanced  
→ DataGridClasses  
actualColNum: int  
children: Array  
column: AdvancedDataGridColumn  
columnSpan: int  
depth: int  
headerItem: IListItemRenderer  
index: int  
internalLabelFunction: Function  
parent: AdvancedDataGridHeaderInfo  
visible: Boolean  
visibleChildren: Array  
visibleIndex: int  
AdvancedDataGridHeaderInfo()

**AdvancedDataGridHeaderRenderer** ■

P: mx.controls.advancedData  
GridClasses  
→ UIComponent  
IDataRenderer  
IDropInListItemRenderer  
IListItemRenderer  
data: Object  
label: IUITextField  
listData: BaseListData  
sortItemRenderer: IFactory  
AdvancedDataGridHeaderRenderer()  
getFieldSortInfo(): SortInfo  
mouseEventToHeaderPart(): String  
toolTipShowHandler(): void

**AdvancedDataGridHeaderShiftEvent** ■

P: mx.automation.events  
→ Event  
itemAutomationValue: String  
HEADER\_RELEASE: String = "header-Release"  
movingColumnIndex: int  
newColumnIndex: int  
oldColumnIndex: int  
triggerEvent: Event  
AdvancedDataGridHeaderShiftEvent()

**AdvancedDataGridItemRenderer** ■

P: mx.controls.advancedData  
GridClasses  
→ IUITextField  
IDataRenderer,  
IDropInListItemRenderer  
ILayoutManagerClient  
IListItemRenderer  
IStyleClient



data: Object  
 listData: BaseListData  
 styleDeclaration: CSSStyleDeclaration  
 AdvancedDataGridItemRenderer()  
 clearStyle(): void  
 getClassStyleDeclarations(): Array  
 getStyle(): \*  
 initProtoChain(): void  
 notifyStyleChangeInChildren(): void  
 regenerateStyleCache(): void  
 registerEffects(): void  
 setStyle(): void  
 toolTipShowHandler(): void  
 validateDisplayList(): void  
 validateProperties(): void  
 validateSize(): void

**AdvancedDataGridItemRendererAutomationImpl** ■  
 P: mx.automation.delegates  
 → UITextFieldAutomationImpl  
 advancedDataGrid  
 AdvancedDataGridItemRendererAutomationImpl()  
 init(): void

**AdvancedDataGridItemSelectEvent** ■  
 P: mx.automation.events  
 → Event  
 altKey:Boolean  
 columnIndex:int  
 ctrlKey:Boolean  
 dataField:String  
 headerPart:String  
HEADER\_RELEASE : String = “header-Release”  
 itemAutomationValue:String  
 itemRenderer:IListItemRenderer  
 shiftKey:Boolean  
 triggerEvent:Event  
 AdvancedDataGridItemSelectEvent()

**AdvancedDataGridListData** ■  
 P: mx.controls.advancedDataGridClasses  
 → DataGridListData  
 depth: int  
 disclosureIcon: Class  
 hasChildren: Boolean  
 icon: Class  
 indent: int  
 item: Object  
 open: Boolean  
 AdvancedDataGridListData()

**AdvancedDataGridRendererDescription** ■  
 P: mx.controls.advancedData  
 → Object  
 columnSpan: int  
 renderer: IFactory  
 rowSpan: int  
 AdvancedDataGridRendererDescription()

**AdvancedDataGridRendererProvider** ■  
 P: mx.controls.advancedData  
 → Object  
 IAdvancedDataGridRendererProvider  
 column: AdvancedDataGridColumn  
 columnIndex: int  
 columnSpan: int

dataField: String  
 depth: int  
 renderer: IFactory  
 AdvancedDataGridRendererProvider()  
 describeRendererForItem(): void

**AdvancedDataGridSortItemRenderer** ■  
 P: mx.controls.advancedDataGridClasses  
 → UIComponent  
 grid: AdvancedDataGrid  
 label: IUITextField  
 AdvancedDataGridSortItemRenderer()  
 getFieldSortInfo(): SortInfo  
 getFontStyles(): void

**AdvancedListBase** ■  
 P: mx.controls.listClasses  
 → ScrollControlBase  
 IDataRenderer  
 IFocusManagerComponent  
 IListItemRenderer  
 IDropInListItemRenderer  
 IEffectTargetHost

actualCollection: ICollectionView  
 actualIterator: IViewCursor  
 allowDragSelection: Boolean  
 allowMultipleSelection: Boolean  
 anchorBookmark: CursorBookmark  
 anchorIndex: int  
 cachedDataChangeEffect: Effect  
 caretBookmark: CursorBookmark  
 caretIndex: int  
 caretIndicator: Sprite  
 caretItemRenderer: IListItemRenderer  
 caretUID: String  
 collection: ICollectionView  
 columnCount: int  
 columnWidth: Number  
 data: Object  
 dataEffectCompleted: Boolean  
 dataItemWrappersByRenderer: Dictionary  
 dataProvider: Object  
 dataTipField: String  
 dataTipFunction: Function  
 defaultColumnCount: int  
 defaultRowCount: int  
 dragEnabled: Boolean  
 dragImage: UIComponent  
 dragImageOffsets: Point  
 dragMoveEnabled: Boolean  
 dropEnabled: Boolean  
 explicitColumnCount: int  
 explicitColumnWidth: Number  
 explicitRowCount: int  
 explicitRowHeight: Number  
 freeItemRenderers: Array  
 highlightIndicator: Sprite  
 highlightItemRenderer: IListItemRenderer  
 highlightUID: String  
 iconField: String  
 iconFunction: Function  
 itemRenderer: IFactory  
 itemsNeedMeasurement: Boolean  
 itemsSizeChanged: Boolean  
 iterator: IViewCursor  
 iteratorValid: Boolean  
 keySelectionPending: Boolean  
 labelField: String  
 labelFunction: Function

lastDropIndex: int  
 lastSeekPending: ListBaseSeekPending  
 listContent:  
 listData: BaseListData  
 listItems: Array  
 lockedColumnCount: int  
 lockedRowCount: int  
 menuSelectionMode: Boolean  
 modifiedCollectionView:  
 offscreenExtraColumns: int  
 offscreenExtraColumnsLeft: int  
 offscreenExtraColumnsRight: int  
 offscreenExtraRows: int  
 offscreenExtraRowsBottom: int  
 offscreenExtraRowsTop: int  
 rendererChanged: Boolean  
 reservedItemRenderers: Object  
 rowCount: int  
 rowHeight: Number  
 rowInfo: Array  
 rowMap: Object  
 runDataEffectNextUpdate: Boolean  
 runningDataEffect: Boolean  
 selectable: Boolean  
 selectedData: Object  
 selectedIndex: int  
 selectedIndices: Array  
 selectedItem: Object  
 selectedItems: Array  
 selectionIndicators: Object  
 selectionLayer: Sprite  
 selectionTwens: Object  
 showCaret: Boolean  
 showDataTips: Boolean  
 unconstrainedRenderers: Object  
 value: Object  
 variableRowHeight: Boolean  
 wordWrap: Boolean  
 wordWrapChanged: Boolean  
 addDataEffectItem(): void  
 addDragData(): void  
 addToFreeItemRenderers(): void  
 adjustListContent(): void  
 AdvancedListBase()  
 applySelectionEffect(): void  
 calculateDropIndex(): int  
 calculateDropIndicatorY(): Number  
 clearIndicators(): void  
 clearSelected(): void  
 collectionChangeHandler(): void  
 commitProperties(): void  
 configureScrollBars(): void  
 copySelectedItems(): Array  
 createChildren(): void  
 dragCompleteHandler(): void  
 dragDropHandler(): void  
 dragEnterHandler(): void  
 dragExitHandler(): void  
 dragOverHandler(): void  
 dragScroll(): void  
 dragStartHandler(): void  
 drawCaretIndicator(): void  
 drawHighlightIndicator(): void  
 drawItem(): void  
 drawRowBackgrounds(): void  
 drawSelectionIndicator(): void  
 findKey(): Boolean  
 findString(): Boolean  
 finishDataChangeEffect(): void  
 finishKeySelection(): void  
 getRendererSemanticValue(): Object  
 getReservedOrFreeItemRenderer(): IListItemRenderer

hideDropFeedback(): void  
 indexToColumn(): int  
 indexToItemRenderer(): IListItemRenderer  
 indexToRow(): int  
 indicesToIndex(): int  
 initiateDataChangeEffect(): void  
 invalidateList(): void  
 isItemHighlighted(): Boolean  
 isItemSelected(): Boolean  
 isItemVisible(): Boolean  
 isRendererUnconstrained(): Boolean  
 itemRendererContains(): Boolean  
 itemRendererToIndex(): int  
 itemRendererToIndices(): Point  
 itemToDataTip(): String  
 itemToIcon(): Class  
 itemToItemRenderer(): IListItemRenderer  
 itemToLabel(): String  
 itemToUID(): String  
 measure(): void  
 measureHeightOfItems(): Number  
 measureWidthOfItems(): Number  
 mouseClickHandler(): void  
 mouseDoubleClickHandler(): void  
 mouseDownHandler(): void  
 mouseEventToItemRenderer(): IListItemRenderer  
 mouseMoveHandler(): void  
 mouseOutHandler(): void  
 mouseOverHandler(): void  
 mouseUpHandler(): void  
 mouseWheelHandler(): void  
 moveIndicatorsHorizontally(): void  
 moveIndicatorsVertically(): void  
 moveSelectionHorizontally(): void  
 moveSelectionVertically(): void  
 prepareDataEffect(): void  
 removeDataEffectItem(): void  
 removeIndicators(): void  
 scrollHorizontally(): void  
 scrollPositionToIndex(): int  
 scrollToIndex(): Boolean  
 scrollVertically(): void  
 seekPendingFailureHandler(): void  
 seekPendingResultHandler(): void  
 selectItem(): Boolean  
 setRowCount(): void  
 setRowHeight(): void  
 showDropFeedback(): void  
 unconstrainRenderer(): void  
 updateDisplayList(): void  
 updateList(): void

**AdvancedListBaseAutomationImpl** ■  
 P: mx.automation.delegates.  
 advancedDataGrid  
 → ScrollControlBaseAutomationImpl  
 automationTabularData: Object  
 AdvancedListBaseAutomationImpl()  
 init(): void

**AIREvent** ■●  
 P: mx.events  
 → Event  
 APPLICATION\_ACTIVATE: String  
 APPLICATION\_DEACTIVATE: String  
 WINDOW\_COMPLETE: String  
 AIREvent()

**Alert** ■  
 P: mx.controls  
 → Panel  
 buttonFlags: uint  
 buttonHeight: Number  
 buttonWidth: Number  
 CANCEL: uint  
 cancelLabel: String  
 defaultButtonFlag: uint  
 iconClass: Class  
 NO: uint  
 noLabel: String  
 NONMODAL: uint  
 OK: uint  
 okLabel: String  
 text: String  
 YES: uint  
 yesLabel: String  
 Alert()  
 show(): Alert

**AlertAutomationImpl** ■  
 P: mx.automation.delegates.  
 controls  
 → PanelAutomationImpl  
 AlertAutomationImpl()  
 init(): void

**AlertFormAutomationImpl** ■  
 P: mx.automation.delegates.  
 controls  
 → UIComponentAutomationImpl  
 AlertFormAutomationImpl()  
 componentInitialized(): void  
 init(): void

**AMFChannel** ■  
 P: mx.messaging.channels  
 → PollingChannel  
 polling: Boolean  
 pollingEnabled: Boolean  
 pollingInterval: Number  
 protocol: String  
 AMFChannel()

**AnimateProperty** ■  
 P: mx.effects  
 → TweenEffect  
 fromValue: Number  
 isStyle: Boolean  
 property: String  
 roundValue: Boolean  
 toValue: Number  
 AnimateProperty()

**AnimatePropertyInstance** ■  
 P: mx.effects.effectClasses  
 → TweenEffectInstance  
 fromValue: Number  
 isStyle: Boolean  
 property: String  
 roundValue: Boolean  
 toValue: Number  
 AnimatePropertyInstance()

**AntiAliasType**  
 P: flash.text  
 → Object  
 ADVANCED: String  
 NORMAL: String

**Application** ■  
 P: mx.core  
 → LayoutContainer  
 application: Object  
 controlBar: UIComponent  
 frameRate: Number  
 historyManagementEnabled: Boolean  
 pageTitle: String  
 parameters: Object  
 preloader: Object  
 resetHistory: Boolean  
 scriptRecursionLimit: int  
 scriptTimeLimit: Number  
 url: String  
 usePreloader: Boolean  
 viewSourceURL: String  
 addToCreationQueue(): void  
 Application()

**ApplicationAutomationImpl** ■  
 P: mx.automation.delegates.  
 containers  
 → ContainerAutomationImpl  
 ApplicationAutomationImpl()  
 init(): void

**ApplicationControlBar** ■  
 P: mx.containers  
 → Object  
 dock: Boolean  
 ApplicationControlBar()

**ApplicationDomain**  
 P: flash.system  
 currentDomain: ApplicationDomain  
 parentDomain: ApplicationDomain  
 ApplicationDomain()  
 getDefinition(): Object  
 hasDefinition(): Boolean

**ApplicationTitleBarBackgroundSkin** ■●  
 P: mx.skins.halo  
 → ProgrammaticSkin

**AreaChart** ■  
 P: mx.charts  
 → CartesianChart  
 type: String  
 AreaChart()

**AreaRenderer** ■  
 P: mx.charts.renderers  
 → ProgrammaticSkin  
 IDataRenderer  
 data: Object  
 AreaRenderer()

**AreaSeries** ■  
 P: mx.charts.series  
 → Series  
 IStackable2  
 fillFunction: Function  
 horizontalAxis: IAxis  
 items: Array  
 itemType: Class  
 minField: String  
 renderDataType: Class  
 sortOnXField: Boolean  
 stacker: StackedSeries  
 stackTotals: Dictionary  
 verticalAxis: IAxis  
 xField: String  
 yField: String

applyItemRendererProperties(): void  
 AreaSeries()  
 commitProperties(): void  
 invalidateData(): void  
 invalidateMapping(): void  
 stack(): Number  
 stackAll(): Object

**AreaSeriesAutomationImpl** ■  
 P: mx.automation.delegates.charts  
 → SeriesAutomationImpl  
 AreaSeriesAutomationImpl()  
 init(): void

**AreaSeriesItem** ■  
 P: mx.charts.series.items  
 → ChartItem  
 fill: IFill  
 min: Number  
 minFilter: Number  
 minNumber: Number  
 minValue: Object  
 x: Number  
 xFilter: Number  
 xNumber: Number  
 xValue: Object  
 y: Number  
 yFilter: Number  
 yNumber: Number  
 yValue: Object  
 AreaSeriesItem()

**AreaSeriesRenderData** ■  
 P: mx.charts.series.renderData  
 → RenderData  
 element: AreaSeries  
 radius: Number  
 renderedBase: Number  
 AreaSeriesRenderData()

**AreaSeriesRenderData** ■  
 P: mx.charts.series.renderData  
 → RenderData  
 element: AreaSeries  
 radius: Number  
 renderedBase: Number  
 AreaSeriesRenderData()

**AreaSet** ■  
 P: mx.charts.series  
 → StackedSeries

**ArgumentError**  
 P: Top Level  
 → Error  
 ArgumentError()

**arguments**  
 P: Top Level  
 callee: Function  
 length: Number

**Array**  
 P: Top Level  
 → Object  
 CASEINSENSITIVE: uint  
 DESCENDING: uint  
 length: uint  
 NUMERIC: uint  
 RETURNINDEXEDARRAY: uint  
 UNIQESORT: uint  
 Array()  
 concat(): Array  
 every(): Boolean  
 filter(): Array  
 forEach(): void  
 indexOf(): int  
 join(): String  
 lastIndexOf(): int  
 map(): Array  
 pop(): \*

push(): uint  
 reverse(): Array  
 shift(): \*  
 slice(): Array  
 some(): Boolean  
 sort(): Array  
 sortOn(): Array  
 splice(): Array  
 toLocaleString(): String  
 toString(): String  
 unshift(): uint

**ArrayCollection** ■  
 P: mx.collections  
 → ListCollectionView  
 IExternalizable  
 source: Array  
 ArrayCollection()

**ArrayUtil** ■  
 P: mx.utils  
 → Object  
 getItemIndex(): int  
 toArray(): Array

**AsyncErrorEvent**  
 P: flash.events  
 → ErrorEvent  
 ASYNC\_ERROR: String  
 error: Error  
 AsyncErrorEvent()  
 clone(): Event  
 toString(): String

**AsyncMessage** ■  
 P: mx.messaging.messages  
 → AbstractMessage  
 correlationId: String  
 SUBTOPIC\_HEADER: String  
 AsyncMessage()

**AsyncRequest** ■  
 P: mx.rpc  
 → Producer  
 AsyncRequest()  
 invoke(): void

**AsyncResponder** ■  
 P: mx.rpc  
 → Object  
 IResponder  
 AsyncResponder()  
 fault(): void  
 result(): void

**AsyncToken** ■  
 P: mx.rpc  
 → EventDispatcher  
 message: IMessage  
 responders: Array  
 result: Object  
 addResponder(): void  
 AsyncToken()  
 hasResponder(): Boolean

**Automation** ■  
 P: mx.automation  
 → Object  
 automationManager: IAutomationManager  
 automationObjectHelper: IAutomationObjectHelper  
 errorShown: Boolean  
 initialized: Boolean

[mouseSimulator:IAutomationMouseSimulator](#)  
[recordedLinesCount: Number](#)  
[recordReplayLimit: Number](#)  
[restrictionNeeded: Boolean](#)  
[decrementRecordedLinesCount\(\): Number](#)  
[incrementRecordedLinesCount\(\): Number](#)  
[isLicensePresent\(\): Boolean](#)  
[registerDelegateClass\(\): void](#)

**AutomationDragEvent** ■

P: mx.automation.events  
→ MouseEvent  
action:String  
[draggedItem:IAutomationObject](#)  
[DRAG\\_COMPLETE : String = “drag-Complete”](#)  
[DRAG\\_DROP : String = “dragDrop”](#)  
[DRAG\\_START : String = “dragStart”](#)  
[dropParent:IAutomationObject](#)  
[AutomationDragEvent\(\)](#)

**AutomationDragEventWithPositionInfo** ■

P: mx.automation.events  
→ AutomationDragEvent  
[DRAG\\_COMPLETE : String = “drag-Complete”](#)  
[DRAG\\_DROP : String = “dragDrop”](#)  
[DRAG\\_START : String = “dragStart”](#)  
[AutomationDragEventWithPosition-Info\(\)](#)  
[clone\(\):Event](#)

**AutomationDPart** ■

P: mx.automation

**AutomationError** ■

P: mx.automation  
→ Error  
code: Number  
[ILLEGAL\\_OPERATION: Number](#)  
[ILLEGAL\\_RUNTIME\\_ID: Number](#)  
[OBJECT\\_NOT\\_FOUND: Number](#)  
[OBJECT\\_NOT\\_UNIQUE: Number](#)  
[OBJECT\\_NOT\\_VISIBLE: Number](#)  
[AutomationError\(\)](#)

**AutomationEvent** ■

P: mx.automation.events  
→ Event  
[BEGIN\\_RECORD : String = “beginRecord”](#)  
[END\\_RECORD : String = “endRecord”](#)  
[AutomationEvent\(\)](#)

**AutomationID** ■

P: mx.automation  
→ Object  
length: int  
addFirst(): void  
addLast(): void  
[AutomationID\(\)](#)  
concat(): AutomationID  
equals(): Boolean  
isEmpty(): Boolean  
[parse\(\): AutomationID](#)  
[peekFirst\(\): AutomationIDPart](#)  
[peekLast\(\): AutomationIDPart](#)  
[removeFirst\(\): AutomationIDPart](#)  
[removeLast\(\): AutomationIDPart](#)  
[toString\(\):String](#)

**AutomationIDPart** ■

P: mx.automation  
→ Object

**AutomationRecordEvent** ■

P: mx.automation.events  
→ Event  
args:Array  
[automationObject:IAutomationObject](#)  
[cacheable:Boolean](#)  
[name:String](#)  
[RECORD : String = “record”](#)  
[replayableEvent:Event](#)  
[AutomationRecordEvent\(\)](#)

**AutomationReplayEvent** ■

P: mx.automation.events  
→ Event  
[automationObject:IAutomationObject](#)  
[replayableEvent:Event](#)  
[REPLAY : String = “replay”](#)  
[succeeded:Boolean](#)  
[AutomationReplayEvent\(\)](#)

**AverageAggregator** ■

P: mx.olap.aggregators  
→ Object  
[IOLAPCustomAggregator](#)  
[computeBegin\(\): Object](#)  
[computeEnd\(\): Number](#)  
[computeLoop\(\): void](#)  
[computeObjectBegin\(\): Object](#)  
[computeObjectEnd\(\): Number](#)  
[computeObjectLoop\(\): void](#)

**AVM1Movie**

P: flash.display  
→ Display Object

**AxisBase** ■

→ EventDispatcher  
P: mx.charts.chartClasses  
[chartDataProvider: Object](#)  
[displayName: String](#)  
[title: String](#)  
[unitSize: Number](#)  
[AxisBase\(\)](#)  
[dataChanged\(\): void](#)  
[describeData\(\): Array](#)  
[registerDataTransform\(\): void](#)  
[unregisterDataTransform\(\): void](#)

**AxisLabel** ■

P: mx.charts  
position: Number  
text: String  
value: Object  
[AxisLabel\(\)](#)

**AxisLabelSet** ■

P: mx.charts.chartClasses  
[accurate: Boolean](#)  
labels: Array  
[minorTicks: Array](#)  
ticks: Array  
[AxisLabelSet\(\)](#)

**AxisRenderer** ■

P: mx.charts  
→ DualStyleObject  
[IAxisRenderer](#)  
axis: IAxis  
[chart: ChartBase](#)  
[gutters: Rectangle](#)

[heightLimit: Number](#)  
[highlightElements: Boolean](#)  
[horizontal: Boolean](#)  
[labelFunction: Function](#)  
[labelRenderer: IFactory](#)  
[length: Number](#)  
[minorTicks: Array](#)  
[otherAxes: Array](#)  
[placement: String](#)  
[ticks: Array](#)  
[titleRenderer: IFactory](#)  
[adjustGutters\(\): Rectangle](#)  
[AxisRenderer\(\)](#)  
[chartStateChanged\(\): void](#)  
[invalidateDisplayList\(\): void](#)  
[invalidateSize\(\): void](#)  
[measure\(\): void](#)  
[move\(\): void](#)  
[setActualSize\(\): void](#)  
[updateDisplayList\(\): void](#)

**AxisRendererAutomationImpl** ■

P: mx.automation.delegates.charts  
→ UIComponentAutomationImpl  
[AxisRendererAutomationImpl\(\)](#)  
[init\(\): void](#)

**Back** ■

P: mx.effects.easing  
[easeIn\(\): Number](#)  
[easeInOut\(\): Number](#)  
[easeOut\(\): Number](#)

**BarChart** ■

→ CartesianChart  
P: mx.charts  
type: String  
[BarChart\(\)](#)

**BarSeries** ■

→ Series  
[IStackable2](#)  
[IBar](#)  
P: mx.charts.series  
[barWidthRatio: Number](#)  
[fillFunction: Function](#)  
[horizontalAxis: IAxis](#)  
[items: Array](#)  
[itemType: Class](#)  
[labelField: String](#)  
[labelFunction: Function](#)  
[maxBarWidth: Number](#)  
[minField: String](#)  
[offset: Number](#)  
[renderDataType: Class](#)  
[stacker: StackedSeries](#)  
[stackTotals: Dictionary](#)  
[verticalAxis: IAxis](#)  
[xField: String](#)  
[yField: String](#)  
[applyItemRendererProperties\(\): void](#)  
[BarSeries\(\)](#)  
[commitProperties\(\): void](#)  
[createChildren\(\): void](#)  
[invalidateData\(\): void](#)  
[invalidateMapping\(\): void](#)  
[stack\(\): Number](#)  
[stackAll\(\): Object](#)

**BarSeriesAutomationImpl** ■

P: mx.automation.delegates.charts  
→ SeriesAutomationImpl  
[BarSeriesAutomationImpl\(\)](#)  
[init\(\): void](#)

**BarSeriesItem** ■

→ ChartItem  
P: mx.charts.series.items  
fill: IFill  
min: Number  
minFilter: Number  
minNumber: Number  
minValue: Object  
x: Number  
xFilter: Number  
xNumber: Number  
xValue: Object  
y: Number  
yFilter: Number  
yNumber: Number  
yValue: Object  
[BarSeriesItem\(\)](#)

**BarSeriesRenderData** ■

P: mx.charts.series.renderData  
→ RenderData  
[labelData: Object](#)  
[labelScale: Number](#)  
[renderedBase: Number](#)  
[renderedHalfWidth: Number](#)  
[renderedYOffset: Number](#)  
[BarSeriesRenderData\(\)](#)

**BarSet** ■

P: mx.charts.series  
→ StackedSeries  
[IBar](#)  
[barWidthRatio: Number](#)  
[maxBarWidth: Number](#)  
[offset: Number](#)  
[BarSet\(\)](#)  
[stack\(\): void](#)

**Base64Decoder** ■

P: mx.utils  
→ Object  
[Base64Decoder\(\)](#)  
[decode\(\): void](#)  
[reset\(\): void](#)  
[toArray\(\): ByteArray](#)

**Base64Encoder** ■

P: mx.utils  
→ Object  
[CHARSET\\_UTF\\_8: String](#)  
[insertNewLines: Boolean](#)  
[newLine: int](#)  
[Base64Encoder\(\)](#)  
[encode\(\): void](#)  
[encodeBytes\(\): void](#)  
[encodeUTFBytes\(\): void](#)  
[reset\(\): void](#)  
[toString\(\): String](#)

**BaseListData** ■

P: mx.controls.listClasses  
→ Object  
columnIndex: int  
label: String  
owner: UIComponent  
rowIndex: int  
uid: String  
[BaseListData\(\)](#)

**BevelFilter**

P: flash.filters  
→ BitmapFilter  
angle: Number  
[blurX: Number](#)

[blurY: Number](#)  
[distance: Number](#)  
[highlightAlpha: Number](#)  
[highlightColor: uint](#)  
[knockout: Boolean](#)  
[quality: int](#)  
[shadowAlpha: Number](#)  
[shadowColor: uint](#)  
[strength: Number](#)  
type: String  
[BevelFilter\(\)](#)  
[clone\(\): BitmapFilter](#)

**BindingUtils**

P: mx.binding.utils  
→ Object  
[bindProperty\(\): ChangeWatcher](#)  
[bindSetter\(\): ChangeWatcher](#)

**Bitmap**

P: flash.display  
→ Display Object  
[bitmapData: BitmapData](#)  
[pixelSnapping: String](#)  
[smoothing: Boolean](#)  
[Bitmap\(\)](#)

**BitmapAsset** ■

P: mx.core  
→ FlexBitmap  
[IFlexAsset](#)  
[IFlexDisplayObject](#)  
[measuredHeight: Number](#)  
[measuredWidth: Number](#)  
[BitmapAsset\(\)](#)  
[move\(\): void](#)  
[setActualSize\(\): void](#)

**BitmapData**

P: flash.display  
→ Object  
[IBitmapDrawable](#)  
height: int  
rect: Rectangle  
transparent: Boolean  
width: int  
[applyFilter\(\): void](#)  
[BitmapData\(\)](#)  
[clone\(\): BitmapData](#)  
[colorTransform\(\): void](#)  
[compare\(\): Object](#)  
[copyChannel\(\): void](#)  
[copyPixels\(\): void](#)  
[dispose\(\): void](#)  
[draw\(\): void](#)  
[fillRect\(\): void](#)  
[floodFill\(\): void](#)  
[generateFilterRect\(\): Rectangle](#)  
[getColorBoundsRect\(\): Rectangle](#)  
[getPixel\(\): uint](#)  
[getPixel32\(\): uint](#)  
[getPixels\(\): ByteArray](#)  
[hitTest\(\): Boolean](#)  
[lock\(\): void](#)  
[merge\(\): void](#)  
[noise\(\): void](#)  
[paletteMap\(\): void](#)  
[perlinNoise\(\): void](#)  
[pixelDissolve\(\): int](#)  
[scroll\(\): void](#)  
[setPixel\(\): void](#)  
[setPixel32\(\): void](#)  
[setPixels\(\): void](#)  
[threshold\(\): uint](#)  
[unlock\(\): void](#)

**BitmapDataChannel**

P: flash.display  
→ Object  
[ALPHA: uint](#)  
[BLUE: uint](#)  
[GREEN: uint](#)  
[RED: uint](#)

**BitmapFill** ■

P: mx.graphics  
→ Object  
[IFill](#)  
[offsetX: Number](#)  
[offsetY: Number](#)  
[originX: Number](#)  
[originY: Number](#)  
[repeat: Boolean](#)  
[rotation: Number](#)  
[scaleX: Number](#)  
[scaleY: Number](#)  
[smooth: Boolean](#)  
[source: Object](#)  
[BitmapFill\(\)](#)

**BitmapFilter**

P: flash.filters  
→ Object  
[clone\(\): BitmapFilter](#)

**BitmapFilterQuality**

P: flash.filters  
→ Object  
[HIGH: int](#)  
[LOW: int](#)  
[MEDIUM: int](#)

**BitmapFilterType**

P: flash.filters  
→ Object  
[FULL: String](#)  
[INNER: String](#)  
[OUTER: String](#)

**BlendMode**

P: flash.display  
→ Object  
[ADD: String](#)  
[ALPHA: String](#)  
[DARKEN: String](#)  
[DIFFERENCE: String](#)  
[ERASE: String](#)  
[HARDLIGHT: String](#)  
[INVERT: String](#)  
[LAYER: String](#)  
[LIGHTEN: String](#)  
[MULTIPLY: String](#)  
[NORMAL: String](#)  
[OVERLAY: String](#)  
[SCREEN: String](#)  
[SUBTRACT: String](#)

**Blur** ■

P: mx.effects  
→ TweenEffect  
[blurXFrom: Number](#)  
[blurXTo: Number](#)  
[blurYFrom: Number](#)  
[blurYTo: Number](#)  
[Blur\(\)](#)

**BlurFilter**

P: flash.filters  
→ BitmapFilter  
[blurX: Number](#)  
[blurY: Number](#)  
[quality: int](#)  
[BlurFilter\(\)](#)  
[clone\(\): BitmapFilter](#)

**BlurInstance** ■

P: mx.effects.effectClasses  
→ TweenEffectInstance  
[blurXFrom: Number](#)  
[blurXTo: Number](#)  
[blurYFrom: Number](#)  
[blurYTo: Number](#)  
[BlurInstance\(\)](#)

**Boolean**

P: Top Level  
→ Object  
[Boolean\(\)](#)  
[toString\(\): String](#)  
[valueOf\(\): Boolean](#)

**Border** ■

P: mx.skins  
→ ProgrammaticSkin  
[IBorder](#)  
[borderMetrics: EdgeMetrics](#)  
[Border\(\)](#)

**Bounce** ■

P: mx.effects.easing  
→ Object  
[easeIn\(\): Number](#)  
[easeInOut\(\): Number](#)  
[easeOut\(\): Number](#)

**BoundedValue** ■

P: mx.charts.chartClasses  
→ Object  
[lowerMargin: Number](#)  
[upperMargin: Number](#)  
[value: Number](#)  
[BoundedValue\(\)](#)

**Box**

P: mx.containers  
→ Container  
[direction: String](#)  
[Box\(\)](#)  
[measure\(\): void](#)  
[pixelsToPercent\(\): Number](#)  
[updateDisplayList\(\): void](#)

**BoxAutomationImpl** ■

P: mx.automation.delegates.containers  
→ ContainerAutomationImpl  
[BoxAutomationImpl\(\)](#)  
[init\(\): void](#)

**BoxDirection** ■

P: mx.containers  
→ Object  
[HORIZONTAL: String](#)  
[VERTICAL: String](#)

**BoxDivider** ■

P: mx.containers.dividedBoxClasses  
→ UIComponent

**BoxItemRenderer** ■

P: mx.charts.renderers  
→ ProgrammaticSkin  
[IDataRenderer](#)  
[data: Object](#)  
[BoxItemRenderer\(\)](#)

**BrokenImageBorderSkin** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**BrowserChangeEvent** ■

P: mx.events  
→ Event  
[APPLICATION\\_URL\\_CHANGE: String](#)  
[BROWSER\\_URL\\_CHANGE: String](#)  
[lastURL: String](#)  
[url: String](#)  
[URL\\_CHANGE: String](#)  
[BrowserChangeEvent\(\)](#)

**BrowserInvokeEvent** ●

P: flash.events  
→ Event  
arguments: Array  
[BROWSER\\_INVOKE: String](#)  
[isHTTPS: Boolean](#)  
[isUserEvent: Boolean](#)  
[sandboxType: String](#)  
[securityDomain: String](#)  
[BrowserInvokeEvent\(\)](#)  
[clone\(\): Event](#)

**BrowserManager** ■

P: mx.managers  
→ Object  
[getInstance\(\): IBrowserManager](#)

**BubbleChart** ■

P: mx.charts  
→ CartesianChart  
[radiusAxis: IAxis](#)  
[BubbleChart\(\)](#)

**BubbleSeries** ■

P: mx.charts.series  
→ Series  
[fillFunction: Function](#)  
[horizontalAxis: IAxis](#)  
[items: Array](#)  
[itemType: Class](#)  
[legendData: Array](#)  
[maxRadius: Number](#)  
[minRadius: Number](#)  
[RADIUS\\_AXIS: String](#)  
[radiusAxis: IAxis](#)  
[radiusField: String](#)  
[renderData: Object](#)  
[renderDataType: Class](#)  
[verticalAxis: IAxis](#)  
[xField: String](#)  
[yField: String](#)  
[applyItemRendererProperties\(\): void](#)  
[beginInterpolation\(\): Object](#)  
[BubbleSeries\(\)](#)  
[commitProperties\(\): void](#)  
[describeData\(\): Array](#)  
[findDataPoints\(\): Array](#)  
[getElementBounds\(\): void](#)  
[getMissingInterpolationValues\(\): void](#)  
[styleChanged\(\): void](#)  
[updateData\(\): void](#)  
[updateDisplayList\(\): void](#)  
[updateFilter\(\): void](#)  
[updateMapping\(\): void](#)  
[updateTransform\(\): void](#)

**BubbleSeriesAutomationImpl** ■

P: mx.automation.delegates.charts  
→ SeriesAutomationImpl  
BubbleSeriesAutomationImpl()  
init() : void

**BubbleSeriesItem** ■

P: mx.charts.series.items  
→ ChartItem  
fill: IFill  
x: Number  
xFilter: Number  
xNumber: Number  
xValue: Object  
y: Number  
yFilter: Number  
yNumber: Number  
yValue: Object  
z: Number  
zFilter: Number  
zNumber: Number  
zValue: Object  
BubbleSeriesItem()

**BubbleSeriesRenderData** ■

P: mx.charts.series.renderData  
→ RenderData

**Button** ■

P: mx.controls  
→ UIComponent  
IDataRenderer  
IDropInListItemRenderer  
IFocusManagerComponent  
IListItemRenderer  
IFontContextComponent  
IButton  
autoRepeat: Boolean  
data: Object  
emphasized: Boolean  
fontContext: IFlexModuleFactory  
label: String  
labelPlacement: String  
listData: BaseListData  
selected: Boolean  
selectedField: String  
stickyHighlighting: Boolean  
textField: IUITextField  
toggle: Boolean  
Button()  
clickHandler() : void  
mouseDownHandler() : void  
mouseUpHandler() : void  
rollOutHandler() : void  
rollOverHandler() : void

**ButtonAsset** ■

P: mx.core  
→ FlexSimpleButton  
IFlexAsset  
IFlexDisplayObject  
measuredHeight: Number  
measuredWidth: Number  
ButtonAsset()  
move() : void  
setActualSize() : void

**ButtonAutomationImpl** ■

P: mx.automation.delegates.controls  
→ UIComponentAutomationImpl  
ButtonAutomationImpl()  
init() : void

**ButtonBar** ■

P: mx.controls  
→ NavBar  
IFocusManagerComponent

**ButtonBarAutomationImpl** ■

P: mx.automation.delegates.controls  
→ NavBarAutomationImpl  
ButtonBarAutomationImpl()  
init() : void

**ButtonBarButtonSkin** ■

P: mx.skins.halo  
→ Border

**ButtonLabelPlacement** ■

P: mx.controls  
→ Object  
BOTTOM: String  
LEFT: String  
RIGHT: String  
TOP: String

**ButtonSkin** ■

P: mx.skins.halo  
→ Border

**ByteArray**

P: flash.utils  
→ Object  
IDataInput  
IDataOutput  
bytesAvailable: uint  
defaultObjectEncoding: uint  
endian: String  
length: uint  
objectEncoding: uint  
position: uint  
ByteArray()  
compress() : void  
readBoolean() : Boolean  
readByte() : int  
readBytes() : void  
readDouble() : Number  
readFloat() : Number  
readInt() : int  
readMultiByte() : String  
readObject() : \*  
readShort() : int  
readUnsignedByte() : uint  
readUnsignedInt() : uint  
readUnsignedShort() : uint  
readUTF() : String  
readUTFBytes() : String  
toString() : String  
uncompress() : void  
writeBoolean() : void  
writeByte() : void  
writeBytes() : void  
writeDouble() : void  
writeFloat() : void  
writeInt() : void  
writeMultiByte() : void  
writeObject() : void  
writeShort() : void  
writeUnsignedInt() : void  
writeUTF() : void  
writeUTFBytes() : void

**ByteArrayAsset** ■

P: mx.core  
→ ByteArray  
IFlexAsset

**CalendarLayoutChangeEvent** ■

P: mx.events  
→ Event  
CHANGE: String  
newDate: Date  
triggerEvent: Event  
CalendarLayoutChangeEvent()

**Camera**

P: flash.media  
→ EventDispatcher  
activityLevel: Number  
bandwidth: int  
currentFPS: Number  
fps: Number  
height: int  
index: int  
keyFrameInterval: int  
loopback: Boolean  
motionLevel: int  
motionTimeout: int  
muted: Boolean  
name: String  
names: Array  
quality: int  
width: int  
getCamera() : Camera  
setKeyFrameInterval() : void  
setLoopback() : void  
setMode() : void  
setMotionLevel() : void  
setQuality() : void

**CandlestickChart** ■

P: mx.charts  
→ CartesianChart

**CandlestickItemRenderer** ■

P: mx.charts.renderers  
→ ProgrammaticSkin  
IDataRenderer  
data: Object  
CandlestickItemRenderer()

**CandlestickSeries** ■

P: mx.charts.series  
→ HLOCSeriesBase  
fillFunction: Function  
CandlestickSeries()  
findDataPoints() : Array

**Canvas** ■

P: mx.containers  
→ Container  
IConstraintLayout  
constraintColumns: Array  
constraintRows: Array  
Canvas()  
measure() : void  
updateDisplayList() : void

**CanvasAutomationImpl** ■

P: mx.automation.delegates.containers  
→ ContainerAutomationImpl  
CanvasAutomationImpl() : void  
init() : void

**Capabilities**

P: flash.system  
→ Object  
avHardwareDisable: Boolean  
hasAccessibility: Boolean  
hasAudio: Boolean

hasAudioEncoder: Boolean  
hasEmbeddedVideo: Boolean  
hasIME: Boolean  
hasMP3: Boolean  
hasPrinting: Boolean  
hasScreenBroadcast: Boolean  
hasScreenPlayback: Boolean  
hasStreamingAudio: Boolean  
hasStreamingVideo: Boolean  
hasTLS: Boolean  
hasVideoEncoder: Boolean  
isDebugger: Boolean  
language: String  
localFileReadDisable: Boolean  
manufacturer: String  
os: String  
pixelAspectRatio: Number  
playerType: String  
screenColor: String  
screenDPI: Number  
screenResolutionX: Number  
screenResolutionY: Number  
serverString: String  
version: String

**CapsStyle**

P: flash.display  
→ Object  
NONE: String  
ROUND: String  
SQUARE: String

**CartesianCanvasValue** ■

P: mx.charts.chartClasses  
→ Object

**CartesianChart** ■

P: mx.charts.chartClasses  
→ ChartBase  
computedGutters: Rectangle  
dataRegion: Rectangle  
horizontalAxis: IAxis  
horizontalAxisRatio: Number  
horizontalAxisRenderers: Array  
selectedChartItems: Array  
CartesianChart()  
commitProperties() : void  
getFirstItem() : ChartItem  
getLastItem() : ChartItem  
getNextItem() : ChartItem  
getPreviousItem() : ChartItem  
initSecondaryMode() : void  
measure() : void  
styleChanged() : void  
updateAxisLayout() : void  
updateDisplayList() : void

**CartesianChartAutomationImpl** ■

P: mx.automation.delegates.charts  
→ ChartBaseAutomationImpl  
CartesianChartAutomationImpl()  
init() : void

**CartesianDataCanvas** ■

P: mx.charts.chartClasses  
→ ChartElement  
dataChildren: Array  
dataTransform: DataTransform  
horizontalAxis: IAxis  
includeInRanges: Boolean  
verticalAxis: IAxis

addChild() : DisplayObject  
addChildAt() : DisplayObject  
addDataChild() : void  
beginBitmapFill() : void  
beginFill() : void  
CartesianDataCanvas()  
clear() : void  
commitProperties() : void  
curveTo() : void  
dataToLocal() : Point  
describeData() : Array  
drawCircle() : void  
drawEllipse() : void  
drawRect() : void  
drawRoundedRect() : void  
endFill() : void  
invalidateData() : void  
invalidateDisplayList() : void  
lineStyle() : void  
lineTo() : void  
localToData() : Array  
mappingChanged() : void  
moveTo() : void  
removeAllChildren() : void  
removeChild() : DisplayObject  
removeChildAt() : DisplayObject  
stripNaNs() : void  
updateDataChild() : void  
updateDisplayList() : void  
updateFilter() : void  
validateData() : void  
validateTransform() : Boolean

**CartesianTransform** ■

P: mx.charts.chartClasses  
→ Data Transform  
HORIZONTAL\_AXIS: String  
pixelHeight: Number  
pixelWidth: Number  
VERTICAL\_AXIS: String  
CartesianTransform()  
invertTransform() : Array  
transformCache() : void

**CategoryAxis** ■

P: mx.charts  
→ AxisBase  
IAxis  
baseline: Number  
categoryField: String  
dataFunction: Function  
dataProvider: Object  
labelFunction: Function  
padding: Number  
ticksBetweenLabels: Boolean  
CategoryAxis()  
filterCache() : void  
formatForScreen() : String  
getLabelEstimate() : AxisLabelSet  
getLabels() : AxisLabelSet  
invertTransform() : Object  
mapCache() : void  
preferDropLabels() : Boolean  
reduceLabels() : AxisLabelSet  
transformCache() : void  
update() : void

**ChangeWatcher** ■

P: mx.binding.utils  
→ Object  
canWatch() : Boolean  
ChangeWatcher()  
getEvents() : Object  
getValue() : Object

isWatching() : Boolean  
reset() : void  
setHandler() : void  
unwatch() : void  
watch() : ChangeWatcher

**Channel** ■

P: mx.messaging  
→ EventDispatcher  
IMXMLObject  
channelSets: Array  
connected: Boolean  
connectTimeout: int  
endpoint: String  
failoverURLs: Array  
id: String  
protocol: String  
reconnecting: Boolean  
recordMessageSizes: Boolean  
recordMessageTimes: Boolean  
requestTimeout: int  
uri: String  
applySettings() : void  
Channel()  
connect() : void  
connectFailed() : void  
connectSuccess() : void  
connectTimeoutHandler() : void  
disconnect() : void  
disconnectFailed() : void  
disconnectSuccess() : void  
flexClientWaitHandler() : void  
getMessageResponder() : MessageResponder  
internalConnect() : void  
internalDisconnect() : void  
internalSend() : void  
logout() : void  
send() : void  
setCredentials() : void

**ChannelError** ■

P: mx.messaging.errors  
→ MessagingError

**ChannelEvent** ■

P: mx.messaging.events  
→ Event  
channel: Channel  
CONNECT: String  
connected: Boolean  
DISCONNECT: String  
reconnecting: Boolean  
rejected: Boolean  
ChannelEvent()  
clone() : Event  
createEvent() : ChannelEvent  
toString() : String

**ChannelFaultEvent** ■

P: mx.messaging.events  
→ ChannelEvent  
FAULT: String  
faultCode: String  
faultDetail: String  
faultString: String  
rootCause: Object  
ChannelFaultEvent()  
clone() : Event  
createErrorMessage() : ErrorMessage  
createEvent() : ChannelFaultEvent  
toString() : String

**ChannelSet** ■

P: mx.messaging  
→ EventDispatcher  
channelIds: Array  
channels: Array  
clustered: Boolean  
connected: Boolean  
currentChannel: Channel  
messageAgents: Array  
addChannel() : void  
channelConnectHandler() : void  
channelDisconnectHandler() : void  
channelFaultHandler() : void  
ChannelSet()  
connect() : void  
disconnect() : void  
logout() : void  
removeChannel() : void  
send() : void  
setCredentials() : void  
toString() : String

**ChartBase** ■

P: mx.charts.chartClasses  
→ UIComponent  
IFocusManagerComponent  
allElements: Array  
annotationElements: Array  
backgroundElements: Array  
chartState: uint  
clipContent: Boolean  
dataProvider: Object  
dataRegion: Rectangle  
dataTipFunction: Function  
dataTipLayerIndex: int  
dataTipMode: String  
description: String  
dragEnabled: Boolean  
dragImage: UIComponent  
dragMoveEnabled: Boolean  
dropEnabled: Boolean  
HORIZONTAL: String  
labelElements: Array  
legendData: Array  
mouseSensitivity: Number  
selectedChartItem: ChartItem  
selectedChartItems: Array  
selectionMode: String  
series: Array  
seriesFilters: Array  
showAllDataTips: Boolean  
showDataTips: Boolean  
VERTICAL: String  
addDragData() : void  
applySeriesSet() : Array  
ChartBase()  
clearSelection() : void  
customizeSeries() : void  
dragCompleteHandler() : void  
dragDropHandler() : void  
dragEnterHandler() : void  
dragExitHandler() : void  
dragOverHandler() : void  
dragStartHandler() : void  
findDataPoints() : Array  
getAllDataPoints() : Array  
getFirstItem() : ChartItem  
getItemsInRegion() : Array  
getLastItem() : ChartItem  
getNextItem() : ChartItem  
getPreviousItem() : ChartItem  
hideData() : void  
hideDropFeedback() : void  
invalidateChildOrder() : void

invalidateData() : void  
invalidateSeries() : void  
invalidateSeriesStyles() : void  
legendDataChanged() : void  
positionAllDataTips() : void  
positionDataTips() : void  
setActualSize() : void  
showDropFeedback() : void  
updateAllDataTips() : void

**ChartBaseAutomationImpl** ■

P: mx.automation.delegates.charts  
→ UIComponentAutomationImpl  
ChartBaseAutomationImpl()  
getLocalPoint() : Point  
init() : void  
isDragEventPositionBased() : Boolean  
keyDownHandler() : void  
keyDownHandler1() : void  
replayAutomatableEvent() : Boolean

**ChartElement** ■

P: mx.charts.chartClasses  
→ DualStyleObject  
IChartElement2  
chart: ChartBase  
chartDataProvider: Object  
cursor: IViewCursor  
dataProvider: Object  
dataTransform: DataTransform  
labelContainer: Sprite  
addChild() : DisplayObject  
addChildAt() : DisplayObject  
ChartElement()  
chartStateChanged() : void  
claimStyles() : uint  
collectTransitions() : void  
createDataID() : Number  
dataChanged() : void  
dataToLocal() : Point  
describeData() : Array  
findDataPoints() : Array  
getAllDataPoints() : Array  
localToData() : Array  
mappingChanged() : void  
processNewDataProvider() : void

**ChartEvent** ■

P: mx.charts.events  
→ MouseEvent  
CHART\_CLICK: String  
CHART\_DOUBLE\_CLICK: String  
ChartEvent()

**ChartItem** ■

P: mx.charts  
→ EventDispatcher  
currentState: String  
DISABLED: String  
element: IChartElement  
FOCUSED: String  
FOCUSEDSELECTED: String  
index: int  
item: Object  
itemRenderer: IFlexDisplayObject  
NONE: String  
ROLLOVER: String  
SELECTED: String  
ChartItem()  
clone() : ChartItem



**ChartItemDragProxy** ■  
P: mx.charts.chartClasses  
→ UIComponent

**ChartItemEvent** ■  
P: mx.charts.events  
→ MouseEvent  
[CHANGE](#): String  
[hitData](#): HitData  
[hitSet](#): Array  
[ITEM\\_CLICK](#): String  
[ITEM\\_DOUBLE\\_CLICK](#): String  
[ITEM\\_MOUSE\\_DOWN](#): String  
[ITEM\\_MOUSE\\_MOVE](#): String  
[ITEM\\_MOUSE\\_UP](#): String  
[ITEM\\_ROLL\\_OUT](#): String  
[ITEM\\_ROLL\\_OVER](#): String  
[ChartItemEvent\(\)](#)

**ChartLabel** ■  
P: mx.charts.chartClasses  
→ UIComponent  
*IDataRenderer*  
[data](#): Object  
[ChartLabel\(\)](#)  
[createChildren\(\)](#) : void  
[invalidateSize\(\)](#) : void  
[measure\(\)](#) : void  
[updateDisplayList\(\)](#) : void

**ChartSelectionChangeEvent** ■  
P: mx.automation.events  
→ Event  
[altKey](#): Boolean  
[CHANGE](#) : String = "change"  
[ctrlKey](#): Boolean  
[itemAutomationValue](#): String  
[selectionInfo](#): Array  
[shiftKey](#): Boolean  
[triggerEvent](#): Event  
[ChartSelectionChangeEvent\(\)](#)

**ChartState** ■  
P: mx.charts.chartClasses  
→ Object  
[HIDING\\_DATA](#): uint  
[NONE](#): uint  
[PREPARING\\_TO\\_HIDE\\_DATA](#): uint  
[PREPARING\\_TO\\_SHOW\\_DATA](#): uint  
[SHOWING\\_DATA](#): uint

**CheckBox** ■  
P: mx.controls  
→ Button

**CheckBoxAutomationImpl** ■  
P: mx.automation.delegates.  
controls  
→ ButtonAutomationImpl  
[CheckBoxAutomationImpl\(\)](#)  
[init\(\)](#) : void

**CheckBoxIcon** ■  
P: mx.skins.halo  
→ Border

**ChildExistenceChangedEvent** ■  
P: mx.events  
→ Event  
[CHILD\\_ADD](#): String  
[CHILD\\_REMOVE](#): String  
[OVERLAY\\_CREATED](#): String  
[relatedObject](#): DisplayObject  
[ChildExistenceChangedEvent\(\)](#)

**ChildItemPendingError** ■  
P: mx.collections.errors  
→ ItemPendingError

**CircleItemRenderer** ■  
P: mx.charts.renderers  
→ ProgrammaticSkin  
*IDataRenderer*  
[data](#): Object  
[CircleItemRenderer\(\)](#)

**Circular** ■  
P: mx.effects.easing  
→ Object  
[easeIn\(\)](#) : Number  
[easeInOut\(\)](#) : Number  
[easeOut\(\)](#) : Number

**Class**  
P: Top Level  
→ Object

**ClassFactory** ■  
P: mx.core  
→ Object  
*IFactory*  
[generator](#): Class  
[properties](#): Object  
[ClassFactory\(\)](#)  
[newInstance\(\)](#) : \*

**Clipboard** ●  
P: flash.desktop  
→ Object  
[formats](#): Array  
[generalClipboard](#): Clipboard  
[clear\(\)](#) : void  
[clearData\(\)](#) : void  
[Clipboard\(\)](#)  
[getData\(\)](#) : Object  
[hasFormat\(\)](#) : Boolean  
[setData\(\)](#) : Boolean  
[setDataHandler\(\)](#) : Boolean

**ClipboardFormats** ●  
P: flash.desktop  
→ Object  
[BITMAP\\_FORMAT](#): String  
[FILE\\_LIST\\_FORMAT](#): String  
[HTML\\_FORMAT](#): String  
[TEXT\\_FORMAT](#): String  
[URL\\_FORMAT](#): String

**ClipboardTransferMode** ●  
P: flash.desktop  
→ Object

[CLONE\\_ONLY](#): String  
[CLONE\\_PREFERRED](#): String  
[ORIGINAL\\_ONLY](#): String  
[ORIGINAL\\_PREFERRED](#): String

**CloseEvent** ■  
P: mx.events  
→ Event  
[CLOSE](#): String  
[detail](#): int  
[CloseEvent\(\)](#)

**CollectionEvent** ■  
P: mx.events  
→ Event  
[COLLECTION\\_CHANGE](#): String  
[items](#): Array  
[kind](#): String

[location](#): int  
[oldLocation](#): int  
[CollectionEvent\(\)](#)

**CollectionEventKind** ■  
P: mx.events  
→ Object  
[ADD](#): String  
[MOVE](#): String  
[REFRESH](#): String  
[REMOVE](#): String  
[REPLACE](#): String  
[RESET](#): String  
[UPDATE](#): String

**CollectionViewError**  
P: mx.collections.errors  
→ Error

**ColorMatrixFilter**  
P: flash.filters  
→ BitmapFilter  
[matrix](#): Array  
[clone\(\)](#) : BitmapFilter  
[ColorMatrixFilter\(\)](#)

**ColorPicker** ■  
P: mx.controls  
→ ComboBase  
[colorField](#): String  
[labelField](#): String  
[selectedColor](#): uint  
[selectedIndex](#): int  
[showTextField](#): Boolean  
[swatchStyleFilters](#): Object  
[close\(\)](#) : void  
[ColorPicker\(\)](#)  
[open\(\)](#) : void

**ColorPickerAutomationImpl** ■  
P: mx.automation.delegates.  
controls  
→ ComboBaseAutomationImpl  
[ColorPickerAutomationImpl\(\)](#)  
[init\(\)](#) : void

**ColorPickerEvent** ■  
P: mx.events  
→ Event  
[CHANGE](#): String  
[color](#): uint  
[ENTER](#): String  
[index](#): int  
[ITEM\\_ROLL\\_OUT](#): String  
[ITEM\\_ROLL\\_OVER](#): String  
[ColorPickerEvent\(\)](#)

**ColorPickerSkin** ■  
P: mx.skins.halo  
→ Border

**ColorTransform**  
P: flash.geom  
→ Object  
[alphaMultiplier](#): Number  
[alphaOffset](#): Number  
[blueMultiplier](#): Number  
[blueOffset](#): Number  
[color](#): uint  
[greenMultiplier](#): Number  
[greenOffset](#): Number  
[redMultiplier](#): Number  
[redOffset](#): Number  
[ColorTransform\(\)](#)

[concat\(\)](#) : void  
[toString\(\)](#) : String

**ColorUtil** ■  
P: mx.utils  
→ Object  
[adjustBrightness\(\)](#) : uint  
[adjustBrightness2\(\)](#) : uint  
[rgbMultiply\(\)](#) : uint

**ColumnChart** ■  
P: mx.charts  
→ CartesianChart  
[extendLabelToEnd](#): Boolean  
[maxLabelWidth](#): int  
[showLabelVertically](#): Boolean  
[type](#): String  
[ColumnChart\(\)](#)

**ColumnSeries** ■  
P: mx.charts.series  
→ Series  
*IColumn*  
*IStackable2*  
[columnWidthRatio](#): Number  
[fillFunction](#): Function  
[horizontalAxis](#): IAxis  
[items](#): Array  
[itemType](#): Class  
[labelField](#): String  
[labelFunction](#): Function  
[maxColumnWidth](#): Number  
[minField](#): String  
[offset](#): Number  
[renderDataType](#): Class  
[sortOnXField](#): Boolean  
[stacker](#): StackedSeries  
[stackTotals](#): Dictionary  
[verticalAxis](#): IAxis  
[xField](#): String  
[yField](#): String  
[applyItemRendererProperties\(\)](#) : void  
[ColumnSeries\(\)](#)  
[commitProperties\(\)](#) : void  
[createChildren\(\)](#) : void  
[invalidateData\(\)](#) : void  
[invalidateMapping\(\)](#) : void  
[stack\(\)](#) : Number  
[stackAll\(\)](#) : Object

**ColumnSeriesAutomationImpl** ■  
P: mx.automation.delegates.charts  
→ SeriesAutomationImpl  
[ColumnSeriesAutomationImpl\(\)](#)  
[init\(\)](#) : void

**ColumnSeriesItem** ■  
P: mx.charts.series.items  
→ ChartItem  
[fill](#): IFill  
[min](#): Number  
[minFilter](#): Number  
[minNumber](#): Number  
[minValue](#): Object  
[x](#): Number  
[xFilter](#): Number  
[xNumber](#): Number  
[xValue](#): Object  
[y](#): Number  
[yFilter](#): Number  
[yNumber](#): Number  
[yValue](#): Object  
[ColumnSeriesItem\(\)](#)

**ColumnSeriesRenderData** ■  
P: mx.charts.series.renderData  
→ RenderData  
[labelData](#): Object  
[labelScale](#): Number  
[renderedBase](#): Number  
[renderedHalfWidth](#): Number  
[renderedXOffset](#): Number  
[ColumnSeriesRenderData\(\)](#)

**ColumnSet** ■  
P: mx.charts.series  
→ StackedSeries  
*IColumn*  
[columnWidthRatio](#): Number  
[maxColumnWidth](#): Number  
[offset](#): Number  
[buildSubSeries\(\)](#) : void  
[describeData\(\)](#) : Array  
[formatDataTip\(\)](#) : String

**ComboBase** ■  
P: mx.controls  
→ UIComponent  
*IIMESupport*  
*IFocusManagerComponent*  
[arrowButtonStyleFilters](#): Object  
[borderMetrics](#): EdgeMetrics  
[collection](#): ICollectionView  
[dataProvider](#): Object  
[editable](#): Boolean  
[offset](#): Number  
[iterator](#): IViewCursor  
[restrict](#): String  
[selectedIndex](#): int  
[selectedItem](#): Object  
[text](#): String  
[textInput](#): TextInput  
[textInputStyleFilters](#): Object  
[value](#): Object  
[calculatePreferredSizeFromData\(\)](#) : Object  
[collectionChangeHandler\(\)](#) : void  
[ComboBase\(\)](#)  
[downArrowButton\\_buttonDownHandler\(\)](#) : void  
[itemToUID\(\)](#) : String  
[measure\(\)](#) : void  
[textInput\\_changeHandler\(\)](#) : void  
[updateDisplayList\(\)](#) : void

**ComboBaseAutomationImpl** ■  
P: mx.automation.delegates.  
controls  
→ UIComponentAutomationImpl  
[ComboBaseAutomationImpl\(\)](#)  
[init\(\)](#) : void

**ComboBox** ■  
P: mx.controls  
→ ComboBase  
*IDataRenderer*  
*IDropInListItemRenderer*  
*IListItemRenderer*  
[data](#): Object  
[dataProvider](#): Object  
[dropdown](#): ListBase  
[dropdownFactory](#): IFactory  
[dropdownStyleFilters](#): Object  
[dropdownWidth](#): Number  
[itemRenderer](#): IFactory  
[labelField](#): String

[labelFunction](#): Function  
[listData](#): BaseListData  
[prompt](#): String  
[rowCount](#): int  
[selectedIndex](#): int  
[selectedItem](#): Object  
[selectedLabel](#): String  
[calculatePreferredSizeFromData\(\)](#) : Object  
[close\(\)](#) : void  
[ComboBox\(\)](#)  
[itemToLabel\(\)](#) : String  
[measure\(\)](#) : void  
[open\(\)](#) : void

**ComboBoxArrowSkin** ■  
P: mx.skins.halo  
→ Border

**ComboBoxAutomationImpl** ■  
P: mx.automation.delegates.  
controls  
→ ComboBaseAutomationImpl  
[ComboBoxAutomationImpl\(\)](#)

**CommandMessage** ■  
P: mx.messaging.messages  
→ AsyncMessage  
[ADD\\_SUBSCRIPTIONS](#): String  
[AUTHENTICATION\\_MESSAGE\\_REF\\_TYPE](#): String  
[CLIENT\\_PING\\_OPERATION](#): uint  
[CLIENT\\_SYNC\\_OPERATION](#): uint  
[CLUSTER\\_REQUEST\\_OPERATION](#): uint  
[CREDENTIALS\\_CHARSET\\_HEADER](#): String  
[LOGIN\\_OPERATION](#): uint  
[LOGOUT\\_OPERATION](#): uint  
[MESSAGING\\_VERSION](#): String  
[MULTI\\_SUBSCRIBE\\_OPERATION](#): uint  
[NEEDS\\_CONFIG\\_HEADER](#): String  
[operation](#): uint  
[POLL\\_OPERATION](#): uint  
[POLL\\_WAIT\\_HEADER](#): String  
[PRESERVE\\_DURABLE\\_HEADER](#): String  
[REMOVE\\_SUBSCRIPTIONS](#): String  
[SELECTOR\\_HEADER](#): String  
[SUBSCRIBE\\_OPERATION](#): uint  
[SUBSCRIPTION\\_INVALIDATE\\_OPERATION](#): uint  
[SUBTOPIC\\_SEPARATOR](#): String  
[UNKNOWN\\_OPERATION](#): uint  
[UNSUBSCRIBE\\_OPERATION](#): uint  
[CommandMessage\(\)](#)  
[getOperationAsString\(\)](#) : String  
[toString\(\)](#) : String

**ComponentDescriptor** ■  
P: mx.core  
→ Object  
[document](#): Object  
[events](#): Object  
[id](#): String  
[properties](#): Object  
[propertiesFactory](#): Function  
[type](#): Class  
[ComponentDescriptor\(\)](#)  
[invalidateProperties\(\)](#) : void  
[toString\(\)](#) : String

**CompositeEffect** ■  
P: mx.effects  
→ Effect  
[children](#): Array  
[addChild\(\)](#) : void  
[CompositeEffect\(\)](#)

**CompositeEffectInstance** ■  
P: mx.effects.effectClasses  
→ EffectInstance  
[addChildSet\(\)](#) : void  
[CompositeEffectInstance\(\)](#)  
[onEffectEnd\(\)](#) : void  
[onTweenEnd\(\)](#) : void  
[onTweenUpdate\(\)](#) : void

**CompressionAlgorithm** ●  
P: flash.utils  
→ Object  
[DEFLATE](#): String  
[ZLIB](#): String

**Concurrency** ■  
P: mx.rpc.mx.xml  
→ Object  
[LAST](#): String  
[MULTIPLE](#): String  
[SINGLE](#): String

**ConfigMap** ■  
P: mx.messaging.config  
→ Proxy  
[propertyList](#): Array  
[callProperty\(\)](#) : \*  
[ConfigMap\(\)](#)  
[deleteProperty\(\)](#) : Boolean  
[getProperty\(\)](#) : \*  
[hasProperty\(\)](#) : Boolean  
[nextName\(\)](#) : String  
[nextNameIndex\(\)](#) : int  
[nextValue\(\)](#) : \*  
[setProperty\(\)](#) : void

**ConstraintColumn** ■  
P: mx.containers.utilityClasses  
→ EventDispatcher  
*IMXMLObject*  
[explicitWidth](#): Number  
[id](#): String  
[maxWidth](#): Number  
[minWidth](#): Number  
[percentWidth](#): Number  
[width](#): Number  
[ConstraintColumn\(\)](#)  
[initialized\(\)](#) : void  
[setActualWidth\(\)](#) : void

**ConstraintRow** ■  
P: mx.containers.utilityClasses  
→ EventDispatcher  
*IMXMLObject*  
[explicitHeight](#): Number  
[height](#): Number  
[id](#): String  
[maxHeight](#): Number  
[minHeight](#): Number  
[percentHeight](#): Number  
[ConstraintRow\(\)](#)  
[initialized\(\)](#) : void  
[setActualHeight\(\)](#) : void

**Consumer** ■  
P: mx.messaging  
→ AbstractConsumer  
[selector](#): String  
[subtopic](#): String  
[Consumer\(\)](#)

**Container** ■  
P: mx.core  
→ UIComponent  
*IContainer*  
*IDataRenderer*  
*IFocusManagerContainer*  
*IListItemRenderer*  
*IRawChildrenContainer*  
[actualCreationPolicy](#): String  
[autoLayout](#): Boolean  
[borderMetrics](#): EdgeMetrics  
[childDescriptors](#): Array  
[clipContent](#): Boolean  
[contentMouseX](#): Number  
[contentMouseY](#): Number  
[creatingContentPane](#): Boolean  
[creationIndex](#): int  
[creationPolicy](#): String  
[data](#): Object  
[defaultButton](#): IFlexDisplayObject  
[horizontalLineScrollSize](#): Number  
[horizontalPageScrollSize](#): Number  
[horizontalScrollBar](#): ScrollBar  
[horizontalScrollPolicy](#): String  
[horizontalScrollPosition](#): Number  
[icon](#): Class  
[label](#): String  
[maxHorizontalScrollPosition](#): Number  
[maxVerticalScrollPosition](#): Number  
[numChildren](#): int  
[rawChildren](#): IChildList  
[verticalLineScrollSize](#): Number  
[verticalPageScrollSize](#): Number  
[verticalScrollBar](#): ScrollBar  
[verticalScrollPolicy](#): String  
[verticalScrollPosition](#): Number  
[viewMetrics](#): EdgeMetrics  
[viewMetricsAndPadding](#): EdgeMetrics  
[addChild\(\)](#) : DisplayObject  
[addChildAt\(\)](#) : DisplayObject  
[attachOverlay\(\)](#) : void  
[Container\(\)](#)  
[contentToGlobal\(\)](#) : Point  
[contentToLocal\(\)](#) : Point  
[createBorder\(\)](#) : void  
[createComponentFromDescriptor\(\)](#) : IFlexDisplayObject  
[createComponentsFromDescriptors\(\)](#) : void  
[executeBindings\(\)](#) : void  
[executeChildBindings\(\)](#) : void  
[getChildAt\(\)](#) : DisplayObject  
[getChildByName\(\)](#) : DisplayObject  
[getChildIndex\(\)](#) : int  
[getChildren\(\)](#) : Array  
[globalToContent\(\)](#) : Point  
[layoutChrome\(\)](#) : void  
[localToContent\(\)](#) : Point  
[removeAllChildren\(\)](#) : void  
[removeChild\(\)](#) : DisplayObject  
[removeChildAt\(\)](#) : DisplayObject  
[scrollChildren\(\)](#) : void  
[setChildIndex\(\)](#) : void  
[updateDisplayList\(\)](#) : void



**ContainerAutomationImpl** ■

P: UIComponentAutomationImpl  
→ UIComponentAutomationImpl  
ContainerAutomationImpl()  
init() : void

**ContainerCreationPolicy** ■

P: mx.core  
→ Object  
[ALL: String](#)  
[AUTO: String](#)  
[NONE: String](#)  
[QUEUED: String](#)

**ContainerLayout** ■

P: mx.core  
→ Object  
[ABSOLUTE: String](#)  
[HORIZONTAL: String](#)  
[VERTICAL: String](#)

**ContainerMovieClip** ■

P: mx.flash  
→ UIMovieClip  
content: IUIComponent  
ContainerMovieClip()

**ContainerMovieClipAutomationImpl** ■

P: mx.automation.delegates.  
flashflexkit  
→ UIMovieClipAutomationImpl  
ContainerMovieClipAutomationImpl()  
init():void

**ContextMenu**

P: flash.ui  
→ NativeMenu  
builtInItems: ContextMenuBuiltInItems  
customItems: Array  
ContextMenu()  
display() : void  
hideBuiltInItems() : void

**ContextMenuBuiltInItems**

P: flash.ui  
→ Object  
forwardAndBack: Boolean  
loop: Boolean  
play: Boolean  
print: Boolean  
quality: Boolean  
rewind: Boolean  
save: Boolean  
zoom: Boolean  
ContextMenuBuiltInItems()

**ContextMenuEvent**

P: flash.events  
→ Event  
contextMenuOwner: InteractiveObject  
[MENU\\_ITEM\\_SELECT: String](#)  
[MENU\\_SELECT: String](#)  
mouseTarget: InteractiveObject  
clone() : Event  
ContextMenuEvent()  
toString() : String

**ContextMenuItem**

P: flash.ui  
→ NativeMenuItem  
caption: String  
separatorBefore: Boolean  
visible: Boolean  
ContextMenuItem()

**ContextualClassFactory** ■

P: mx.core  
→ ClassFactory  
moduleFactory: IFlexModuleFactory  
ContextualClassFactory()  
newInstance() : \*

**ControlBar** ■

P: mx.containers  
→ Box

**ConvolutionFilter**

P: flash.filters  
→ BitmapFilter  
alpha: Number  
bias: Number  
clamp: Boolean  
color: uint  
divisor: Number  
matrix: Array  
matrixX: Number  
matrixY: Number  
preserveAlpha: Boolean  
clone() : BitmapFilter  
ConvolutionFilter()

**CountAggregator** ■

P: mx.olap.aggregators  
→ Object  
IOLAPCustomAggregator  
computeBegin() : Object  
computeEnd() : Number  
computeLoop() : void  
computeObjectBegin() : Object  
computeObjectEnd() : Number  
computeObjectLoop() : void

**CreditCardValidator** ■

P: mx.validators  
→ Validator  
allowedFormatChars: String  
cardNumberListener: IValidatorListener  
cardNumberProperty: String  
cardNumberSource: Object  
cardTypeListener: IValidatorListener  
cardTypeProperty: String  
cardTypeSource: Object  
invalidCharError: String  
invalidNumberError: String  
noNumError: String  
noTypeError: String  
wrongLengthError: String  
wrongTypeError: String  
CreditCardValidator()  
doValidation() : Array  
validateCreditCard() : Array

**CreditCardValidatorCardType** ■

P: mx.validators  
→ Object  
[AMERICAN\\_EXPRESS: String](#)  
[DINERS\\_CLUB: String](#)  
[DISCOVER: String](#)  
[MASTER\\_CARD: String](#)  
[VISA: String](#)

**CrossItemRenderer** ■

P: mx.charts.renderers  
→ ProgrammaticSkin  
IDataRenderer  
data: Object  
thickness: Number  
CrossItemRenderer()

**CSMSettings**

P: flash.text  
→ Object  
fontSize: Number  
insideCutoff: Number  
outsideCutoff: Number  
CSMSettings()

**CSSStyleDeclaration** ■

P: mx.styles  
→ EventDispatcher  
defaultFactory: Function  
factory: Function  
overrides: Object  
clearStyle() : void  
CSSStyleDeclaration()  
getStyle() : \*  
setStyle() : void

**CubeEvent** ■

P: mx.events  
→ Event  
[CUBE\\_COMPLETE: String](#)  
[CUBE\\_PROGRESS: String](#)  
message: String  
progress: int  
[QUERY\\_PROGRESS: String](#)  
total: int  
CubeEvent() : void

**Cubic** ■

P: mx.effects.easing  
→ Object  
easeIn() : Number  
easeInOut() : Number  
easeOut() : Number

**CuePointEvent** ■

P: mx.events  
→ Event  
[CUE\\_POINT: String](#)  
cuePointName: String  
cuePointTime: Number  
cuePointType: String  
CuePointEvent()

**CuePointManager** ■

P: mx.controls.videoClasses  
→ Object  
addCuePoint() : Object  
CuePointManager()  
getCuePointByName() : Object  
getCuePoints() : Array  
removeAllCuePoints() : void  
removeCuePoint() : Object  
setCuePoints() : void

**CurrencyFormatter** ■

P: mx.flash  
→ Formatter  
alignSymbol: String  
currencySymbol: String  
decimalSeparatorFrom: String  
decimalSeparatorTo: String  
precision: Object  
rounding: String  
thousandsSeparatorFrom: String  
thousandsSeparatorTo: String  
useNegativeSign: Object  
useThousandsSeparator: Object  
CurrencyFormatter()  
format() : String

**CurrencyValidator** ■

P: mx.validators  
→ Validator  
alignSymbol: String  
allowNegative: Object  
currencySymbol: String  
currencySymbolError: String  
decimalPointCountError: String  
decimalSeparator: String  
exceedsMaxError: String  
invalidCharError: String  
invalidFormatCharsError: String  
lowerThanMinError: String  
maxValue: Object  
minValue: Object  
negativeError: String  
precision: Object  
precisionError: String  
separationError: String  
thousandsSeparator: String  
CurrencyValidator()  
doValidation() : Array  
validateCurrency() : Array

**CurrencyValidatorAlignSymbol** ■

P: mx.validators  
→ Object  
ANY: String  
LEFT: String  
RIGHT: String

**CursorBookmark** ■

P: mx.collections  
→ Object  
CURRENT: CursorBookmark  
FIRST: CursorBookmark  
LAST: CursorBookmark  
value: Object  
CursorBookmark()  
getViewIndex() : int

**CursorError**

→ Error  
CursorError() : void

**CursorManager** ■

P: mx.managers  
→ Object  
currentCursorID: int  
currentCursorXOffset: Number  
currentCursorYOffset: Number  
NO\_CURSOR: int  
getInstance()  
hideCursor() : void  
removeAllCursors() : void  
removeBusyCursor() : void  
removeCursor() : void  
setBusyCursor() : void  
setCursor() : int  
showCursor() : void

**CursorManagerPriority** ■

P: mx.managers  
→ Object  
HIGH: int  
LOW: int  
MEDIUM: int

**DataDescription** ■

P: mx.charts.chartClasses  
→ Object  
boundedValues: Array  
max: Number  
min: Number

minInterval: Number  
padding: Number  
[REQUIRED\\_BOUNDED\\_VALUES: uint](#)  
[REQUIRED\\_MIN\\_INTERVAL: uint](#)  
[REQUIRED\\_MIN\\_MAX: uint](#)  
[REQUIRED\\_PADDING: uint](#)  
DataDescription()

**DataEvent**

P: flash.events  
→ TextEvent  
DATA: String  
data: String  
[UPLOAD\\_COMPLETE\\_DATA: String](#)  
clone() : Event  
DataEvent()  
toString() : String

**DataGrid** ■

P: mx.controls  
→ DataGridBase  
columns: Array  
draggableColumns: Boolean  
editable: Boolean  
editedItemPosition: Object  
editedItemRenderer: IListItemRenderer  
horizontalScrollPosition: Number  
imeMode: String  
itemEditorInstance: IListItemRenderer  
minColumnWidth: Number  
resizableColumns: Boolean  
sortableColumns: Boolean  
adjustListContent() : void  
clearSeparators() : void  
createItemEditor() : void  
DataGrid()  
destroyItemEditor() : void  
drawColumnBackground() : void  
drawHeaderBackground() : void  
drawHorizontalLine() : void  
drawLinesAndColumnBackgrounds() : void  
drawLinesAndColumnGraphics() : void  
drawRowBackground() : void  
drawSeparators() : void  
drawVerticalLine() : void  
isItemEditable() : Boolean  
placeSortArrow() : void  
scrollPositionToIndex() : int  
scrollVertically() : void

**DataGridAutomationImpl** ■

P: mx.automation.delegates.  
controls  
→ ComboBoxAutomationImpl  
IIMESupport  
automationTabularData: Object  
DataGridAutomationImpl()  
init() : void

**DataGridBase** ■

P: mx.controls.dataGridClasses  
→ ListBase  
IFontContextComponent  
columnCaretIndicator: Sprite  
columnHighlightIndicator: Sprite  
columnMap: Object  
fontContext: IFlexModuleFactory  
freeItemRenderersTable: Dictionary  
header: DataGridHeaderBase  
headerHeight: Number  
headerMask: Shape  
lockedColumnAndRowContent: DataGridLockedRowContentHolder

lockedColumnContent: ListBaseContentHolder  
lockedColumnCount: int  
lockedColumnHeader: DataGridHeaderBase  
lockedRowContent: DataGridLockedRowContentHolder  
lockedRowCount: int  
showHeaders: Boolean  
adjustRow() : void  
calculateRowHeight() : Number  
clearRow() : void  
createColumnItemRenderer() : IListItemRenderer  
DataGridBase()  
drawVisibleItem() : void  
finishKeySelection() : void  
itemRendererToIndices() : Point  
layoutColumnItemRenderer() : Point  
makeListData() : BaseListData  
makeRow() : Number  
moveSelectionVertically() : void  
prepareRowArray() : void  
removeExtraRow() : void  
setRowInfo() : void  
setupColumnItemRenderer() : IListItemRenderer  
updateRendererDisplayList() : void

**DataGridColumn** ■

P: mx.controls.dataGridClasses  
→ CSSStyleDeclaration  
IIMESupport  
dataField: String  
dataTipField: String  
dataTipFunction: Function  
draggable: Boolean  
editable: Boolean  
editorDataField: String  
editorHeightOffset: Number  
editorUsesEnterKey: Boolean  
editorWidthOffset: Number  
editorXOffset: Number  
editorYOffset: Number  
headerRenderer: IFactory  
headerText: String  
headerWordWrap: \*  
imeMode: String  
itemEditor: IFactory  
itemRenderer: IFactory  
labelFunction: Function  
minWidth: Number  
nullItemRenderer: IFactory  
renderersEditor: Boolean  
resizable: Boolean  
showDataTips: \*  
sortable: Boolean  
sortCompareFunction: Function  
sortDescending: Boolean  
visible: Boolean  
width: Number  
wordWrap: \*  
DataGridColumn()  
getItemRendererFactory() : IFactory  
itemToDataTip() : String  
itemToLabel() : String  
UIComponent

**DataGridColumnDropIndicator** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**DataGridColumnResizeSkin** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**DataGridDragProxy** ■

P: mx.controls.dataGridClasses  
→ UIComponent  
DataGridDragProxy()  
measure():void

**DataGridEvent** ■

P: mx.events  
→ Event  
[COLUMN\\_STRETCH: String](#)  
columnIndex: int  
dataField: String  
[HEADER\\_RELEASE: String](#)  
[ITEM\\_EDIT\\_BEGIN: String](#)  
[ITEM\\_EDIT\\_BEGINNING: String](#)  
[ITEM\\_EDIT\\_END: String](#)  
[ITEM\\_FOCUS\\_IN: String](#)  
[ITEM\\_FOCUS\\_OUT: String](#)  
itemRenderer: IListItemRenderer  
localX: Number  
reason: String  
rowIndex: int  
DataGridEvent()

**DataGridEventReason** ■

P: mx.events  
→ Object  
[CANCELLED: String](#)  
[NEW\\_CsOLUMN: String](#)  
[NEW\\_ROW: String](#)  
[OTHER: String](#)

**DataGridHeader** ■

P: mx.controls.dataGridClasses  
→ DataGridHeaderBase  
bottomOffset: Number  
cachedHeaderHeight: Number  
cachedPaddingBottom: Number  
cachedPaddingTop: Number  
dataGrid: DataGrid  
enabled: Boolean  
headerItems: Array  
leftOffset: Number  
needRightSeparator: Boolean  
needRightSeparatorEvents: Boolean  
rightOffset: Number  
topOffset: Number  
clearSeparators() : void  
createChildren() : void  
DataGridHeader()  
drawColumnDragOverlay() : void  
drawHeaderBackground() : void  
drawHeaderIndicator() : void  
drawSelectionIndicator() : void  
drawSeparators() : void  
measure() : void  
placeSortArrow() : void  
updateDisplayList() : void

**DataGridHeaderBackgroundSkin** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**DataGridHeaderBase** ■

P: mx.controls.dataGridClasses  
→ UIComponent

**DataGridHeaderSeparator** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**DataGridItemRenderer** ■

P: mx.controls.dataGridClasses  
→ UITextField  
IDataRenderer  
IDropInListItemRenderer  
ILayoutManagerClient  
IListItemRenderer  
IStyleClient  
data: Object  
listData: BaseListData  
styleDeclaration: CSSStyleDeclaration  
clearStyle() : void  
DataGridItemRenderer()  
getClassStyleDeclarations() : Array  
getStyle() : \*  
initProtoChain() : void  
notifyStyleChangeInChildren() : void  
regenerateStyleCache() : void  
registerEffects() : void  
setStyle() : void  
toolTipShowHandler() : void  
validateDisplayList() : void  
validateProperties() : void  
validateSize() : void

**DataGridItemRendererAutomationImpl** ■

P: mx.automation.delegates.  
controls  
→ UITextFieldAutomationImpl  
DataGridItemRendererAutomationImpl()  
init() : void

**DataGridListData** ■

P: mx.controls.dataGridClasses  
→ BaseListData  
dataField: String  
DataGridListData()

**DataGridLockedRowContentHolder** ■

P: mx.controls.dataGridClasses  
→ ListBaseContentHolder  
measuredHeight: Number  
DataGridLockedRowContentHolder()

**DataGridSortArrow** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**DataTip** ■

P: mx.charts.chartClasses  
→ UIComponent  
IDataRenderer  
data: Object  
[maxTipWidth: Number](#)  
createChildren() : void  
DataTip()  
measure() : void  
updateDisplayList() : void

**DataTransform** ■

P: mx.charts.chartClasses  
→ EventDispatcher  
axes: Object  
elements: Array  
dataChanged() : void  
DataTransform()  
describeData() : Array

getAxis() : IAxis  
invertTransform() : Array  
setAxis() : void  
transformCache() : void

**Date**

P: Top Level  
→ Object  
date: Number  
dateUTC: Number  
day: Number  
dayUTC: Number  
fullYear: Number  
fullYearUTC: Number  
hours: Number  
hoursUTC: Number  
milliseconds: Number  
millisecondsUTC: Number  
minutes: Number  
minutesUTC: Number  
month: Number  
monthUTC: Number  
seconds: Number  
secondsUTC: Number  
time: Number  
timezoneOffset: Number  
Date()  
getDate() : Number  
getDay() : Number  
getFullYear() : Number  
getHours() : Number  
getMilliseconds() : Number  
getMinutes() : Number  
getMonth() : Number  
getSeconds() : Number  
getTime() : Number  
getTimezoneOffset() : Number  
getUTCDate() : Number  
getUTCDay() : Number  
getUTCFullYear() : Number  
getUTCHours() : Number  
getUTCMilliseconds() : Number  
getUTCMinutes() : Number  
getUTCMonth() : Number  
getUTCSeconds() : Number  
parse() : Number  
setDate() : Number  
setFullYear() : Number  
setHours() : Number  
setMilliseconds() : Number  
setMinutes() : Number  
setMonth() : Number  
setSeconds() : Number  
setTime() : Number  
setUTCDate() : Number  
setUTCFullYear() : Number  
setUTCHours() : Number  
setUTCMilliseconds() : Number  
setUTCMinutes() : Number  
setUTCMonth() : Number  
setUTCSeconds() : Number  
toString() : String  
toLocaleDateString() : String  
toLocaleDateString() : String  
toLocaleTimeString() : String  
toString() : String  
toTimeString() : String  
toUTCString() : String  
UTC() : Number  
valueOf() : Number

**DateBase ■**

P: mx.flash  
→ Object  
dayNamesLong: Array  
dayNamesShort: Array  
monthNamesLong: Array  
monthNamesShort: Array  
timeOfDay: Array

**DateChooser ■**

P: mx.controls  
→ UIComponent  
IFocusManagerComponent  
IFontContextComponent  
allowDisjointSelection: Boolean  
allowMultipleSelection: Boolean  
calendarLayoutStyleFilters: Object  
dayNames: Array  
disabledDays: Array  
disabledRanges: Array  
displayedMonth: int  
displayedYear: int  
firstDayOfWeek: Object  
maxYear: int  
minYear: int  
monthNames: Array  
monthSymbol: String  
nextMonthStyleFilters: Object  
nextYearStyleFilters: Object  
prevMonthStyleFilters: Object  
prevYearStyleFilters: Object  
selectableRange: Object  
selectedDate: Date  
selectedRanges: Array  
showToday: Boolean  
yearNavigationEnabled: Boolean  
yearSymbol: String  
DateChooser()

**DateChooserAutomationImpl ■**

P: mx.automation.delegates.  
controls  
→ UIComponentAutomationImpl  
DateChooserAutomationImpl() : void  
init() : void  
replayAutomatableEvent() : Boolean

**DateChooserEvent ■**

P: mx.events  
→ Event  
detail: String  
SCROLL: String  
triggerEvent: Event  
DateChooserEvent()

**DateChooserEventDetail ■**

P: mx.events  
→ Object  
NEXT\_MONTH: String  
NEXT\_YEAR: String  
PREVIOUS\_MONTH: String  
PREVIOUS\_YEAR: String

**DateChooserIndicator ■**

P: mx.skins.halo  
→ ProgrammaticSkin

**DateChooserMonthArrowSkin ■**

P: mx.skins.halo  
→ Border

**DateChooserYearArrowSkin ■**

P: mx.skins.halo  
→ Border

**DateField ■**

P: mx.controls  
→ ComboBase  
IDataRenderer  
IDropInListItemRenderer  
IFocusManagerComponent  
IListItemRenderer  
data: Object  
dayNames: Array  
disabledDays: Array  
disabledRanges: Array  
displayedMonth: int  
displayedYear: int  
dropdown: DateChooser  
dropdownFactory: IFactory  
firstDayOfWeek: Object  
formatString: String  
labelFunction: Function  
listData: BaseListData  
maxYear: int  
minYear: int  
monthNames: Array  
monthSymbol: String  
parseFunction: Function  
selectableRange: Object  
selectedDate: Date  
showToday: Boolean  
yearNavigationEnabled: Boolean  
yearSymbol: String  
close() : void  
DateField()  
dateToString() : String  
open() : void  
stringToDate() : Date

**DateFieldAutomationImpl ■**

P: mx.automation.delegates.  
controls  
→ ComboBaseAutomationImpl  
DateFieldAutomationImpl() : void  
init() : void

**DateFormatter ■**

P: mx.flash  
→ Formatter  
formatString: String  
DateFormatter()  
format() : String  
parseDateString() : Date

**DateRangeUtilities ■**

P: mx.charts.chartClasses  
→ Object

**DateTimeAxis ■**

P: mx.charts  
→ NumericAxis  
alignLabelsToUnits: Boolean  
dataInterval: Number  
dataUnits: String  
disabledDays: Array  
disabledRanges: Array  
displayLocalTime: Boolean  
interval: Number  
labelUnits: String  
maximum: Date  
minimum: Date  
minorTickInterval: Number  
minorTickUnits: String  
parseFunction: Function  
requiredDescribedFields: uint  
unitSize: Number  
buildMinorTickCache() : Array  
DateTimeAxis()

filterCache() : void  
formatDays() : String  
formatMilliseconds() : String  
formatMinutes() : String  
formatMonths() : String  
formatSeconds() : String  
formatYears() : String  
reduceLabels() : AxisLabelSet  
transformCache() : void

**DateValidator ■**

P: mx.validators  
→ Validator  
allowedFormatChars: String  
dayListener: IValidatorListener  
dayProperty: String  
daySource: Object  
formatError: String  
inputFormat: String  
invalidCharError: String  
monthListener: IValidatorListener  
monthProperty: String  
monthSource: Object  
validateAsString: Object  
wrongDayError: String  
wrongLengthError: String  
wrongMonthError: String  
wrongYearError: String  
yearListener: IValidatorListener  
yearProperty: String  
yearSource: Object  
DateValidator()  
doValidation() : Array  
validateDate() : Array

**DefaultDataDescriptor ■**

P: mx.controls.treeClasses  
→ Object  
ITreeDataDescriptor2  
IMenuDataDescriptor  
addChildAt() : Boolean  
DefaultDataDescriptor()  
getChildren() : ICollectionView  
getData() : Object  
getGroupName() : String  
getHierarchicalCollectionAdaptor() : ICollectionView  
getNodeDepth() : int  
getParent() : Object  
getType() : String  
hasChildren() : Boolean  
isBranch() : Boolean  
isEnabled() : Boolean  
isToggled() : Boolean  
removeChildAt() : Boolean  
setEnabled() : void  
setToggled() : void

**DefaultListEffect ■**

P: mx.effects  
→ Parallel  
color: Number  
fadeOutDuration: Number  
fadeOutDuration: Number  
growDuration: Number  
removedElementOffset: Number  
shrinkDuration: Number

**DefaultTileListEffect ■**

P: mx.effects  
→ Parallel  
color: Number  
fadeOutDuration: Number  
fadeOutDuration: Number  
moveDuration: Number

**DeferredInstanceFromClass ■**

P: mx.core  
→ Object  
IDeferredInstance  
DeferredInstanceFromClass()  
getInstance() : Object

**DeferredInstanceFromFunction ■**

P: mx.core  
→ Object  
IDeferredInstance  
DeferredInstanceFromFunction()  
getInstance() : Object

**DefinitionError**

P: Top Level  
→ Error  
DefinitionError()

**DeleteObjectSample**

P: flash.sampler  
→ Sample  
id: Number  
size: Number

**DescribeTypeCache ■**

P: mx.utils  
→ Object  
describeType() : DescribeTypeCacheRecord  
registerCacheHandler() : void

**DescribeTypeCacheRecord ■**

P: mx.utils  
→ Proxy

**DiamondItemRenderer ■**

P: mx.charts.renderers  
→ ProgrammaticSkin  
IDataRenderer  
data: Object  
DiamondItemRenderer()

**Dictionary**

P: flash.utils  
→ Object

**DisplacementMapFilter**

P: flash.filters  
→ BitmapFilter  
alpha: Number  
color: uint  
componentX: uint  
componentY: uint  
mapBitmap: BitmapData  
mapPoint: Point  
mode: String  
scaleX: Number  
scaleY: Number  
clone() : BitmapFilter  
DisplacementMapFilter()

**DisplacementMapFilterMode**

P: flash.filters  
→ Object  
CLAMP: String  
COLOR: String  
IGNORE: String  
WRAP: String

**DisplayObject**

P: flash.display  
→ EventDispatcher  
IBitmapDrawable  
accessibilityProperties: AccessibilityProperties

alpha: Number  
blendMode: String  
cacheAsBitmap: Boolean  
filters: Array  
height: Number  
loaderInfo: LoaderInfo  
mask: DisplayObject  
mouseX: Number  
mouseY: Number  
name: String  
opaqueBackground: Object  
parent: DisplayObjectContainer  
root: DisplayObject  
rotation: Number  
scale9Grid: Rectangle  
scaleX: Number  
scaleY: Number  
scrollRect: Rectangle  
stage: Stage  
transform: Transform  
visible: Boolean  
width: Number  
x: Number  
y: Number  
getBounds() : Rectangle  
getRect() : Rectangle  
globalToLocal() : Point  
hitTestObject() : Boolean  
hitTestPoint() : Boolean  
localToGlobal() : Point

**DisplayObjectContainer**

P: flash.display  
→ InteractiveObject  
mouseChildren: Boolean  
numChildren: int  
tabChildren: Boolean  
textSnapshot: TextSnapshot  
addChild() : DisplayObject  
addChildAt() : DisplayObject  
areInaccessibleObjectsUnderPoint() : Boolean  
contains() : Boolean  
DisplayObjectContainer()  
getChildAt() : DisplayObject  
getChildByName() : DisplayObject  
getChildIndex() : int  
getObjectsUnderPoint() : Array  
removeChild() : DisplayObject  
removeChildAt() : DisplayObject  
setChildIndex() : void  
swapChildren() : void  
swapChildrenAt() : void

**Dissolve ■**

P: mx.effects  
→ TweenEffect  
alphaFrom: Number  
alphaTo: Number  
color: uint  
targetArea: RoundedRectangle  
Dissolve()

**DissolveInstance ■**

P: mx.effects.effectClasses  
→ TweenEffectInstance  
alphaFrom: Number  
alphaTo: Number  
color: uint  
targetArea: RoundedRectangle  
DissolveInstance()

**DividedBox ■**

P: mx.containers  
→ Box  
dividerClass: Class  
liveDragging: Boolean  
numDividers: int  
resizeToContent: Boolean  
DividedBox()  
getDividerAt() : BoxDivider  
moveDivider() : void

**DividedBoxAutomationImpl ■**

P: mx.automation.delegates.  
containers  
→ ContainerAutomationImpl  
DividedBoxAutomationImpl()  
init() : void  
replayAutomatableEvent() : Boolean

**DividerEvent ■**

P: mx.events  
→ Event  
delta: Number  
DIVIDER\_DRAG: String  
DIVIDER\_PRESS: String  
DIVIDER\_RELEASE: String  
dividerIndex: int  
DividerEvent()

**DockIcon ●**

P: flash.desktop  
→ Interactivelcon  
bitmaps: Array  
height: int  
menu: NativeMenu  
width: int  
bounce()

**DownloadProgressBar ■**

P: mx.preloaders  
→ Sprite  
IPreloaderDisplay  
backgroundAlpha: Number  
backgroundColor: uint  
backgroundImage: Object  
backgroundSize: String  
barFrameRect: RoundedRectangle  
barRect: RoundedRectangle  
borderRect: RoundedRectangle  
DOWNLOAD\_PERCENTAGE: uint  
downloadingLabel: String  
initializingLabel: String  
label: String  
labelFormat: TextFormat  
labelRect: Rectangle  
MINIMUM\_DISPLAY\_TIME: uint  
percentFormat: TextFormat  
percentRect: Rectangle  
preloader: Sprite  
showLabel: Boolean  
showPercentage: Boolean  
stageHeight: Number  
stageWidth: Number  
visible: Boolean  
center() : void  
completeHandler() : void  
createChildren() : void  
DownloadProgressBar()  
getPercentLoaded() : Number  
initialize() : void  
initProgressHandler() : void  
progressHandler() : void  
rsICompleteHandler() : void  
rsIErrorHandler() : void  
rsIProgressHandler() : void

setProgress() : void  
showDisplayForDownloading() : Boolean  
showDisplayForInit() : Boolean

**DragEvent ■**

P: mx.events  
→ MouseEvent  
action: String  
DRAG\_COMPLETE: String  
DRAG\_DROP: String  
DRAG\_ENTER: String  
DRAG\_EXIT: String  
DRAG\_OVER: String  
DRAG\_START: String  
draggedItem: Object  
dragInitiator: UIComponent  
dragSource: DragSource  
DragEvent()

**DragManager ■**

P: mx.managers  
→ Object  
COPY: String  
isDragging: Boolean  
LINK: String  
MOVE: String  
NONE: String  
acceptDragDrop() : void  
doDrag() : void  
getFeedback() : String  
showFeedback() : void

**DragManagerAutomationImpl ■**

P: mx.automation.delegates  
→ UIComponentAutomationImpl  
DragManagerAutomationImpl()  
init() : void

**DragSource ■**

P: mx.core  
→ Object  
formats: Array  
addData() : void  
addHandler() : void  
dataForFormat() : Object  
DragSource()  
hasFormat() : Boolean

**DRMAuthenticateEvent ●**

P: flash.events  
→ Event  
AUTHENTICATION\_TYPE\_DRM: String  
AUTHENTICATION\_TYPE\_PROXY: String  
authenticationType: String  
DRM\_AUTHENTICATE: String  
header: String  
originator: Object  
passwordPrompt: String  
urlPrompt: String  
usernamePrompt: String  
clone() : Event  
DRMAuthenticateEvent()  
toString() : String

**DRMErrorEvent ●**

P: flash.events  
→ ErrorEvent  
DRM\_ERROR: String  
subErrorCode: int  
clone() : Event  
DRMErrorEvent()  
toString() : String

**DRMStatusEvent** ●

P: flash.events  
→ Event  
detail: String  
DRM\_STATUS: String  
isAnonymous: Boolean  
isAvailableOffline: Boolean  
offlineLeasePeriod: uint  
voucherEndDate: Date  
clone() : Event  
DRMStatusEvent()  
toString() : String

**DropdownEvent** ■

P: mx.events  
→ Event  
CLOSE: String  
OPEN: String  
triggerEvent: Event  
DropdownEvent()

**DropShadowFilter**

P: flash.filters  
→ BitmapFilter  
alpha: Number  
angle: Number  
blurX: Number  
blurY: Number  
color: uint  
distance: Number  
hideObject: Boolean  
inner: Boolean  
knockout: Boolean  
quality: int  
strength: Number  
clone() : BitmapFilter  
DropShadowFilter()

**DualStyleObject** ■

P: mx.charts.chartClasses  
→ UIComponent  
internalStyleName: Object  
DualStyleObject()

**DynamicEvent** ■

P: mx.events  
→ Event

**EdgeMetrics** ■

P: mx.core  
→ Object  
bottom: Number  
EMPTY: EdgeMetrics  
left: Number  
right: Number  
top: Number  
clone() : EdgeMetrics  
EdgeMetrics()

**Effect** ■

P: mx.effects  
→ EventDispatcher  
IEffect  
className: String  
customFilter: EffectTargetFilter  
duration: Number  
effectTargetHost: IEffectTargetHost  
endValuesCaptured: Boolean  
filter: String  
hideFocusRing: Boolean  
instanceClass: Class  
isPlaying: Boolean  
perElementOffset: Number

relevantProperties: Array  
relevantStyles: Array  
repeatCount: int  
repeatDelay: int  
startDelay: int  
suspendBackgroundProcessing: Boolean  
target: Object  
targets: Array  
triggerEvent: Event  
applyValueToTarget() : void  
captureEndValues() : void  
captureMoreStartValues() : void  
captureStartValues() : void  
createInstance() : IEffectInstance  
createInstances() : Array  
deleteInstance() : void  
Effect()  
effectEndHandler() : void  
effectStartHandler() : void  
end() : void  
filterInstance() : Boolean  
getAffectedProperties() : Array  
getValueFromTarget() : \*  
initInstance() : void  
pause() : void  
play() : Array  
resume() : void  
reverse() : void  
stop() : void

**EffectEvent** ■

P: mx.events  
→ Event  
EFFECT\_END: String  
EFFECT\_START: String  
effectInstance: IEffectInstance  
EffectEvent() : void

**EffectInstance** ■

P: mx.effects  
→ EventDispatcher  
IEffectInstance  
className: String  
duration: Number  
effect: IEffect  
effectTargetHost: IEffectTargetHost  
hideFocusRing: Boolean  
playheadTime: Number  
propertyChanges: PropertyChanges  
repeatCount: int  
repeatDelay: int  
startDelay: int  
suspendBackgroundProcessing: Boolean  
target: Object  
triggerEvent: Event  
EffectInstance() : void  
end() : void  
finishEffect() : void  
finishRepeat() : void  
initEffect() : void  
pause() : void  
play() : void  
resume() : void  
reverse() : void  
startEffect() : void  
stop() : void

**EffectManager** ■

P: mx.effects  
→ EventDispatcher  
endEffectsForTarget() : void  
resumeEventHandling() : void  
suspendEventHandling() : void

**EffectTargetFilter** ■

P: mx.effects  
→ Object  
filterFunction: Function  
filterProperties: Array  
filterStyles: Array  
requiredSemantics: Object  
defaultFilterFunction() : Boolean  
EffectTargetFilter()  
filterInstance() : Boolean

**Elastic** ■

P: mx.effects.easing  
→ Object  
easeIn() : Number  
easeInOut() : Number  
easeOut() : Number

**EmailValidator** ■

P: mx.validators  
→ Validator  
invalidCharError: String  
invalidDomainError: String  
invalidIPDomainError: String  
invalidPeriodsInDomainError: String  
missingAtSignError: String  
missingPeriodInDomainError: String  
missingUsernameError: String  
tooManyAtSignsError: String  
doValidation() : Array  
EmailValidator()  
validateEmail() : Array

**EncryptedLocalStore** ●

P: flash.data  
→ Object  
getItem() : ByteArray  
removeItem() : void  
reset() : void  
setItem() : void

**Endian**

P: flash.utils  
→ Object  
BIG\_ENDIAN: String  
LITTLE\_ENDIAN: String

**EOFError**

P: flash.errors  
→ Error  
EOFError()

**Error**

P: Top Level  
→ Object  
length: int = 1  
message: String  
name: String  
Error()  
getStackTrace() : String  
toString() : String

**ErrorEvent**

P: flash.events  
→ TextEvent  
ERROR: String  
errorID: int  
clone() : Event  
ErrorEvent()  
toString() : String

**ErrorMessage** ■

P: mx.messaging.messages  
→ AsyncMessage

extendedData: Object  
faultCode: String  
faultDetail: String  
faultString: String  
MESSAGE\_DELIVERY\_IN\_DOUBT: String  
RETRYABLE\_HINT\_HEADER: String  
rootCause: Object  
ErrorMessage()

**EvalError**

P: Top Level  
→ Error  
EvalError()

**Event**

P: flash.events  
→ Object  
ACTIVATE: String  
ADDED: String  
ADDED\_TO\_STAGE: String  
bubbles: Boolean  
CANCEL: String  
cancelable: Boolean  
CHANGE: String  
CLOSE: String  
CLOSING: String  
COMPLETE: String  
CONNECT: String  
currentTarget: Object  
DEACTIVATE: String  
DISPLAYING: String  
ENTER\_FRAME: String  
eventPhase: uint  
EXITING: String  
FULLSCREEN: String  
HTML\_BOUNDS\_CHANGE: String  
HTML\_DOM\_INITIALIZE: String  
HTML\_RENDERER: String  
ID3: String  
INIT: String  
LOCATION\_CHANGE: String  
MOUSE\_LEAVE: String  
NETWORK\_CHANGE: String  
OPEN: String  
REMOVED: String  
REMOVED\_FROM\_STAGE: String  
RENDER: String  
RESIZE: String  
SCROLL: String  
SELECT: String  
SOUND\_COMPLETE: String  
TAB\_CHILDREN\_CHANGE: String  
TAB\_ENABLED\_CHANGE: String  
TAB\_INDEX\_CHANGE: String  
target: Object  
type: String  
UNLOAD: String  
USER\_IDLE: String  
USER\_PRESENT: String  
clone() : Event  
Event()  
formatToString() : String  
isDefaultPrevented() : Boolean  
preventDefault() : void  
stopImmediatePropagation() : void  
stopPropagation() : void  
toString() : String

**EventDispatcher**

P: flash.events  
→ Object  
IEventDispatcher  
addEventListener() : void

dispatchEvent() : Boolean  
EventDispatcher() : void  
hasEventListener() : Boolean  
removeEventListener() : void  
willTrigger() : Boolean

**EventPhase**

P: flash.events  
→ Object  
AT\_TARGET: uint  
BUBBLING\_PHASE: uint  
CAPTURING\_PHASE: uint

**EventPriority** ■

P: mx.core  
→ Object  
BINDING: int  
CURSOR\_MANAGEMENT: int  
DEFAULT: int  
DEFAULT\_HANDLER: int  
EFFECT: int

**Exponential** ■

P: mx.effects.easing  
→ Object  
easeIn() : Number  
easeInOut() : Number  
easeOut() : Number

**ExternalInterface**

P: flash.external  
→ Object  
available: Boolean  
marshallExceptions: Boolean  
objectID: String  
addCallback() : void  
call() : \*

**Fade** ■

P: mx.effects  
→ TweenEffect  
alphaFrom: Number  
alphaTo: Number  
Fade()

**FadeInstance** ■

P: mx.effects.effectClasses  
→ TweenEffectInstance  
alphaFrom: Number  
alphaTo: Number  
FadeInstance()

**Fault** ■

P: mx.rpc  
→ Error  
\_faultCode: String  
\_faultDetail: String  
\_faultString: String  
faultCode: String  
faultDetail: String  
faultString: String  
rootCause: Object  
Fault()  
toString() : String

**FaultEvent** ■

P: mx.rpc.events  
→ AbstractEvent  
FAULT: String  
fault: Fault  
headers: Object  
createEvent() : FaultEvent  
createEventFromMessageFault() : FaultEvent  
FaultEvent() : void

toString() : String

**File** ●

P: flash.filesystem  
→ FileReference  
applicationDirectory: File  
applicationStorageDirectory: File  
desktopDirectory: File  
documentsDirectory: File  
exists: Boolean  
icon: Icon  
isDirectory: Boolean  
isHidden: Boolean  
isPackage: Boolean  
isSymbolicLink: Boolean  
lineEnding: String  
nativePath: String  
parent: File  
separator: String  
systemCharset: String  
url: String  
userDirectory: File  
browseForDirectory() : void  
browseForOpen() : void  
browseForOpenMultiple() : void  
browseForSave() : void  
cancel() : void  
canonicalize() : void  
clone() : File  
copyTo() : void  
copyToAsync() : void  
createDirectory() : void  
createTempDirectory() : File  
createTempFile() : File  
deleteDirectory() : void  
deleteDirectoryAsync() : void  
deleteFile() : void  
deleteFileAsync() : void  
File()  
getDirectoryListing() : Array  
getDirectoryListingAsync() : void  
getRelativePath() : String  
getRootDirectories() : Array  
moveTo() : void  
moveToAsync() : void  
moveToTrash() : void  
moveToTrashAsync() : void  
resolvePath() : File

**FileEvent** ■●

P: mx.events  
→ Event  
DIRECTORY\_CHANGE: String  
DIRECTORY\_CHANGING: String  
DIRECTORY\_CLOSING: String  
DIRECTORY\_OPENING: String  
file: File  
FILE\_CHOOSE: String  
FileEvent()

**FileFilter**

P: flash.net  
→ Object  
description: String  
extension: String  
macType: String  
FileFilter() : void

**FileListEvent** ●

P: flash.events  
→ Event  
DIRECTORY\_LISTING: String  
files: Array  
SELECT\_MULTIPLE: String  
FileListEvent()

**FileMode** ●

P: flash.filesystem  
→ Object  
APPEND: String  
READ: String  
UPDATE: String  
WRITE: String

**FileReference**

P: flash.net  
→ EventDispatcher  
creationDate: Date  
creator: String  
extension: String  
modificationDate: Date  
name: String  
size: Number  
type: String  
browse() : Boolean  
cancel() : void  
download() : void  
FileReference() : void  
upload() : void  
uploadUnencoded() : void

**FileReferenceList**

P: flash.net  
→ EventDispatcher  
fileList: Array  
browse() : Boolean  
FileReferenceList()

**FileStream** ●

P: flash.filesystem  
→ EventDispatcher  
IDataInput  
IDataOutput  
bytesAvailable: uint  
endian: String  
objectEncoding: uint  
position: Number  
readAhead: Number  
close() : void  
FileStream()  
open() : void  
openAsync() : void  
readBoolean() : Boolean  
readByte() : int  
readBytes() : void  
readDouble() : Number  
readFloat() : Number  
readInt() : int  
readMultiByte() : String  
readObject() : \*  
readShort() : int  
readUnsignedByte() : uint  
readUnsignedInt() : uint  
readUnsignedShort() : uint  
readUTF() : String  
readUTFBytes() : String  
truncate() : void  
writeBoolean() : void  
writeByte() : void  
writeBytes() : void  
writeDouble() : void  
writeFloat() : void  
writeInt() : void  
writeMultiByte() : void  
writeObject() : void  
writeShort() : void  
writeUnsignedInt() : void  
writeUTF() : void  
writeUTFBytes() : void

**FileSystemComboBox** ■●

P: mx.controls  
→ ComboBox  
COMPUTER: File  
directory: File  
indent: int  
showIcons: Boolean  
FileSystemComboBox()

**FileSystemDataGrid** ■●

P: mx.controls  
→ DataGrid  
backHistory: Array  
canNavigateBack: Boolean  
canNavigateDown: Boolean  
canNavigateForward: Boolean  
canNavigateUp: Boolean  
COMPUTER: File  
creationDateColumn: DataGridColumn  
dateFormatString: String  
directory: File  
enumerationMode: String  
extensions: Array  
filterFunction: Function  
forwardHistory: Array  
modificationDateColumn: DataGridColumn  
nameColumn: DataGridColumn  
nameCompareFunction: Function  
selectedPath: String  
selectedPaths: Array  
showExtensions: Boolean  
showHidden: Boolean  
showIcons: Boolean  
sizeColumn: DataGridColumn  
sizeDisplayMode: String  
typeColumn: DataGridColumn  
clear() : void  
FileSystemDataGrid()  
findIndex() : int  
findItem() : File  
navigateBack() : void  
navigateDown() : void  
navigateForward() : void  
navigateTo() : void  
navigateUp() : void  
refresh() : void

**FileSystemEnumerationMode** ■●

P: mx.controls  
→ Object  
DIRECTORIES\_FIRST: String  
DIRECTORIES\_ONLY: String  
FILES\_AND\_DIRECTORIES: String  
FILES\_FIRST: String  
FILES\_ONLY: String

**FileSystemHistoryButton** ■●

P: mx.controls  
→ PopUpButton  
dataProvider: Object  
FileSystemHistoryButton()

**FileSystemList** ■●

P: mx.controls  
→ List  
backHistory: Array  
canNavigateBack: Boolean  
canNavigateDown: Boolean  
canNavigateForward: Boolean  
canNavigateUp: Boolean  
COMPUTER: File  
directory: File  
enumerationMode: String



extensions: Array  
 filterFunction: Function  
 forwardHistory: Array  
 nameCompareFunction: Function  
 selectedPath: String  
 selectedPaths: Array  
 showExtensions: Boolean  
 showHidden: Boolean  
 showIcons: Boolean  
 clear(): void  
 FileSystemList()  
 findIndex(): int  
 findItem(): File  
 navigateBack(): void  
 navigateDown(): void  
 navigateForward(): void  
 navigateTo(): void  
 navigateUp(): void  
 refresh(): void

#### FileSystemSizeDisplayMode ■●

P: mx.controls  
 → Object  
 BYTES: String  
 KILOBYTES: String

#### FileSystemTree ■●

P: mx.controls  
 → Tree  
 COMPUTER: File  
 directory: File  
 enumerationMode: String  
 extensions: Array  
 filterFunction: Function  
 nameCompareFunction: Function  
 openPaths: Array  
 selectedPath: String  
 selectedPaths: Array  
 showExtensions: Boolean  
 showHidden: Boolean  
 showIcons: Boolean  
 clear(): void  
 closeSubdirectory(): void  
 FileSystemTree()  
 findIndex(): int  
 findItem(): File  
 openSubdirectory(): void  
 refresh(): void

#### FlexBitmap ■

P: mx.core  
 → Bitmap  
 FlexBitmap()  
 toString(): String

#### FlexClient ■

→ EventDispatcher  
 P: mx.messaging  
 id: String  
 getInstance(): FlexClient

#### FlexContentHolder ■

P: mx.flash  
 → ContainerMovieClip  
 content: IUIComponent  
 FlexContentHolder()  
 initialize(): void  
 notifySizeChanged(): void  
 setActualSize(): void  
 setFlexContent(): void

#### FlexContentHolderAutomation-Impl ■

P: mx.automation.delegates.  
 flashflexkit  
 → UIMovieClipAutomationImpl  
 FlexContentHolderAutomationImpl()  
 init():void

#### FlexEvent ■

P: mx.events  
 → Event  
 ADD: String  
 APPLICATION\_COMPLETE: String  
 BUTTON\_DOWN: String  
 CREATION\_COMPLETE: String  
 CURSOR\_UPDATE: String  
 DATA\_CHANGE: String  
 ENTER: String  
 ENTER\_STATE: String  
 EXIT\_STATE: String  
 HIDE: String  
 IDLE: String  
 INIT\_COMPLETE: String  
 INIT\_PROGRESS: String  
 INITIALIZE: String  
 INVALID: String  
 LOADING: String  
 PREINITIALIZE: String  
 REMOVE: String  
 REPEAT\_END: String  
 REPEAT\_START: String  
 SHOW: String  
 TRANSFORM\_CHANGE: String  
 UPDATE\_COMPLETE: String  
 URL\_CHANGED: String  
 VALID: String  
 VALUE\_COMMIT: String  
 FlexEvent()

#### FlexHTMLLoader ■●

P: mx.core  
 → HTMLLoader  
 IFocusManagerComplexComponent  
 focusEnabled: Boolean  
 mouseFocusEnabled: Boolean  
 assignFocus(): void  
 drawFocus(): void  
 FlexHTMLLoader()  
 setFocus(): void  
 toString(): String

#### FlexMouseEvent ■

P: mx.events  
 → MouseEvent  
 MOUSE\_DOWN\_OUTSIDE: String  
 MOUSE\_WHEEL\_OUTSIDE: String  
 FlexMouseEvent()

#### FlexMovieClip ■

P: mx.core  
 → MovieClip  
 FlexMovieClip()  
 toString(): String

#### FlexNativeMenu ■●

P: mx.controls  
 → EventDispatcher  
 ILayoutManagerClient  
 IFlexContextMenu  
 dataDescriptor: IMenuDataDescriptor  
 dataProvider: Object  
 hasRoot: Boolean  
 initialized: Boolean  
 keyEquivalentField: String

keyEquivalentFunction: Function  
 keyEquivalentModifiersFunction: Function  
 labelField: String  
 labelFunction: Function  
 mnemonicIndexField: String  
 mnemonicIndexFunction: Function  
 nativeMenu: NativeMenu  
 nestLevel: int  
 processedDescriptors: Boolean  
 showRoot: Boolean  
 updateCompletePendingFlag: Boolean  
 commitProperties(): void  
 display(): void  
 FlexNativeMenu()  
 invalidateProperties(): void  
 itemToKeyEquivalent(): String  
 itemToKeyEquivalentModifiers(): Array  
 itemToLabel(): String  
 itemToMnemonicIndex(): int  
 parseLabelToMnemonicIndex(): int  
 parseLabelToString(): String  
 setContextMenu(): void  
 unsetContextMenu(): void  
 validateDisplayList(): void  
 validateNow(): void  
 validateProperties(): void  
 validateSize(): void

#### FlexNativeMenuEvent ■●

P: mx.events  
 → Event  
 index: int  
 item: Object  
 ITEM\_CLICK: String  
 label: String  
 MENU\_SHOW: String  
 nativeMenu: NativeMenu  
 nativeMenuItem: NativeMenuItem  
 FlexNativeMenuEvent()

#### FlexNativeWindowBoundsEvent ■●

P: mx.events  
 → NativeWindowBoundsEvent  
 WINDOW\_MOVE: String  
 WINDOW\_RESIZE: String  
 FlexNativeWindowBoundsEvent()

#### FlexPrintJob ■

P: mx.printing  
 → Object  
 pageHeight: Number  
 pageWidth: Number  
 printAsBitmap: Boolean  
 addObject(): void  
 FlexPrintJob()  
 send(): void  
 start(): Boolean

#### FlexPrintJobScaleType ■

P: mx.printing  
 → Object  
 FILL\_PAGE: String  
 MATCH\_HEIGHT: String  
 MATCH\_WIDTH: String  
 NONE: String  
 SHOW\_ALL: String

#### FlexShape ■

P: mx.core  
 → Shape  
 FlexShape()  
 toString(): String

#### FlexSimpleButton ■

P: mx.core  
 → SimpleButton  
 FlexSimpleButton()  
 toString(): String

#### FlexSprite ■

P: mx.core  
 → Sprite  
 FlexSprite()  
 toString(): String

#### FlexTextField ■

P: mx.core  
 → TextField  
 FlexTextField()  
 toString(): String

#### FlexVersion ■

P: mx.core  
 → Object  
 compatibilityErrorFunction: Function  
 compatibilityVersion: uint  
 compatibilityVersionString: String  
 CURRENT\_VERSION: uint  
 VERSION\_2\_0: uint  
 VERSION\_2\_0\_1: uint  
 VERSION\_3\_0: uint  
 VERSION\_ALREADY\_READ: String  
 VERSION\_ALREADY\_SET: String

#### FocusDirection ●

P: flash.display  
 → Object  
 BOTTOM: String  
 NONE: String  
 TOP: String

#### FocusEvent ■

P: flash.events  
 → Event  
 direction: String  
 FOCUS\_IN: String  
 FOCUS\_OUT: String  
 KEY\_FOCUS\_CHANGE: String  
 keyCode: uint  
 MOUSE\_FOCUS\_CHANGE: String  
 relatedObject: InteractiveObject  
 shiftKey: Boolean  
 clone(): Event  
 FocusEvent()  
 toString(): String

#### FocusManager ■

P: mx.managers  
 → Object  
 IFocusManager  
 defaultButton: IButton  
 defaultButtonEnabled: Boolean  
 focusPane: Sprite  
 nextTabIndex: int  
 showFocusIndicator: Boolean  
 activate(): void  
 deactivate(): void  
 findFocusManagerComponent(): IFocusManagerComponent  
 FocusManager()  
 getFocus(): IFocusManagerComponent  
 getNextFocusManagerComponent(): IFocusManagerComponent  
 hideFocus(): void  
 setFocus(): void  
 showFocus(): void  
 toString(): String

#### Font ■

P: flash.text  
 → Object  
 fontName: String  
 fontStyle: String  
 fontType: String  
 enumerateFonts(): Array  
 hasGlyphs(): Boolean  
 registerFont(): void

#### FontAsset ■

P: mx.core  
 → Font  
 IFlexAsset

#### FontStyle ■

P: flash.text  
 → Object  
 BOLD: String  
 BOLD\_ITALIC: String  
 ITALIC: String  
 REGULAR: String

#### FontType ■

P: flash.text  
 → Object  
 DEVICE: String  
 EMBEDDED: String

#### Form ■

P: mx.containers  
 → Container  
 maxLabelWidth: Number  
 Form()  
 measure(): void  
 updateDisplayList(): void

#### Formatter ■

P: mx.flash  
 → Object  
 defaultInvalidFormatError: String  
 defaultInvalidValueError: String  
 error: String  
 resourceManager: IResourceManager  
 format(): String  
 Formatter()  
 resourcesChanged(): void

#### FormAutomationImpl ■

P: mx.automation.delegates.  
 containers  
 → ContainerAutomationImpl  
 FormAutomationImpl()  
 void  
 init(): void

#### FormHeading ■

P: mx.containers  
 → UIComponent  
 label: String  
 FormHeading()

#### FormItem ■

P: mx.containers  
 → Container  
 direction: String  
 itemLabel: Label  
 label: String  
 required: Boolean  
 FormItem()  
 measure(): void  
 updateDisplayList(): void

#### FormItemAutomationImpl ■

P: mx.automation.delegates.  
 containers  
 → ContainerAutomationImpl  
 FormItemAutomationImpl()  
 void  
 init(): void

#### FormItemDirection ■

P: mx.containers  
 → Object  
 HORIZONTAL: String  
 VERTICAL: String

#### FormItemLabel ■

P: mx.controls  
 → Label

#### FrameLabel ■

P: flash.display  
 → Object  
 frame: int  
 name: String

#### FullScreenEvent ■

P: flash.events  
 → ActivityEvent  
 FULL\_SCREEN: String  
 fullScreen: Boolean  
 clone(): Event  
 FullScreenEvent()  
 toString(): String

#### Function ■

P: Top Level  
 → Object  
 apply(): \*  
 call(): \*

#### Glow ■

P: mx.effects  
 → TweenEffect  
 alphaFrom: Number  
 alphaTo: Number  
 blurXFrom: Number  
 blurXTo: Number  
 blurYFrom: Number  
 blurYTo: Number  
 color: uint  
 inner: Boolean  
 knockout: Boolean  
 strength: Number  
 Glow()

#### GlowFilter ■

P: flash.filters  
 → BitmapFilter  
 alpha: Number  
 blurX: Number  
 blurY: Number  
 color: uint  
 inner: Boolean  
 knockout: Boolean  
 quality: int  
 strength: Number  
 clone(): BitmapFilter  
 GlowFilter()

#### GlowInstance ■

P: mx.effects.effectClasses  
 → TweenEffectInstance  
 alphaFrom: Number  
 alphaTo: Number  
 blurXFrom: Number  
 blurXTo: Number  
 blurYFrom: Number

blurYTo: Number  
 color: uint  
 inner: Boolean  
 knockout: Boolean  
 strength: Number  
 GlowInstance()

#### GradientBase ■

P: mx.graphics  
 → EventDispatcher  
 entries: Array  
 GradientBase()

#### GradientBevelFilter ■

P: flash.filters  
 → BitmapFilter  
 alphas: Array  
 angle: Number  
 blurX: Number  
 blurY: Number  
 colors: Array  
 distance: Number  
 knockout: Boolean  
 quality: int  
 ratios: Array  
 strength: Number  
 type: String  
 clone(): BitmapFilter  
 GradientBevelFilter()

#### GradientEntry ■

P: mx.graphics  
 → EventDispatcher  
 alpha: Number  
 color: uint  
 ratio: Number  
 GradientEntry()

#### GradientGlowFilter ■

P: flash.filters  
 → BitmapFilter  
 alphas: Array  
 angle: Number  
 blurX: Number  
 blurY: Number  
 colors: Array  
 distance: Number  
 knockout: Boolean  
 quality: int  
 ratios: Array  
 strength: Number  
 type: String  
 clone(): BitmapFilter  
 GradientGlowFilter()

#### GradientType ■

P: flash.display  
 → Object  
 LINEAR: String = "linear"  
 RADIAL: String = "radial"

#### Graphics ■

P: flash.display  
 → Object  
 beginBitmapFill(): void  
 beginFill(): void  
 beginGradientFill(): void  
 clear(): void  
 curveTo(): void  
 drawCircle(): void  
 drawEllipse(): void  
 drawRect(): void  
 drawRoundRect(): void  
 endFill(): void  
 lineGradientStyle(): void

lineStyle(): void  
 lineTo(): void  
 moveTo(): void

#### GraphicsUtil ■

P: mx.utils  
 → Object  
 drawRoundRectComplex(): void

#### Grid ■

P: mx.containers  
 → Box  
 Grid()  
 measure(): void  
 updateDisplayList(): void

#### GridFitType ■

P: flash.text  
 → Object  
 NONE: String  
 PIXEL: String  
 SUBPIXEL: String

#### GridItem ■

P: mx.containers  
 → HBox  
 colSpan: int  
 rowSpan: int  
 GridItem()

#### GridLines ■

→ ChartElement  
 P: mx.charts

#### GridRow ■

P: mx.containers  
 → HBox  
 GridRow()  
 updateDisplayList(): void

#### Grouping ■

P: mx.collections  
 → Object  
 compareFunction: Function  
 fields: Array  
 groupingObjectFunction: Function  
 label: String  
 Grouping()

#### GroupingCollection ■

P: mx.collections  
 → HierarchicalData  
 IGroupingCollection  
 grouping: Grouping  
 source: Object  
 summaries: Array  
 timer: Timer  
 cancelRefresh(): void  
 getParent(): \*  
 getRoot(): Object  
 GroupingCollection()  
 refresh(): Boolean

#### GroupingField ■

P: mx.collections  
 → Object  
 caseInsensitive: Boolean  
 compareFunction: Function  
 descending: Boolean  
 groupingFunction: Function  
 groupingObjectFunction: Function  
 name: String  
 numeric: Boolean  
 summaries: Array  
 GroupingField()

**HaloBorder** ■

P: mx.skins.halo  
→ RectangularBorder

**HaloColors** ■

P: mx.skins.halo  
→ Object  
[addHaloColors\(\)](#) : void  
[getCacheKey\(\)](#) : String

**HaloDefaults** ■

P: mx.charts.styles  
→ Object  
[createSelector\(\)](#) : CSSStyleDeclaration  
[init\(\)](#) : void

**HaloFocusRect** ■

P: mx.skins.halo  
→ ProgrammaticSkin  
*IStyleClient*

**HBox** ■

P: mx.containers  
→ Box

**HDividedBox** ■

P: mx.containers  
→ DividedBox

**HeaderEvent** ■

P: mx.rpc.events  
→ AbstractEvent  
[HEADER:](#) String  
[header:](#) Object  
[createEvent\(\)](#) : HeaderEvent  
HeaderEvent()  
toString() : String

**HierarchicalCollectionView** ■

P: mx.collections  
→ EventDispatcher  
*IHierarchicalCollectionView*  
[hasRoot:](#) Boolean  
[length:](#) int  
[openNodes:](#) Object  
[showRoot:](#) Boolean  
[source:](#) IHierarchicalData  
[addChild\(\)](#) : Boolean  
[addChildAt\(\)](#) : Boolean  
[closeNode\(\)](#) : void  
[contains\(\)](#) : Boolean  
[createCursor\(\)](#) : IViewCursor  
[getNodeDepth\(\)](#) : int  
[getParentItem\(\)](#) : \*  
*HierarchicalCollectionView()*  
[openNode\(\)](#) : void  
[removeChild\(\)](#) : Boolean  
[removeChildAt\(\)](#) : Boolean  
[xmlNotification\(\)](#) : void

**HierarchicalCollectionView Cursor** ■

P: mx.collections  
→ EventDispatcher  
*IHierarchicalCollectionViewCursor*  
[afterLast:](#) Boolean  
[beforeFirst:](#) Boolean  
[currentDepth:](#) int  
[view:](#) ICollectionView  
*HierarchicalCollectionViewCursor()*

**HierarchicalData** ■

P: mx.collections  
→ EventDispatcher  
*IHierarchicalData*

[childrenField:](#) String  
[source:](#) Object  
[canHaveChildren\(\)](#) : Boolean  
[getChildren\(\)](#) : Object  
[getData\(\)](#) : Object  
[getRoot\(\)](#) : Object  
[hasChildren\(\)](#) : Boolean  
*HierarchicalData()*

**HistoryManager** ■

P: mx.managers  
→ Object  
[initialize\(\)](#) : void  
[register\(\)](#) : void  
[save\(\)](#) : void  
[unregister\(\)](#) : void

**HitData** ■

P: mx.charts  
→ Object  
[chartItem:](#) ChartItem  
[contextColor:](#) uint  
[dataTipFunction:](#) Function  
[displayText:](#) String  
[distance:](#) Number  
[element:](#) IChartElement  
[id:](#) Number  
[item:](#) Object  
[x:](#) Number  
[y:](#) Number  
*HitData()* : void

**HLOCChart** ■

P: mx.charts  
→ CartesianChart

**HLOCItemRenderer** ■

P: mx.charts.renderers  
→ ProgrammaticSkin  
*IDataRenderer*  
[data:](#) Object  
*HLOCItemRenderer()*

**HLOCSeries** ■

P: mx.charts.series  
→ HLOCSeriesBase

**HLOCSeriesBase** ■

P: mx.charts.chartClasses  
→ Series  
*IColumn*  
[closeField:](#) String  
[columnWidthRatio:](#) Number  
[highField:](#) String  
[horizontalAxis:](#) IAxis  
[items:](#) Array  
[itemType:](#) Class  
[legendData:](#) Array  
[lowField:](#) String  
[maxColumnWidth:](#) Number  
[offset:](#) Number  
[openField:](#) String  
[renderData:](#) Object  
[renderDataTypes:](#) Class  
[verticalAxis:](#) IAxis  
[xField:](#) String  
[beginInterpolation\(\)](#) : Object  
[commitProperties\(\)](#) : void  
[describeData\(\)](#) : Array  
[formatDataTip\(\)](#) : String  
[getElementBounds\(\)](#) : void  
[getItemsInRegion\(\)](#) : Array  
[getMissingInterpolationValues\(\)](#) : void  
*HLOCSeriesBase()*

[styleChanged\(\)](#) : void  
[updateData\(\)](#) : void  
[updateDisplayList\(\)](#) : void  
[updateFilter\(\)](#) : void  
[updateMapping\(\)](#) : void  
[updateTransform\(\)](#) : void

**HLOCSeriesBaseAutomation-Impl** ■

P: mx.automation.delegates.charts  
→ SeriesAutomationImpl  
*HLOCSeriesBaseAutomationImpl()*  
[init\(\)](#) : void

**HLOCSeriesItem** ■

P: mx.charts.series.items  
→ ChartItem  
[close:](#) Number  
[closeFilter:](#) Number  
[closeNumber:](#) Number  
[closeValue:](#) Object  
[fill:](#) IFill  
[high:](#) Number  
[highFilter:](#) Number  
[highNumber:](#) Number  
[highValue:](#) Object  
[low:](#) Number  
[lowFilter:](#) Number  
[lowNumber:](#) Number  
[lowValue:](#) Object  
[open:](#) Number  
[openFilter:](#) Number  
[openNumber:](#) Number  
[openValue:](#) Object  
[x:](#) Number  
[xFilter:](#) Number  
[xNumber:](#) Number  
[xValue:](#) Object  
*HLOCSeriesItem()*

**HLOCSeriesRenderData** ■

P: mx.charts.series.renderData  
→ RenderData  
[renderedHalfWidth:](#) Number  
[renderedXOffset:](#) Number  
*HLOCSeriesRenderData()*

**HorizontalList** ■

P: mx.controls  
→ TileBase

**HRule** ■

P: mx.controls  
→ UIComponent

**HScrollbar** ■

P: mx.controls  
→ ScrollBar

**HSlider** ■

P: mx.controls  
→ Slider

**HTML** ■ ●

P: mx.controls  
→ ScrollControlBase  
*IDataRenderer*  
*IDropInListItemRenderer*  
*IListItemRenderer*  
*IFocusManagerComponent*  
[contentHeight:](#) Number  
[contentWidth:](#) Number  
[data:](#) Object  
[historyLength:](#) int

[historyPosition:](#) int  
[htmlHost:](#) HTMLHost  
[htmlLoader:](#) HTMLLoader  
[htmlLoaderFactory:](#) IFactory  
[htmlText:](#) String  
[JavaScriptWindow:](#) Object  
[listData:](#) BaseListData  
[loaded:](#) Boolean  
[location:](#) String  
[paintsDefaultBackground:](#) Boolean  
[pdfCapability:](#) int  
[runtimeApplicationDomain:](#) ApplicationDomain  
[userAgent:](#) String  
[cancelLoad\(\)](#) : void  
[getHistoryAt\(\)](#) : HTMLHistoryItem  
[historyBack\(\)](#) : void  
[historyForward\(\)](#) : void  
[historyGo\(\)](#) : void  
*HTML()*  
[reload\(\)](#) : void

**HTMLHistoryItem** ●

P: flash.html  
→ Object  
[isPost:](#) Boolean  
[originalUrl:](#) String  
[title:](#) String  
[url:](#) String  
*HTMLHistoryItem()* : void

**HTMLHost** ●

P: flash.html  
→ Object  
[htmlLoader:](#) HTMLLoader  
[windowRect:](#) Rectangle  
*createWindow()* : HTMLLoader  
*HTMLHost()*  
[updateLocation\(\)](#) : void  
[updateStatus\(\)](#) : void  
[updateTitle\(\)](#) : void  
[windowBlur\(\)](#) : void  
[windowClose\(\)](#) : void  
[windowFocus\(\)](#) : void

**HTMLLoader** ●

P: flash.html  
→ Sprite  
[authenticate:](#) Boolean  
[cacheResponse:](#) Boolean  
[contentHeight:](#) Number  
[contentWidth:](#) Number  
[hasFocusableContent:](#) Boolean  
[height:](#) Number  
[historyLength:](#) uint  
[historyPosition:](#) uint  
[htmlHost:](#) HTMLHost  
[loaded:](#) Boolean  
[location:](#) String  
[manageCookies:](#) Boolean  
[paintsDefaultBackground:](#) Boolean  
[pdfCapability:](#) int  
[runtimeApplicationDomain:](#) ApplicationDomain  
[scrollH:](#) Number  
[scrollV:](#) Number  
[textEncodingFallback:](#) String  
[textEncodingOverride:](#) String  
[useCache:](#) Boolean  
[userAgent:](#) String  
[width:](#) Number  
[window:](#) Object  
[cancelLoad\(\)](#) : void  
[createRootWindow\(\)](#) : HTMLLoader

[getHistoryAt\(\)](#) : HTMLHistoryItem  
[historyBack\(\)](#) : void  
[historyForward\(\)](#) : void  
[historyGo\(\)](#) : void  
*HTMLLoader()*  
[load\(\)](#) : void  
[loadString\(\)](#) : void  
[reload\(\)](#) : void

**HTMLPDFCapability** ●

P: flash.html  
→ Object  
[ERROR\\_CANNOT\\_LOAD\\_READER:](#) int  
[ERROR\\_INSTALLED\\_READER\\_NOT\\_FOUND:](#) int  
[ERROR\\_INSTALLED\\_READER\\_TOO\\_OLD:](#) int  
[ERROR\\_PREFERRED\\_READER\\_TOO\\_OLD:](#) int  
[STATUS\\_OK:](#) int

**HTMLUncaughtScriptException Event** ●

P: flash.events  
→ Event  
[exceptionValue:](#) \*  
[stackTrace:](#) Array  
*UNCAUGHT\_SCRIPT\_EXCEPTION:*  
[clone\(\)](#) : Event  
*HTMLUncaughtScriptExceptionEvent()* : void

**HTMLWindowCreateOptions** ●

P: flash.html  
→ Event  
[fullscreen:](#) Boolean  
[height:](#) Number  
[locationBarVisible:](#) Boolean  
[menuBarVisible:](#) Boolean  
[resizable:](#) Boolean  
[scrollBarsVisible:](#) Boolean  
[statusBarVisible:](#) Boolean  
[toolBarVisible:](#) Boolean  
[width:](#) Number  
[x:](#) Number  
[y:](#) Number

**HTTPChannel** ■

P: mx.messaging.channels  
→ PollingChannel  
[polling:](#) Boolean  
[pollingEnabled:](#) Boolean  
[pollingInterval:](#) Number  
[protocol:](#) String  
*HTTPChannel()*  
[internalPingComplete\(\)](#) : void

**HTTPRequestMessage** ■

P: mx.messaging.messages  
→ AcknowledgeMessage  
[CONTENT\\_TYPE\\_FORM:](#) String  
[CONTENT\\_TYPE\\_SOAP\\_XML:](#) String  
[CONTENT\\_TYPE\\_XML:](#) String  
[contentType:](#) String  
[DELETE\\_METHOD:](#) String  
[GET\\_METHOD:](#) String  
[HEAD\\_METHOD:](#) String  
[httpHeaders:](#) Object  
[method:](#) String  
[OPTIONS\\_METHOD:](#) String  
[POST\\_METHOD:](#) String  
[PUT\\_METHOD:](#) String

[recordHeaders:](#) Boolean  
[TRACE\\_METHOD:](#) String  
[url:](#) String  
*HTTPRequestMessage()*

**HTTPService** ■

P: mx.rpc.http  
→ AbstractInvoker  
[channelSet:](#) ChannelSet  
[CONTENT\\_TYPE\\_FORM:](#) String  
[CONTENT\\_TYPE\\_XML:](#) String  
[contentType:](#) String  
[DEFAULT\\_DESTINATION\\_HTTP:](#) String  
[DEFAULT\\_DESTINATION\\_HTTPS:](#) String  
[destination:](#) String  
[ERROR\\_DECODING:](#) String  
[ERROR\\_ENCODING:](#) String  
[ERROR\\_URL\\_REQUIRED:](#) String  
[headers:](#) Object  
[method:](#) String  
[request:](#) Object  
[requestTimeout:](#) int  
[RESULT\\_FORMAT\\_ARRAY:](#) String  
[RESULT\\_FORMAT\\_E4X:](#) String  
[RESULT\\_FORMAT\\_FLASHVARS:](#) String  
[RESULT\\_FORMAT\\_OBJECT:](#) String  
[RESULT\\_FORMAT\\_TEXT:](#) String  
[RESULT\\_FORMAT\\_XML:](#) String  
[resultFormat:](#) String  
[rootURL:](#) String  
[url:](#) String  
[useProxy:](#) Boolean  
[xmlDecode:](#) Function  
[xmlEncode:](#) Function  
[disconnect\(\)](#) : void  
*HTTPService()*  
[logout\(\)](#) : void  
[send\(\)](#) : AsyncToken  
[setCredentials\(\)](#) : void  
[setRemoteCredentials\(\)](#) : void

**HTTPService** ■

P: mx.rpc.http.mxxml  
→ HTTPService  
*IMXMLSupport*  
*IMXMLObject*  
[concurrency:](#) String  
[showBusyCursor:](#) Boolean  
[cancel\(\)](#) : AsyncToken  
*HTTPService()*  
[initialized\(\)](#) : void  
[send\(\)](#) : AsyncToken

**HTTPStatusEvent**

P: flash.events  
→ Event  
[HTTP\\_RESPONSE\\_STATUS:](#) String  
[HTTP\\_STATUS:](#) String  
[responseHeaders:](#) Array  
[responseURL:](#) String  
[status:](#) int  
[clone\(\)](#) : Event  
*HTTPStatusEvent()*  
[toString\(\)](#) : String

**IAbstractEffect** ■

P: mx.effects

**IAdvancedDataGridRenderer Provider** ■

P: mx.controls.advanced  
*describeRendererForItem()* : void

**IAutomationClass** ■

P: mx.automation  
[name:](#) String  
[propertyNameMap:](#) Object  
[superClassName:](#) String  
[getDescriptorForEvent\(\)](#) : IAutomationEventDescriptor  
[getDescriptorForEventByName\(\)](#) : IAutomationEventDescriptor  
[getDescriptorForMethodByName\(\)](#) : IAutomationMethodDescriptor  
[getPropertyDescriptors\(\)](#) : Array

**IAutomationEnvironment** ■

P: mx.automation  
*getAutomationClassByInstance()* : IAutomationClass  
*IAutomationClass*  
*getAutomationClassByName()* : IAutomationClass

**IAutomationEventDescriptor** ■

P: mx.automation  
[eventClassName:](#) String  
[eventType:](#) String  
[name:](#) String  
[getArgDescriptors\(\)](#) : Array  
[record\(\)](#) : Array  
[replay\(\)](#) : Object

**IAutomationManager** ■

P: mx.automation  
[automationEnvironment:](#) Object  
[recording:](#) Boolean  
[replaying:](#) Boolean  
[beginRecording\(\)](#) : void  
[createID\(\)](#) : AutomationID  
[createIDPart\(\)](#) : AutomationIDPart  
[decrementCacheCounter\(\)](#) : int  
[endRecording\(\)](#) : void  
[getAutomationClassName\(\)](#) : String  
[getAutomationName\(\)](#) : String  
[getChildren\(\)](#) : Array  
[getChildrenFromIDPart\(\)](#) : Array  
[getElementFromPoint\(\)](#) : IAutomationObject  
[getParent\(\)](#) : IAutomationObject  
[getProperties\(\)](#) : Array  
[getRectangle\(\)](#) : Array  
[getTabularData\(\)](#) : IAutomationTabularData  
[incrementCacheCounter\(\)](#) : int  
[isSynchronized\(\)](#) : Boolean  
[isVisible\(\)](#) : Boolean  
[recordAutomatableEvent\(\)](#) : void  
[replayAutomatableEvent\(\)](#) : Boolean  
[resolveID\(\)](#) : Array  
[resolveIDPart\(\)](#) : Array  
[resolveIDPartToSingleObject\(\)](#) : IAutomationObject  
[resolveIDToSingleObject\(\)](#) : IAutomationObject  
[showInHierarchy\(\)](#) : Boolean

**IAutomationPropertyDescriptor** ■

P: mx.automation  
[asType:](#) String  
[defaultValue:](#) String  
[forDescription:](#) Boolean  
[forVerification:](#) Boolean  
[name:](#) String

**IAutomationTabularData** ■

P: mx.automation  
[columnNames:](#) Array  
[firstVisibleRow:](#) int  
[lastVisibleRow:](#) int  
[numColumns:](#) int  
[numRows:](#) int  
[getAutomationValueForData\(\)](#) : Array  
[getValues\(\)](#) : Array

**IAutomationMethodDescriptor** ■

P: mx.automation  
[name:](#) String  
[returnType:](#) String  
[getArgDescriptors\(\)](#) : Array  
[record\(\)](#) : Array  
[replay\(\)](#) : Object

**HaloBorder-IAxis****IAutomationMouseSimulator** ■

P: mx.automation  
[getMouseX\(\)](#) : Number  
[getMouseY\(\)](#) : Number

**IAutomationObject** ■

P: mx.automation  
[automationDelegate:](#) Object  
[automationName:](#) String  
[automationTabularData:](#) Object  
[automationValue:](#) Array  
[numAutomationChildren:](#) int  
[showInAutomationHierarchy:](#) Boolean  
[createAutomationIDPart\(\)](#) : Object  
[getAutomationChildAt\(\)](#) : IAutomationObject  
[replayAutomatableEvent\(\)](#) : Boolean  
[resolveAutomationIDPart\(\)](#) : Array

**IAutomationObjectHelper** ■

P: mx.automation  
[recording:](#) Boolean  
[replaying:](#) Boolean  
[addSynchronization\(\)](#) : void  
[getAutomationComposite\(\)](#) : IAutomationObject  
[helpCreateIDPart\(\)](#) : AutomationIDPart  
[helpResolveIDPart\(\)](#) : Array  
[isAutomationComposite\(\)](#) : Boolean  
[replayClick\(\)](#) : Boolean  
[replayClickOffStage\(\)](#) : Boolean  
[replayKeyboardEvent\(\)](#) : Boolean  
[replayKeyDownKeyUp\(\)](#) : Boolean  
[replayMouseEvent\(\)](#) : Boolean

**IAutomationPropertyDescriptor** ■

P: mx.automation  
[asType:](#) String  
[defaultValue:](#) String  
[forDescription:](#) Boolean  
[forVerification:](#) Boolean  
[name:](#) String

**IAutomationTabularData** ■

P: mx.automation  
[columnNames:](#) Array  
[firstVisibleRow:](#) int  
[lastVisibleRow:](#) int  
[numColumns:](#) int  
[numRows:](#) int  
[getAutomationValueForData\(\)](#) : Array  
[getValues\(\)](#) : Array

**IAxis** ■

P: mx.charts.chartClasses  
→ IEventDispatcher  
[baseline:](#) Number  
[chartDataProvider:](#) Object  
[displayName:](#) String  
[title:](#) String  
[unitSize:](#) Number  
[dataChanged\(\)](#) : void  
[filterCache\(\)](#) : void  
[formatForScreen\(\)](#) : String  
[getLabelEstimate\(\)](#) : AxisLabelSet  
[getLabels\(\)](#) : AxisLabelSet  
[invertTransform\(\)](#) : Object  
[mapCache\(\)](#) : void  
[preferDropLabels\(\)](#) : Boolean  
[reduceLabels\(\)](#) : AxisLabelSet  
[registerDataTransform\(\)](#) : void  
[transformCache\(\)](#) : void  
[unregisterDataTransform\(\)](#) : void  
[update\(\)](#) : void

**IAxisRenderer** ■  
P: mx.charts.chartClasses  
→ IUIComponent

axis: IAxis  
gutters: Rectangle  
heightLimit: Number  
horizontal: Boolean  
minorTicks: Array  
otherAxes: Array  
placement: String  
ticks: Array  
adjustGutters(): Rectangle  
chartStateChanged(): void

**IBar** ■  
P: mx.charts.chartClasses  
barWidthRatio: Number  
maxBarWidth: Number  
offset: Number

**IBitmapDrawable**  
P: flash.display

**IBorder** ■  
P: mx.core  
borderMetrics: EdgeMetrics

**IBrowserManager** ■  
P: mx.managers  
→ IEventDispatcher  
base: String  
fragment: String  
title: String  
url: String  
init(): void  
initForHistoryManager(): void  
setFragment(): void  
setTitle(): void

**IButton** ■  
P: mx.core  
→ IUIComponent  
emphasized: Boolean  
callLater(): void

**ICartElement** ■  
P: mx.charts.chartClasses  
→ IFlexDisplayObject  
chartDataProvider: Object  
dataTransform: DataTransform  
labelContainer: Sprite  
chartStateChanged(): void  
claimStyles(): uint  
collectTransitions(): void  
describeData(): Array  
findDataPoints(): Array  
mappingChanged(): void

**ICartElement2** ■  
P: mx.charts.chartClasses  
→ ICartElement  
dataToLocal(): Point  
getAllDataPoints(): Array  
localToData(): Array

**ICildList** ■  
P: mx.core  
numChildren: int  
addChild(): DisplayObject  
addChildAt(): DisplayObject  
contains(): Boolean  
getChildAt(): DisplayObject  
getChildByName(): DisplayObject  
getChildIndex(): int

getObjectsUnderPoint(): Array  
removeChild(): DisplayObject  
removeChildAt(): DisplayObject  
setChildIndex(): void

**ICollectionView** ■  
P: mx.collections  
→ IEventDispatcher  
filterFunction: Function  
length: int  
sort: Sort  
contains(): Boolean  
createCursor(): IViewCursor  
disableAutoUpdate(): void  
enableAutoUpdate(): void  
itemUpdated(): void  
refresh(): Boolean

**IColumn** ■  
P: mx.charts.chartClasses  
columnWidthRatio: Number  
maxColumnWidth: Number  
offset: Number

**Icon** ●  
P: flash.desktop  
→ EventDispatcher  
bitmaps: Array

**IConstraintClient** ■  
P: mx.core  
getConstraintValue(): \*  
setConstraintValue(): void

**IConstraintLayout** ■  
P: mx.containers.utilityClasses  
constraintColumns: Array  
constraintRows: Array

**IContainer** ■  
P: mx.core  
→ IUIComponent  
buttonMode: Boolean  
creatingContentPane: Boolean  
defaultButton: IFlexDisplayObject  
doubleClickEnabled: Boolean  
dropTarget: DisplayObject  
focusManager: IFocusManager  
focusRect: Object  
graphics: Graphics  
hitArea: Sprite  
horizontalScrollPosition: Number  
mouseChildren: Boolean  
mouseEnabled: Boolean  
numChildren: int  
soundTransform: SoundTransform  
tabChildren: Boolean  
tabEnabled: Boolean  
tabIndex: int  
textSnapshot: TextSnapshot  
useHandCursor: Boolean  
verticalScrollPosition: Number  
viewMetrics: EdgeMetrics  
addChild(): DisplayObject  
addChildAt(): DisplayObject  
areInaccessibleObjectsUnderPoint(): Boolean  
contains(): Boolean  
getChildAt(): DisplayObject  
getChildByName(): DisplayObject  
getChildIndex(): int  
getObjectsUnderPoint(): Array  
removeChild(): DisplayObject  
removeChildAt(): DisplayObject

setChildIndex(): void  
startDrag(): void  
stopDrag(): void  
swapChildren(): void  
swapChildrenAt(): void

**ID3Info**  
P: flash.media  
→ Object  
album: String  
artist: String  
comment: String  
genre: String  
songName: String  
track: String  
year: String

**IDataInput**  
P: flash.utils  
bytesAvailable: uint  
endian: String  
objectEncoding: uint  
readBoolean(): Boolean  
readByte(): int  
readBytes(): void  
readDouble(): Number  
readFloat(): Number  
readInt(): int  
readMultiByte(): String  
readObject(): \*  
readShort(): int  
readUnsignedByte(): uint  
readUnsignedInt(): uint  
readUnsignedShort(): uint  
readUTF(): String  
readUTFBytes(): String

**IDataOutput**  
P: flash.utils  
endian: String  
objectEncoding: uint  
writeBoolean(): void  
writeByte(): void  
writeBytes(): void  
writeDouble(): void  
writeFloat(): void  
writeInt(): void  
writeMultiByte(): void  
writeObject(): void  
writeShort(): void  
writeUnsignedInt(): void  
writeUTF(): void  
writeUTFBytes(): void

**IDataRenderer** ■  
P: mx.controls.listClasses

**IDataRenderer** ■  
P: mx.core  
data: Object

**IDeferredInstance** ■  
P: mx.core  
getInstance(): Object

**IDeferredInstantiationUIComponent** ■  
P: mx.core  
→ IUIComponent  
cacheHeuristic: Boolean  
cachePolicy: String  
descriptor: UIComponentDescriptor  
id: String  
createReferenceOnParentDocument(): void

deleteReferenceOnParentDocument(): void  
executeBindings(): void  
swapChildren(): void  
registerEffects(): void

**IDropInListItemRenderer** ■  
P: mx.controls.listClasses  
Implements the *listData* property  
listData: BaseListData

**IDynamicPropertyOutput**  
P: flash.net  
writeDynamicProperty(): void

**IDynamicPropertyWriter**  
P: flash.net  
writeDynamicProperties(): void

**IEffect** ■  
P: mx.effects  
→ IAbstractEffect  
className: String  
customFilter: EffectTargetFilter  
duration: Number  
effectTargetHost: IEffectTargetHost  
filter: String  
hideFocusRing: Boolean  
isPlaying: Boolean  
perElementOffset: Number  
relevantProperties: Array  
relevantStyles: Array  
target: Object  
targets: Array  
triggerEvent: Event  
captureEndValues(): void  
captureMoreStartValues(): void  
captureStartValues(): void  
createInstance(): IEffectInstance  
createInstances(): Array  
deleteInstance(): void  
end(): void  
getAffectedProperties(): Array  
pause(): void  
play(): Array  
resume(): void  
reverse(): void  
stop(): void

**IEffectInstance** ■  
P: mx.effects  
className: String  
duration: Number  
effect: IEffect  
effectTargetHost: IEffectTargetHost  
hideFocusRing: Boolean  
playheadTime: Number  
propertyChanges: PropertyChanges  
repeatCount: int  
repeatDelay: int  
startDelay: int  
suspendBackgroundProcessing: Boolean  
target: Object  
triggerEvent: Event  
end(): void  
finishEffect(): void  
finishRepeat(): void  
initEffect(): void  
pause(): void  
play(): void  
resume(): void  
reverse(): void  
startEffect(): void  
stop(): void

**IEffectTargetHost** ■  
P: mx.effects  
addDataEffectItem(): void  
getRendererSemanticValue(): Object  
removeDataEffectItem(): void  
unconstrainRenderer(): void

**IEventDispatcher**  
P: flash.events  
addEventListener(): void  
dispatchEvent(): Boolean  
hasEventListener(): Boolean  
removeEventListener(): void  
willTrigger(): Boolean

**IExternalizable**  
P: flash.utils  
readExternal(): void  
writeExternal(): void

**IFactory** ■  
P: mx.core  
newInstance(): \*

**IFill** ■  
P: mx.graphics  
begin(): void  
end(): void

**IFlexAsset** ■  
P: mx.core

**IFlexContextMenu** ■  
P: mx.controls  
setContextMenu(): void  
unsetContextMenu(): void

**IFlexDisplayObject** ■  
P: mx.core  
→ IBitmapDrawable  
accessibilityProperties: Accessibility-Properties  
alpha: Number  
blendMode: String  
cacheAsBitmap: Boolean  
filters: Array  
height: Number  
loaderInfo: LoaderInfo  
mask: DisplayObject  
measuredHeight: Number  
measuredWidth: Number  
mouseX: Number  
mouseY: Number  
name: String  
opaqueBackground: Object  
parent: DisplayObjectContainer  
root: DisplayObject  
rotation: Number  
scale9Grid: Rectangle  
scaleX: Number  
scaleY: Number  
scrollRect: Rectangle  
stage: Stage  
transform: Transform  
visible: Boolean  
width: Number  
x: Number  
y: Number  
getBounds(): Rectangle  
getRect(): Rectangle  
globalToLocal(): Point  
hitTestObject(): Boolean  
hitTestPoint(): Boolean

localToGlobal(): Point  
move(): void  
setActualSize(): void  
**k**  
**IFlexModule** ■  
P: mx.core

**IFlexModuleFactory** ■  
P: mx.core  
create(): Object  
info(): Object

**IFocusManager** ■  
P: mx.managers  
defaultButton: IButton  
defaultButtonEnabled: Boolean  
focusPane: Sprite  
nextTabIndex: int  
showFocusIndicator: Boolean  
activate(): void  
deactivate(): void  
findFocusManagerComponent(): IFocusManagerComponent  
getFocus(): IFocusManagerComponent  
getNextFocusManagerComponent(): IFocusManagerComponent  
hideFocus(): void  
setFocus(): void  
showFocus(): void

**IFocusManagerComplexComponent** ■  
P: mx.managers  
→ IFocusManagerComponent  
hasFocusableContent: Boolean  
assignFocus(): void

**IFocusManagerComponent** ■  
P: mx.managers  
focusEnabled: Boolean  
mouseFocusEnabled: Boolean  
tabEnabled: Boolean  
tabIndex: int  
drawFocus(): void  
setFocus(): void

**IFocusManagerContainer** ■  
P: mx.managers  
→ IEventDispatcher  
focusManager: IFocusManager  
systemManager: ISystemManager  
contains(): Boolean

**IFocusManagerGroup** ■  
P: mx.managers  
groupName: String  
selected: Boolean

**IFontContextComponent** ■  
P: mx.core  
fontContext: IFlexModuleFactory

**IGroupingCollection** ■  
P: mx.collections  
→ IHierarchicalData  
grouping: Grouping  
cancelRefresh(): void  
refresh(): Boolean

**IHierarchicalCollectionViewCursor**  
P: mx.collections  
→ IViewCursor  
currentDepth: int

**IHierarchicalCollectionView** ■  
P: mx.collections  
→ ICollectionView  
hasRoot: Boolean  
openNodes: Object  
showRoot: Boolean  
source: IHierarchicalData  
addChild(): Boolean  
addChildAt(): Boolean  
closeNode(): void  
getChildren(): ICollectionView  
getNodeDepth(): int  
getParentItem(): \*  
openNode(): void  
removeChild(): Boolean  
removeChildAt(): Boolean

**IHierarchicalData** ■  
P: mx.collections  
→ IEventDispatcher  
canHaveChildren(): Boolean  
getChildren(): Object  
getData(): Object  
getRoot(): Object  
hasChildren(): Boolean

**IHistoryManagerClient** ■  
P: mx.managers  
loadState(): void  
saveState(): Object  
toString(): String

**ImageEncoder** ■  
P: mx.graphics.codec  
contentType: String  
encode(): ByteArray  
encodeByteArray(): ByteArray

**IMESupport** ■  
P: mx.core  
imeMode: String

**Invalidating** ■  
P: mx.core  
invalidateDisplayList(): void  
invalidateProperties(): void  
invalidateSize(): void  
validateNow(): void

**LayoutManager** ■  
P: mx.managers  
→ IEventDispatcher  
usePhasedInstantiation: Boolean  
invalidateDisplayList(): void  
invalidateProperties(): void  
invalidateSize(): void  
isInvalid(): Boolean  
validateClient(): void  
validateNow(): void

**LayoutManagerClient** ■  
P: mx.managers  
→ IEventDispatcher  
initialized: Boolean  
nestLevel: int  
processedDescriptors: Boolean  
updateCompletePendingFlag: Boolean  
validateDisplayList(): void  
validateProperties(): void  
validateSize(): void

**IllegalOperationError**  
P: flash.errors  
→ Error  
IllegalOperationError()

**IList** ■  
P: mx.collections  
→ IEventDispatcher  
length: int  
addItem(): void  
addItemAt(): void  
getItemAt(): Object  
getItemIndex(): int  
itemUpdated(): void  
removeAll(): void  
removeItemAt(): Object  
setItemAt(): Object  
toArray(): Array

**IListItemRenderer** ■  
P: mx.controls.listClasses  
→ IDataRenderer

**ILogger** ■  
P: mx.logging  
→ IEventDispatcher  
category: String  
debug(): void  
error(): void  
fatal(): void  
info(): void  
log(): void  
warn(): void

**LoggingTarget** ■  
P: mx.logging  
filters: Array  
level: int  
addLogger(): void  
removeLogger(): void

**Image** ■  
P: mx.controls  
→ SWFLoader  
IDataRenderer  
IDropInListItemRenderer  
IListItemRenderer  
data: Object  
listData: BaseListData  
Image(): void

**ImageSnapshot** ■  
P: mx.graphics  
→ Object  
contentType: String  
data: ByteArray  
defaultEncoder: Class  
height: int  
MAX\_BITMAP\_DIMENSION: int  
properties: Object  
width: int  
captureBitmapData(): BitmapData  
captureImage(): ImageSnapshot  
encodeImageAsBase64(): String  
ImageSnapshot()

**IME**  
P: flash.system  
→ EventDispatcher  
conversionMode: String  
enabled: Boolean  
doConversion(): void  
setCompositionString(): void

**IMEConversionMode**  
P: flash.system  
→ Object  
ALPHANUMERIC\_FULL: String  
ALPHANUMERIC\_HALF: String



[CHINESE](#): String  
[JAPANESE\\_HIRAGANA](#): String  
[JAPANESE\\_KATAKANA\\_FULL](#): String  
[JAPANESE\\_KATAKANA\\_HALF](#): String  
[KOREAN](#): String  
[UNKNOWN](#): String

**IMEEvent**

P: flash.events  
 → TextEvent  
[IME\\_COMPOSITION](#): String  
 clone(): Event  
 IMEEvent()  
 toString(): String

**IMenuBarItemRenderer**

P: mx.controls.menuClasses  
 → IDataRenderer  
 menuBar: MenuBar  
 menuBarItemIndex: int  
 menuBarItemState: String

**IMenuDataDescriptor**

P: mx.controls.menuClasses  
 addChildAt(): Boolean  
 getChildren(): ICollectionView  
 getData(): Object  
 getGroupName(): String  
 getType(): String  
 hasChildren(): Boolean  
 isBranch(): Boolean  
 isEnabled(): Boolean  
 isToggled(): Boolean  
 removeChildAt(): Boolean  
 setEnabled(): void  
 setToggled(): void

**IMenuItemRenderer**

P: mx.controls.menuClasses  
 measuredBranchIconWidth: Number  
 measuredIconWidth: Number  
 measuredTypeIconWidth: Number  
 menu: Menu

**IMessage**

P: mx.messaging.messages  
 body: Object  
 clientId: String  
 destination: String  
 headers: Object  
 messageId: String  
 timestamp: Number  
 timeToLive: Number  
 toString(): String

**IModuleInfo**

P: mx.modules  
 → IEventDispatcher  
 data: Object  
 error: Boolean  
 factory: IFlexModuleFactory  
 loaded: Boolean  
 ready: Boolean  
 setup: Boolean  
 url: String  
 load(): void  
 publish(): void  
 release(): void  
 unload(): void

**IMXMLObject**

P: mx.core  
 initialized(): void

**IMXMLSupport**

P: mx.rpc.mxml  
 concurrency: String  
 showBusyCursor: Boolean

**IOError**

P: flash.errors  
 → Error  
 IOError()

**IndexChangedEvent**

P: mx.events  
 → Event  
[CHANGE](#): String  
[CHILD\\_INDEX\\_CHANGE](#): String  
[HEADER\\_SHIFT](#): String  
 newIndex: Number  
 oldIndex: Number  
 relatedObject: DisplayObject  
 triggerEvent: Event  
 IndexChangedEvent()

**InstanceCache**

P: mx.charts.chartClasses  
 → Object  
 count: int  
 creationCallback: Function  
 discard: Boolean  
 factory: IFactory  
 format: TextFormat  
 hide: Boolean  
 insertPosition: int  
 instances: Array  
 properties: Object  
 remove: Boolean  
 InstanceCache()

**int**

P: Top Level  
 → Object  
[MAX\\_VALUE](#): int  
[MIN\\_VALUE](#): int  
 int()  
 toExponential(): String  
 toFixed(): String  
 toPrecision(): String  
 toString(): String  
 valueOf(): int

**InteractiveIcon**

P: flash.desktop  
 → Icon  
 bitmaps: Array  
 height: int  
 width: int

**InteractiveObject**

P: flash.display  
 → Display Object  
 contextMenu: NativeMenu  
 doubleClickEnabled: Boolean  
 focusRect: Object  
 mouseEnabled: Boolean  
 tabEnabled: Boolean  
 tabIndex: int  
 InteractiveObject(): void

**InterpolationMethod**

P: flash.display  
 → Object  
[LINEAR\\_RGB](#): String  
[RGB](#): String

**InvalidCategoryError**

P: mx.logging.errors  
 → Error  
 InvalidCategoryError()  
 toString(): String

**InvalidChannelError**

P: mx.messaging.errors  
 → ChannelError

**InvalidDestinationError**

P: mx.messaging.errors  
 → ChannelError

**InvalidFilterError**

P: mx.logging.errors  
 → Error  
 InvalidFilterError()  
 toString(): String

**InvalidSWFError**

P: flash.errors  
 → Error  
 InvalidSWFError()

**InvokeEvent**

P: flash.events  
 → Event  
 arguments: Array  
 currentDirectory: File  
[INVOKE](#): String  
 clone(): Event  
 InvokeEvent()

**InvokeEvent**

P: mx.rpc.events  
 → AbstractEvent  
[INVOKE](#): String  
 InvokeEvent()  
 toString(): String

**IOError**

P: flash.errors  
 → Error  
 IOError()

**IOErrorEvent**

P: flash.events  
 → ErrorEvent  
[IO\\_ERROR](#): String  
 clone(): Event  
 IOErrorEvent()  
 toString(): String

**IOLAPAttribute**

P: mx.olap  
 → IOLAPHierarchy

**IOLAPAxisPosition**

P: mx.olap  
 members: IList

**IOLAPCell**

P: mx.olap  
[formattedValue](#): String  
 value: Number

**IOLAPCube**

P: mx.olap  
 dimensions: IList  
 name: String  
 cancelQuery(): void  
 cancelRefresh(): void  
 execute(): AsyncToken

findDimension(): IOLAPDimension  
 refresh(): void

**IOLAPCustomAggregator**

P: mx.olap  
 computeBegin(): Object  
 computeEnd(): Number  
 computeLoop(): void  
 computeObjectBegin(): Object  
 computeObjectEnd(): Number  
 computeObjectLoop(): void

**IOLAPDimension**

P: mx.olap  
 → IOLAPElement  
 attributes: IList  
 cube: IOLAPCube  
 defaultMember: IOLAPMember  
 hierarchies: IList  
 isMeasure: Boolean  
 members: IList  
 findAttribute(): IOLAPAttribute  
 findHierarchy(): IOLAPHierarchy  
 findMember(): IOLAPMember

**IOLAPElement**

P: mx.olap  
 dimension: IOLAPDimension  
 displayName: String  
 name: String  
 uniqueName: String

**IOLAPHierarchy**

P: mx.olap  
 → IOLAPElement  
 allMemberName: String  
 children: IList  
 defaultMember: IOLAPMember  
 hasAll: Boolean  
 levels: IList  
 members: IList  
 findLevel(): IOLAPLevel  
 findMember(): IOLAPMember

**IOLAPLevel**

P: mx.olap  
 → IOLAPElement  
 child: IOLAPLevel  
 depth: int  
 hierarchy: IOLAPHierarchy  
 members: IList  
 parent: IOLAPLevel  
 findMember(): IList

**IOLAPMember**

P: mx.olap  
 → IOLAPElement  
 children: IList  
 hierarchy: IOLAPHierarchy  
 isAll: Boolean  
 isMeasure: Boolean  
 level: IOLAPLevel  
 parent: IOLAPMember  
 findChildMember(): IOLAPMember

**IOLAPQuery**

P: mx.olap  
 getAxis(): IOLAPQueryAxis  
 setAxis(): void

**IOLAPQueryAxis**

P: mx.olap  
 sets: Array  
 tuples: Array

addMember(): void  
 addSet(): void  
 addTuple(): void

**IOLAPResult**

P: mx.olap  
 axes: Array  
 query: IOLAPQuery  
 getAxis(): IOLAPResultAxis  
 getCell(): IOLAPCell

**IOLAPResultAxis**

P: mx.olap  
[positions](#): IList

**IOLAPSchema**

P: mx.olap  
 cubes: IList  
 createCube(): IOLAPCube  
 getCube(): IOLAPCube

**IOLAPSet**

P: mx.olap  
 addElement(): void  
 addElements(): void  
 addTuple(): void  
 crossJoin(): IOLAPSet  
 hierarchize(): IOLAPSet  
 union(): IOLAPSet

**IOLAPTuple**

P: mx.olap  
[explicitMembers](#): IList  
 addMember(): void  
 addMembers(): void

**IOVERRIDE**

P: mx.states  
 apply(): void  
 initialize(): void  
 remove(): void

**IPreloaderDisplay**

P: mx.preloaders  
 → IEventDispatcher  
 backgroundAlpha: Number  
 backgroundColor: uint  
 backgroundImage: Object  
 backgroundSize: String  
 preloader: Sprite  
 stageHeight: Number  
 stageWidth: Number  
 initialize(): void

**IProgrammaticSkin**

P: mx.core  
 validateDisplayList(): void  
 validateNow(): void

**IPropertyChangeNotifier**

P: mx.core  
 → IEventDispatcher

**IRawChildrenContainer**

P: mx.core  
[rawChildren](#): IChildList

**IRectangularBorder**

P: mx.core  
 → IBorder  
[backgroundImageBounds](#): Rectangle  
[hasBackgroundImage](#): Boolean  
[layoutBackgroundImage](#): void

**IRpeater**

P: mx.core  
 container: IContainer  
 count: int  
 currentIndex: int  
 currentItem: Object  
 dataProvider: Object  
 recycleChildren: Boolean  
 startIndex: int  
 executeChildBindings(): void  
 initializeRepeater(): void

**IRpeaterClient**

P: mx.core  
 instanceIndices: Array  
 isDocument: Boolean  
 repeaterIndices: Array  
 repeaters: Array  
 initializeRepeaterArrays(): void

**IResourceBundle**

P: mx.resources  
[bundleName](#): String  
 content: Object  
 locale: String

**IResourceManager**

P: mx.resources  
 → IEventDispatcher  
[localeChain](#): Array  
 addResourceBundle(): void  
 findResourceBundleWithResource(): IResourceBundle  
 getBoolean(): Boolean  
 getBundleNamesForLocale(): Array  
 getClass(): Class  
 getInt(): int  
 getLocales(): Array  
 getNumber(): Number  
 getObject(): \*  
 getResourceBundle(): IResourceBundle  
 getString(): String  
 getStringArray(): Array  
 getUint(): uint  
 installCompiledResourceBundles(): void  
 loadResourceModule(): IEventDispatcher  
 removeResourceBundle(): void  
 removeResourceBundlesForLocale(): void  
 unloadResourceModule(): void  
 update(): void

**IResponder**

P: mx.rpc  
 fault(): void  
 result(): void

**Iris**

P: mx.effects  
 → MaskEffect

**IrisInstance**

P: mx.effects.effectClasses  
 → MaskEffectInstance

**ISimpleStyleClient**

P: mx.styles  
 styleName: Object  
 styleChanged(): void

**ISOAPDecoder**

P: mx.rpc.soap  
[forcePartArrays](#): Boolean  
[headerFormat](#): String  
[ignoreWhitespace](#): Boolean  
[multiplePartsFormat](#): String  
[resultFormat](#): String  
[wsdlOperation](#): WSDLOperation  
 decodeResponse(): SOAPResult

**ISOAPEncoder**

P: mx.rpc.soap  
[ignoreWhitespace](#): Boolean  
[wsdlOperation](#): WSDLOperation  
 encodeRequest(): XML

**IStackable**

P: mx.charts.chartClasses  
 stacker: StackedSeries  
[stackTotals](#): Dictionary  
 stack(): Number

**IStackable2**

P: mx.charts.chartClasses  
 → Istackable  
 stackAll(): Object

**IStateClient**

P: mx.core  
[currentState](#): String

**IStroke**

P: mx.graphics  
 weight: Number  
 apply(): void

**IStyleClient**

P: mx.styles  
 → ISimpleStyleClient  
[className](#): String  
[inheritingStyles](#): Object  
[nonInheritingStyles](#): Object  
[styleDeclaration](#): CSSStyleDeclaration  
 clearStyle(): void  
 getClassStyleDeclarations(): Array  
 getStyle(): \*  
 notifyStyleChangeInChildren(): void  
 regenerateStyleCache(): void  
 registerEffects(): void  
 setStyle(): void

**IStyleModule**

P: mx.styles  
 unload(): void

**ISystemManager**

P: mx.managers  
 → IEventDispatcher  
[cursorChildren](#): IChildList  
 document: Object  
 focusPane: Sprite  
[loaderInfo](#): LoaderInfo  
 numModalWindows: int  
 popUpChildren: IChildList  
[rawChildren](#): IChildList  
 screen: Rectangle  
 stage: Stage  
[toolTipChildren](#): IChildList  
 topLevelSystemManager: ISystemManager  
 activate(): void  
 addFocusManager(): void  
 deactivate(): void  
 getDefinitionByName(): Object

isFontFaceEmbedded(): Boolean  
 isTopLevel(): Boolean  
 removeFocusManager(): void

**ItemClickEvent**

P: mx.events  
 → Event  
 index: int  
 item: Object  
[ITEM\\_CLICK](#): String  
 label: String  
 relatedObject: InteractiveObject  
 ItemClickEvent()

**ItemPendingError**

P: mx.collections.errors  
 → Error  
 + responders: Array  
 + addResponder(IResponder): void  
 + ItemPendingError(String): void

**ItemResponder**

P: mx.collections  
 → Object  
 IResponder  
 fault(): void  
 ItemResponder()  
 result(): void

**IToolTip**

P: mx.core  
 → IUIComponent  
 screen: Rectangle  
 text: String

**IToolTipManagerClient**

→ FlexDisplayObject  
 P: mx.managers

**ITreeDataDescriptor**

P: mx.controls.treeClasses  
 addChildAt(): Boolean  
 getChildren(): ICollectionView  
 getData(): Object  
 hasChildren(): Boolean  
 isBranch(): Boolean  
 removeChildAt(): Boolean

**ITreeDataDescriptor2**

P: mx.controls.treeClasses  
 → ITreeDataDescriptor  
 getHierarchicalCollectionAdaptor(): ICollectionView  
 getNodeDepth(): int  
 getParent(): Object

**IUIComponent**

P: mx.core  
 → IFlexDisplayObject  
[baselinePosition](#): Number  
 document: Object  
 enabled: Boolean  
[explicitHeight](#): Number  
[explicitMaxHeight](#): Number  
[explicitMaxWidth](#): Number  
[explicitMinHeight](#): Number  
[explicitMinWidth](#): Number  
[explicitWidth](#): Number  
 focusPane: Sprite  
 includeInLayout: Boolean  
 isPopUp: Boolean  
[maxHeight](#): Number  
[maxWidth](#): Number  
[measuredMinHeight](#): Number

measuredMinWidth: Number  
minHeight: Number  
minWidth: Number  
owner: DisplayObjectContainer  
percentHeight: Number  
percentWidth: Number  
systemManager: ISystemManager  
tweeningProperties: Array  
getExplicitOrMeasuredHeight() : Number  
getExplicitOrMeasuredWidth() : Number  
initialize() : void  
owns() : Boolean  
parentChanged() : void  
setVisible() : void

**UID** ■

P: mx.core  
uid: String

**IUITextField** ■

P: mx.core  
→ IIMESupport  
alwaysShowSelection: Boolean  
antiAliasType: String  
autoSize: String  
background: Boolean  
backgroundColor: uint  
border: Boolean  
borderColor: uint  
bottomScrollV: int  
caretIndex: int  
condenseWhite: Boolean  
defaultTextFormat: TextFormat  
displayAsPassword: Boolean  
doubleClickEnabled: Boolean  
embedFonts: Boolean  
focusRect: Object  
gridFitType: String  
htmlText: String  
ignorePadding: Boolean  
inheritingStyles: Object  
length: int  
maxChars: int  
maxScrollH: int  
maxScrollV: int  
mouseEnabled: Boolean  
mouseWheelEnabled: Boolean  
multiline: Boolean  
nestLevel: int  
nonInheritingStyles: Object  
nonZeroTextHeight: Number  
numLines: int  
restrict: String  
scrollH: int  
scrollV: int  
selectable: Boolean  
selectionBeginIndex: int  
selectionEndIndex: int  
sharpness: Number  
styleSheet: StyleSheet  
tabEnabled: Boolean  
tabIndex: int  
text: String  
textColor: uint  
textHeight: Number  
textWidth: Number  
thickness: Number  
type: String  
useRichTextClipboard: Boolean  
wordWrap: Boolean  
appendText() : void  
getCharBoundaries() : Rectangle

getCharIndexAtPoint() : int  
getFirstCharInParagraph() : int  
getImageReference() : DisplayObject  
getLineIndexAtPoint() : int  
getLineIndexOfChar() : int  
getLineLength() : int  
getLineMetrics() : TextLineMetrics  
getLineOffset() : int  
getLineText() : String  
getParagraphLength() : int  
getStyle() : \*  
getTextFormat() : TextFormat  
getUITextFormat() : UITextFormat  
replaceSelectedText() : void  
replaceText() : void  
setColor() : void  
setFocus() : void  
setSelection() : void  
setTextFormat() : void  
truncateToFit() : Boolean

**IURIDereferencer** ●

P: flash.security  
dereference() : IDataInput

**IValidatorListener** ■

P: mx.validators  
errorString: String  
validationSubField: String  
validationResultHandler() : void

**IViewCursor** ■

P: mx.collections  
→ IEventDispatcher  
afterLast: Boolean  
beforeFirst: Boolean  
bookmark: CursorBookmark  
current: Object  
view: ICollectionView  
findAny() : Boolean  
findFirst() : Boolean  
findLast() : Boolean  
insert() : void  
moveNext() : Boolean  
movePrevious() : Boolean  
remove() : Object  
seek() : void

**IWindow** ■●

P: mx.core  
maximizable: Boolean  
minimizable: Boolean  
nativeWindow: NativeWindow  
resizable: Boolean  
status: String  
systemChrome: String  
title: String  
titleIcon: Class  
transparent: Boolean  
type: String  
visible: Boolean  
close() : void  
maximize() : void  
minimize() : void  
restore() : void

**IXMLDecoder** ■

P: mx.rpc.xml  
makeObjectBindable: Boolean  
recordXSISType: Boolean  
typeRegistry: SchemeTypeRegistry  
decode():\*  
reset():void

**IXMLEncoder** ■

P: mx.rpc.xml  
xmlSpecialCharsFilter: Function  
encode():XMLList  
reset():void

**IXMLSchemaInstance** ■

P: mx.rpc.xml  
xsiType: QName

**JointStyle**

P: flash.display  
→ Object  
BEVEL: String  
MITER: String  
ROUND: String

**JPEGEncoder** ■

P: mx.graphics.codec  
→ Object  
JImageEncoder  
contentType: String  
encode() : ByteArray  
encodeByteArray() : ByteArray  
JPEGEncoder() : void

**Keyboard**

P: flash.ui  
→ Object  
A: uint  
ALTERNATE: uint  
B: uint  
BACKQUOTE: uint  
BACKSLASH: uint  
BACKSPACE: uint  
C: uint  
CAPS\_LOCK: uint  
capsLock: Boolean  
CharCodeStrings: Array  
COMMA: uint  
COMMAND: uint  
CONTROL: uint  
D: uint  
DELETE: uint  
DOWN: uint  
E: uint  
END: uint  
ENTER: uint  
EQUAL: uint  
ESCAPE: uint  
F: uint  
F1: uint  
F10: uint  
F11: uint  
F12: uint  
F13: uint  
F14: uint  
F15: uint  
F2: uint  
F3: uint  
F4: uint  
F5: uint  
F6: uint  
F7: uint  
F8: uint  
F9: uint  
G: uint  
H: uint  
HOME: uint  
I: uint  
INSERT: uint  
J: uint  
K: uint  
KEYNAME\_BEGIN: String

KEYNAME\_BREAK: String  
KEYNAME\_CLEARDISPLAY: String  
KEYNAME\_CLEARLINE: String  
KEYNAME\_DELETE: String  
KEYNAME\_DELETECHAR: String  
KEYNAME\_DELETELINE: String  
KEYNAME\_DOWNARROW: String  
KEYNAME\_END: String  
KEYNAME\_EXECUTE: String  
KEYNAME\_F1: String  
KEYNAME\_F10: String  
KEYNAME\_F11: String  
KEYNAME\_F12: String  
KEYNAME\_F13: String  
KEYNAME\_F14: String  
KEYNAME\_F15: String  
KEYNAME\_F16: String  
KEYNAME\_F17: String  
KEYNAME\_F18: String  
KEYNAME\_F19: String  
KEYNAME\_F2: String  
KEYNAME\_F20: String  
KEYNAME\_F21: String  
KEYNAME\_F22: String  
KEYNAME\_F23: String  
KEYNAME\_F24: String  
KEYNAME\_F25: String  
KEYNAME\_F26: String  
KEYNAME\_F27: String  
KEYNAME\_F28: String  
KEYNAME\_F29: String  
KEYNAME\_F3: String  
KEYNAME\_F30: String  
KEYNAME\_F31: String  
KEYNAME\_F32: String  
KEYNAME\_F33: String  
KEYNAME\_F34: String  
KEYNAME\_F35: String  
KEYNAME\_F4: String  
KEYNAME\_F5: String  
KEYNAME\_F6: String  
KEYNAME\_F7: String  
KEYNAME\_F8: String  
KEYNAME\_F9: String  
KEYNAME\_FIND: String  
KEYNAME\_HELP: String  
KEYNAME\_HOME: String  
KEYNAME\_INSERT: String  
KEYNAME\_INSERTCHAR: String  
KEYNAME\_INSERTLINE: String  
KEYNAME\_LEFTARROW: String  
KEYNAME\_MENU: String  
KEYNAME\_MODESWITCH: String  
KEYNAME\_NEXT: String  
KEYNAME\_PAGEDOWN: String  
KEYNAME\_PAGEUP: String  
KEYNAME\_PAUSE: String  
KEYNAME\_PREV: String  
KEYNAME\_PRINT: String  
KEYNAME\_PRINTSCREEN: String  
KEYNAME\_REDO: String  
KEYNAME\_RESET: String  
KEYNAME\_RIGHTARROW: String  
KEYNAME\_SCROLLLOCK: String  
KEYNAME\_SELECT: String  
KEYNAME\_STOP: String  
KEYNAME\_SYSREQ: String  
KEYNAME\_SYSTEM: String  
KEYNAME\_UNDO: String  
KEYNAME\_UPARROW: String  
KEYNAME\_USER: String  
L: uint  
LEFT: uint  
LEFTBRACKET: uint

M: uint  
MINUS: uint  
N: uint  
NUMBER\_0: uint  
NUMBER\_1: uint  
NUMBER\_2: uint  
NUMBER\_3: uint  
NUMBER\_4: uint  
NUMBER\_5: uint  
NUMBER\_6: uint  
NUMBER\_7: uint  
NUMBER\_8: uint  
NUMBER\_9: uint  
numLock: Boolean  
NUMPAD: uint  
NUMPAD\_0: uint  
NUMPAD\_1: uint  
NUMPAD\_2: uint  
NUMPAD\_3: uint  
NUMPAD\_4: uint  
NUMPAD\_5: uint  
NUMPAD\_6: uint  
NUMPAD\_7: uint  
NUMPAD\_8: uint  
NUMPAD\_9: uint  
NUMPAD\_ADD: uint  
NUMPAD\_DECIMAL: uint  
NUMPAD\_DIVIDE: uint  
NUMPAD\_ENTER: uint  
NUMPAD\_MULTIPLY: uint  
NUMPAD\_SUBTRACT: uint  
O: uint  
P: uint  
PAGE\_DOWN: uint  
PAGE\_UP: uint  
PERIOD: uint  
Q: uint  
QUOTE: uint  
R: uint  
RIGHT: uint  
RIGHTBRACKET: uint  
S: uint  
SEMICOLON: uint  
SHIFT: uint  
SLASH: uint  
SPACE: uint  
STRING\_BEGIN: String  
STRING\_BREAK: String  
STRING\_CLEARDISPLAY: String  
STRING\_CLEARLINE: String  
STRING\_DELETE: String  
STRING\_DELETECHAR: String  
STRING\_DELETELINE: String  
STRING\_DOWNARROW: String  
STRING\_END: String  
STRING\_EXECUTE: String  
STRING\_F1: String  
STRING\_F10: String  
STRING\_F11: String  
STRING\_F12: String  
STRING\_F13: String  
STRING\_F14: String  
STRING\_F15: String  
STRING\_F16: String  
STRING\_F17: String  
STRING\_F18: String  
STRING\_F20: String  
STRING\_F21: String  
STRING\_F22: String  
STRING\_F23: String  
STRING\_F24: String  
STRING\_F25: String

STRING\_F26: String  
STRING\_F27: String  
STRING\_F28: String  
STRING\_F29: String  
STRING\_F3: String  
STRING\_F30: String  
STRING\_F31: String  
STRING\_F32: String  
STRING\_F33: String  
STRING\_F34: String  
STRING\_F35: String  
STRING\_F4: String  
STRING\_F5: String  
STRING\_F6: String  
STRING\_F7: String  
STRING\_F8: String  
STRING\_F9: String  
STRING\_FIND: String  
STRING\_HELP: String  
STRING\_HOME: String  
STRING\_INSERT: String  
STRING\_INSERTCHAR: String  
STRING\_INSERTLINE: String  
STRING\_LEFTARROW: String  
STRING\_MENU: String  
STRING\_MODESWITCH: String  
STRING\_NEXT: String  
STRING\_PAGEDOWN: String  
STRING\_PAGEUP: String  
STRING\_PAUSE: String  
STRING\_PREV: String  
STRING\_PRINT: String  
STRING\_PRINTSCREEN: String  
STRING\_REDO: String  
STRING\_RESET: String  
STRING\_RIGHTARROW: String  
STRING\_SCROLLLOCK: String  
STRING\_SELECT: String  
STRING\_STOP: String  
STRING\_SYSREQ: String  
STRING\_SYSTEM: String  
STRING\_UNDO: String  
STRING\_UPARROW: String  
STRING\_USER: String  
T: uint  
TAB: uint  
U: uint  
UP: uint  
V: uint  
W: uint  
X: uint  
Y: uint  
Z: uint  
isAccessible() : Boolean

**KeyboardEvent**

P: flash.events  
→ Event  
altKey: Boolean  
charCode: uint  
commandKey: Boolean  
controlKey: Boolean  
ctrlKey: Boolean  
KEY\_DOWN: String  
KEY\_UP: String  
keyCode: uint  
keyLocation: uint  
shiftKey: Boolean  
clone() : Event  
KeyboardEvent()  
toString() : String  
updateAfterEvent() : void

**KeyLocation**

P: flash.ui  
→ Object  
LEFT: uint  
NUM\_PAD: uint  
RIGHT: uint  
STANDARD: uint

**Label** ■

P: mx.controls  
→ UIComponent  
IDataRenderer  
IDropInListItemRenderer  
IListItemRenderer  
IFontContextComponent  
condenseWhite: Boolean  
data: Object  
fontContext: IFlexModuleFactory  
htmlText: String  
listData: BaseListData  
selectable: Boolean  
text: String  
textField: IUITextField  
textHeight: Number  
textWidth: Number  
truncateToFit: Boolean  
getLineMetrics() : TextLineMetrics  
Label()

**LabelAutomationImpl** ■

P: mx.automation.delegates.  
controls  
→ UIComponentAutomationImpl  
init() : void  
LabelAutomationImpl()

**LayoutContainer** ■

P: mx.core  
→ Container  
IConstraintLayout  
boxLayoutClass: Class  
canvasLayoutClass: Class  
constraintColumns: Array  
constraintRows: Array  
layout: String  
LayoutContainer()

**LayoutManager** ■

P: mx.managers  
→ EventDispatcher  
LayoutManager  
usePhasedInstantiation: Boolean  
getInstance() : LayoutManager  
invalidateDisplayList() : void  
invalidateProperties() : void  
invalidateSize() : void  
isInvalid() : Boolean  
LayoutManager()  
validateClient() : void  
validateNow() : void

**Legend** ■

P: mx.charts  
→ Tile  
dataProvider: Object  
legendItemClass: Class  
commitProperties() : void  
Legend()  
measure() : void  
updateDisplayList() : void

**LegendAutomationImpl** ■

P: mx.automation.delegates.charts  
→ ContainerAutomationImpl  
init() : void  
LegendAutomationImpl()

**LegendData** ■

P: mx.charts.chartClasses  
→ Object  
aspectRatio: Number  
element: IChartElement  
label: String  
marker: IFlexDisplayObject  
LegendData()

**LegendItem** ■

P: mx.charts  
→ UIComponent  
element: IChartElement  
label: String  
legendData: Object  
markerAspectRatio: Number  
source: Object  
LegendItem()

**LegendItemAutomationImpl** ■

P: mx.automation.delegates.charts  
→ UIComponentAutomationImpl  
init() : void  
LegendItemAutomationImpl()

**LegendMouseEvent** ■

P: mx.charts.events  
→ MouseEvent  
item: LegendItem  
ITEM\_CLICK: String  
ITEM\_MOUSE\_DOWN: String  
ITEM\_MOUSE\_OUT: String  
ITEM\_MOUSE\_OVER: String  
ITEM\_MOUSE\_UP: String  
LegendMouseEvent()

**Linear** ■

P: mx.effects.easing  
→ Object  
easeIn() : Number  
easeInOut() : Number  
easeNone() : Number  
easeOut() : Number

**LinearAxis** ■

P: mx.charts  
→ NumericAxis  
interval: Number  
maximum: Number  
maximumLabelPrecision: Number  
minimum: Number  
minorInterval: Number  
LinearAxis()

**LinearGradient** ■

P: mx.graphics  
→ GradientBase  
IFill  
angle: Number  
begin() : void  
end() : void  
LinearGradient()

**LinearGradientStroke**

P: mx.graphics  
 → GradientBase  
*IStroke*  
 angle: Number  
 caps: String  
 interpolationMethod: String  
 joints: String  
 miterLimit: Number  
 pixelHinting: Boolean  
 scaleMode: String  
 spreadMethod: String  
 weight: Number  
 apply(): void  
 LinearGradientStroke()

**LineChart**

P: mx.charts  
 → CartesianChart

**LineFormattedTarget**

P: mx.logging.targets  
 → AbstractTarget  
 fieldSeparator: String  
 includeCategory: Boolean  
 includeDate: Boolean  
 includeLevel: Boolean  
 includeTime: Boolean  
 LineFormattedTarget()  
 logEvent(): void

**LineRenderer**

P: mx.charts.renderers  
 → ProgrammaticSkin  
*IDataRenderer*  
 data: Object  
 LineRenderer()

**LineScaleMode**

P: flash.display  
 → Object  
[HORIZONTAL: String](#)  
[NONE: String](#)  
[NORMAL: String](#)  
[VERTICAL: String](#)

**LineSeries**

P: mx.charts.series  
 → Series  
 fillFunction: Function  
 horizontalAxis: IAxis  
 interpolateValues: Boolean  
 items: Array  
 itemType: Class  
 lineSegmentType: Class  
 radius: Number  
 renderDataType: Class  
 sortOnXField: Boolean  
 verticalAxis: IAxis  
 xField: String  
 yField: String  
 applyItemRendererProperties(): void  
 commitProperties(): void  
 LineSeries()

**LineSeriesAutomationImpl**

P: mx.automation.delegates.charts  
 → SeriesAutomationImpl  
 init(): void  
 LineSeriesAutomationImpl()

**LineSeriesItem**

P: mx.charts.series.items  
 → ChartItem  
 fill: IFill  
 x: Number  
 xFilter: Number  
 xNumber: Number  
 xValue: Object  
 y: Number  
 yFilter: Number  
 yNumber: Number  
 yValue: Object  
 LineSeriesItem()

**LineSeriesRenderData**

P: mx.charts.series.renderData  
 → RenderData  
 radius: Number  
 segments: Array  
 validPoints: Number  
 LineSeriesRenderData()

**LineSeriesSegment**

P: mx.charts.series.items  
 → Object  
 element: LineSeries  
 end: uint  
 index: uint  
 items: Array  
 start: uint  
 clone(): LineSeriesSegment  
 LineSeriesSegment()

**LinkBar**

P: mx.controls  
 → NavBar  
 selectedIndex: int  
 LinkBar()  
 styleChanged(): void  
 updateDisplayList(): void

**LinkBarAutomationImpl**

P: mx.automation.delegates.controls  
 → NavBarAutomationImpl  
 init(): void  
 LinkBarAutomationImpl()

**LinkButton**

P: mx.controls  
 → Button

**LinkButtonSkin**

P: mx.skins.halo  
 → Border

**LinkSeparator**

P: mx.skins.halo  
 → ProgrammaticSkin

**List**

P: mx.controls  
 → ListBase  
*IIMESupport*  
 editable: Boolean  
 editedItemPosition: Object  
 editedItemRenderer: IListItemRenderer  
 editorDataField: String  
 editorHeightOffset: Number  
 editorUsesEnterKey: Boolean  
 editorWidthOffset: Number  
 editorXOffset: Number  
 editorYOffset: Number  
 imeMode: String

itemEditor: IFactory  
 itemEditorInstance: IListItemRenderer  
 lockedRowCount: int  
 measuringObjects: Dictionary  
 rendererIsEditor: Boolean  
 createItemEditor(): void  
 createItemRenderer(): IListItemRenderer  
 destroyItemEditor(): void  
 drawRowBackground(): void  
 endEdit(): Boolean  
 isItemEditable(): Boolean  
 layoutEditor(): void  
 List()  
 makeListData(): BaseListData

**ListAutomationImpl**

P: mx.automation.delegates.controls  
 → ListBaseAutomationImpl  
 automationTabularData: Object  
 init(): void  
 ListAutomationImpl()

**ListBase**

P: mx.controls.listClasses  
 → ScrollControlBase  
*IDataRenderer*  
*IFocusManagerComponent*  
*IListItemRenderer*  
*IDropInListItemRenderer*  
*IEffectTargetHost*  
 actualCollection: ICollectionView  
 actualIterator: IViewCursor  
 allowDragSelection: Boolean  
 allowItemSizeChangeNotification: Boolean  
 allowMultipleSelection: Boolean  
 anchorBookmark: CursorBookmark  
 anchorIndex: int  
 cachedItemsChangeEffect: IEffect  
 caretBookmark: CursorBookmark  
 caretIndex: int  
 caretIndicator: Sprite  
 caretItemRenderer: IListItemRenderer  
 caretUID: String  
 collection: ICollectionView  
 columnCount: int  
 columnWidth: Number  
 data: Object  
 dataEffectCompleted: Boolean  
 dataItemWrappersByRenderer: Dictionary  
 dataProvider: Object  
 dataTipField: String  
 dataTipFunction: Function  
 defaultColumnCount: int  
 defaultRowCount: int  
 dragEnabled: Boolean  
 dragImage: IUIComponent  
 dragImageOffsets: Point  
 dragMoveEnabled: Boolean  
 dropEnabled: Boolean  
 explicitColumnCount: int  
 explicitColumnWidth: Number  
 explicitRowCount: int  
 explicitRowHeight: Number  
 factoryMap: Dictionary  
 freeItemRenderers: Array  
 freeItemRenderersByFactory: Dictionary  
 highlightIndicator: Sprite  
 highlightItemRenderer: IListItemRenderer  
 createItemEditor(): IListItemRen-

derer  
 destroyRow(): void  
 dragCompleteHandler(): void  
 dragDropHandler(): void  
 dragEnterHandler(): void  
 dragExitHandler(): void  
 dragOverHandler(): void  
 dragScroll(): void  
 dragStartHandler(): void  
 drawCaretIndicator(): void  
 drawHighlightIndicator(): void  
 drawItem(): void  
 drawRowBackgrounds(): void  
 drawSelectionIndicator(): void  
 findKey(): Boolean  
 findString(): Boolean  
 finishDataChangeEffect(): void  
 finishKeySelection(): void  
 getItemRendererFactory(): IFactory  
 getRendererSemanticValue(): Object  
 getReservedOrFreeItemRenderer(): IListItemRenderer  
 hideDropFeedback(): void  
 indexToColumn(): int  
 indexToItemRenderer(): IListItemRenderer  
 indexToRow(): int  
 indicesToIndex(): int  
 initiateDataChangeEffect(): void  
 invalidateList(): void  
 isItemHighlighted(): Boolean  
 isItemSelectable(): Boolean  
 isItemSelected(): Boolean  
 isItemVisible(): Boolean  
 isRendererUnconstrained(): Boolean  
 itemRendererContains(): Boolean  
 itemRendererToIndex(): int  
 itemRendererToIndices(): Point  
 itemToDataTip(): String  
 itemToIcon(): Class  
 itemToItemRenderer(): IListItemRenderer  
 itemToLabel(): String  
 itemToUID(): String  
 ListBase()  
 makeRowsAndColumns(): Point  
 measure(): void  
 measureHeightOfItems(): Number  
 measureWidthOfItems(): Number  
 mouseClickHandler(): void  
 mouseDoubleClickHandler(): void  
 mouseDownHandler(): void  
 mouseEventToItemRenderer(): IListItemRenderer  
 mouseMoveHandler(): void  
 mouseOutHandler(): void  
 mouseOverHandler(): void  
 mouseUpHandler(): void  
 mouseWheelHandler(): void  
 moveIndicatorsHorizontally(): void  
 moveIndicatorsVertically(): void  
 moveRowVertically(): void  
 moveSelectionHorizontally(): void  
 moveSelectionVertically(): void  
 prepareDataEffect(): void  
 purgeItemRenderers(): void  
 reKeyVisibleData(): void  
 removeDataEffectItem(): void  
 removeFromRowArrays(): void  
 removeIndicators(): void  
 restoreRowArrays(): void  
 scrollHorizontally(): void  
 scrollPositionToIndex(): int  
 scrollToIndex(): Boolean

scrollVertically(): void  
 seekPendingFailureHandler(): void  
 seekPendingResultHandler(): void  
 seekPositionSafely(): Boolean  
 selectItem(): Boolean  
 setRowCount(): void  
 setRowHeight(): void  
 shiftRow(): void  
 showDropFeedback(): void  
 sumRowHeights(): Number  
 truncateRowArrays(): void  
 UIDToItemRenderer(): IListItemRenderer  
 unconstrainRenderer(): void  
 updateDisplayList(): void  
 updateList(): void

**ListBaseAutomationImpl**

P: mx.automation.delegates.controls  
 → ScrollControlBaseAutomationImpl  
 automationTabularData: Object  
 init(): void  
 ListBaseAutomationImpl()  
 updateItemRenderers(): void

**ListBaseContentHolder**

P: mx.controls.listClasses  
 → UIComponent  
 bottomOffset: Number  
 heightExcludingOffsets: Number  
 iterator: IViewCursor  
 leftOffset: Number  
 listItems: Array  
 rightOffset: Number  
 rowInfo: Array  
 selectionLayer: Sprite  
 topOffset: Number  
 visibleData: Object  
 widthExcludingOffsets: Number  
 ListBaseContentHolder()  
 updateDisplayList(): void

**ListBaseContentHolderAutomationImpl**

P: mx.automation.delegates.controls  
 → UIComponentAutomationImpl  
 init(): void  
 ListBaseContentHolderAutomationImpl()

**ListBaseSeekPending**

P: mx.controls.listClasses  
 → Object  
 bookmark: CursorBookmark  
 offset: int  
 ListBaseSeekPending()

**ListBaseSelectionData**

P: mx.controls.listClasses  
 → Object  
 approximate: Boolean  
 data: Object  
 index: int  
 ListBaseSelectionData()

**ListCollectionView**

P: mx.collections  
 → Proxy  
*ICollectionView*  
*IList*  
*IMXMLObject*

filterFunction: Function  
 length: int  
 list: IList  
 localIndex: Array  
 sort: Sort  
 addEventListener(): void  
 addItem(): void  
 addItemAt(): void  
 contains(): Boolean  
 createCursor(): IViewCursor  
 disableAutoUpdate(): void  
 dispatchEvent(): Boolean  
 enableAutoUpdate(): void  
 getItemAt(): Object  
 getItemIndex(): int  
 hasEventListener(): Boolean  
 initialized(): void  
 itemUpdated(): void  
 ListCollectionView()  
 refresh(): Boolean  
 removeAll(): void  
 removeEventListener(): void  
 removeItemAt(): Object  
 setItemAt(): Object  
 toArray(): Array  
 toString(): String  
 willTrigger(): Boolean

**ListData**

P: mx.controls.listClasses  
 → BaseListData  
 icon: Class  
 labelField: String  
 ListData()

**ListDropIndicator**

P: mx.skins.halo  
 → ProgrammaticSkin  
 direction: String  
 ListDropIndicator()

**ListEvent**

P: mx.events  
 → Event  
[CHANGE: String](#)  
 columnIndex: int  
[ITEM\\_CLICK: String](#)  
[ITEM\\_DOUBLE\\_CLICK: String](#)  
[ITEM\\_EDIT\\_BEGIN: String](#)  
[ITEM\\_EDIT\\_BEGINNING: String](#)  
[ITEM\\_EDIT\\_END: String](#)  
[ITEM\\_FOCUS\\_IN: String](#)  
[ITEM\\_FOCUS\\_OUT: String](#)  
[ITEM\\_ROLL\\_OUT: String](#)  
[ITEM\\_ROLL\\_OVER: String](#)  
 itemRenderer: IListItemRenderer  
 reason: String  
 rowIndex: int  
 ListEvent()

**ListEventReason**

P: mx.events  
 → Object  
[CANCELLED: String](#)  
[NEW\\_ROW: String](#)  
[OTHER: String](#)

**ListItemDragProxy**

P: mx.controls.listClasses  
 → UIComponent  
 ListItemDragProxy()  
 measure(): void

**ListItemRenderer**

P: mx.controls.listClasses  
 → UIComponent  
*IDataRenderer*  
*IDropInListItemRenderer*  
*IListItemRenderer*  
*IFontContextComponent*  
 data: Object  
 fontContext: IFlexModuleFactory  
 icon: IFlexDisplayObject  
 label: IUITextField  
 listData: BaseListData  
 ListItemRenderer()  
 toolTipShowHandler(): void

**ListItemRendererAutomationImpl**

P: mx.automation.delegates.controls  
 → UIComponentAutomationImpl  
 init(): void  
 ListItemRendererAutomationImpl()

**ListItemSelectEvent**

P: mx.automation.events  
 → Event  
 altKey: Boolean  
 ctrlKey: Boolean  
[DESELECT: String = "deselect"](#)  
 itemAutomationValue: String  
 itemIndex: uint  
 itemRenderer: IListItemRenderer  
[MULTI\\_SELECT: String = "multiSelect"](#)  
 shiftKey: Boolean  
 triggerEvent: Event  
[SELECT: String = "select"](#)  
[SELECT\\_INDEX: String = "selectIndex"](#)  
 ListItemSelectEvent(): void

**ListRowInfo**

P: mx.controls.listClasses  
 → Object  
 data: Object  
 height: Number  
 itemOldY: Number  
 oldY: Number  
 uid: String  
 y: Number  
 ListRowInfo()

**Loader**

P: flash.display  
 → DisplayObjectContainer  
 content: DisplayObject  
 contentLoaderInfo: LoaderInfo  
 close(): void  
 load(): void  
 loadBytes(): void  
 Loader()  
 unload(): void

**LoaderConfig**

P: mx.messaging.config  
 → Object  
 parameters: Object  
 url: String  
 LoaderConfig()

**LoaderContext**

P: flash.system  
 → Object  
 applicationDomain: ApplicationDomain  
 checkPolicyFile: Boolean  
 securityDomain: SecurityDomain  
 LoaderContext()



**LoaderInfo**

P: flash.display  
→ [EventDispatcher](#)  
[actionScriptVersion](#): uint  
[applicationDomain](#): [ApplicationDomain](#)  
[bytes](#): [ByteArray](#)  
[bytesLoaded](#): uint  
[bytesTotal](#): uint  
[childAllowsParent](#): Boolean  
[childSandboxBridge](#): Object  
[content](#): [DisplayObject](#)  
[contentType](#): String  
[frameRate](#): Number  
[height](#): int  
[loader](#): [Loader](#)  
[loaderURL](#): String  
[parameters](#): Object  
[parentAllowsChild](#): Boolean  
[parentSandboxBridge](#): Object  
[sameDomain](#): Boolean  
[sharedEvents](#): [EventDispatcher](#)  
[swfVersion](#): uint  
[url](#): String  
[width](#): int  
[getLoaderInfoByDefinition\(\)](#) : [LoaderInfo](#)

**LoaderUtil**

P: mx.utils  
→ Object  
[normalizeURL\(\)](#) : String

**LoadEvent**

P: mx.rpc.soap  
[LOAD](#): String  
clone(): Event  
LoadEvent()  
toString(): String

**LocalConnection**

P: flash.net  
→ [EventDispatcher](#)  
[client](#): Object  
[domain](#): String  
[allowDomain\(\)](#) : void  
[allowInsecureDomain\(\)](#) : void  
[close\(\)](#) : void  
[connect\(\)](#) : void  
[LocalConnection\(\)](#)  
[send\(\)](#) : void

**Locale**

P: mx.resources  
→ Object  
[country](#): String  
[language](#): String  
[variant](#): String  
[Locale\(\)](#)  
[toString\(\)](#) : String

**Log**

P: mx.logging  
→ Object  
[addTarget\(\)](#) : void  
[flush\(\)](#) : void  
[getLogger\(\)](#) : [ILogger](#)  
[hasIllegalCharacters\(\)](#) : Boolean  
[isDebug\(\)](#) : Boolean  
[isError\(\)](#) : Boolean  
[isFatal\(\)](#) : Boolean  
[isInfo\(\)](#) : Boolean  
[isWarn\(\)](#) : Boolean  
[removeTarget\(\)](#) : void

**LogAxis**

P: mx.charts  
→ [NumericAxis](#)  
[interval](#): Number  
[maximum](#): Number  
[maximumLabelPrecision](#): Number  
[minimum](#): Number  
[LogAxis\(\)](#)

**LogEvent**

P: mx.logging  
→ Event  
[level](#): int  
[LOG](#): String  
[message](#): String  
[getLevelString\(\)](#) : String  
[LogEvent\(\)](#)

**LogEventLevel**

P: mx.logging  
→ Object  
[ALL](#): int  
[DEBUG](#): int  
[ERROR](#): int  
[FATAL](#): int  
[INFO](#): int  
[WARN](#): int

**LogLogger**

P: mx.logging  
→ [EventDispatcher](#)  
[ILogger](#)  
[category](#): String  
[debug\(\)](#) : void  
[error\(\)](#) : void  
[fatal\(\)](#) : void  
[info\(\)](#) : void  
[log\(\)](#) : void  
[LogLogger\(\)](#)  
[warn\(\)](#) : void

**MaskEffect**

P: mx.effects  
→ Effect  
[createMaskFunction](#): Function  
[moveEasingFunction](#): Function  
[scaleEasingFunction](#): Function  
[scaleXFrom](#): Number  
[scaleXTo](#): Number  
[scaleYFrom](#): Number  
[scaleYTo](#): Number  
[showTarget](#): Boolean  
[xFrom](#): Number  
[xTo](#): Number  
[yFrom](#): Number  
[yTo](#): Number  
[getAffectedProperties\(\)](#) : Array  
[MaskEffect\(\)](#)  
[tweenEventHandler\(\)](#) : void

**MaskEffectInstance**

P: mx.effects.effectClasses  
→ [EffectInstance](#)  
[createMaskFunction](#): Function  
[effectMask](#): Shape  
[moveEasingFunction](#): Function  
[scaleEasingFunction](#): Function  
[scaleXFrom](#): Number  
[scaleXTo](#): Number  
[scaleYFrom](#): Number  
[scaleYTo](#): Number  
[showTarget](#): Boolean  
[targetArea](#): [Rectangle](#)  
[targetVisualBounds](#): [Rectangle](#)

[xFrom](#): Number  
[xTo](#): Number  
[yFrom](#): Number  
[yTo](#): Number  
[defaultCreateMask\(\)](#) : Shape  
[initMaskEffect\(\)](#) : void  
[MaskEffectInstance\(\)](#)  
[onMoveTweenEnd\(\)](#) : void  
[onMoveTweenUpdate\(\)](#) : void  
[onScaleTweenEnd\(\)](#) : void  
[onScaleTweenUpdate\(\)](#) : void  
[pause\(\)](#) : void  
[resume\(\)](#) : void  
[reverse\(\)](#) : void

**Math**

P: Top Level  
→ Object  
[E](#): Number  
[LN10](#): Number  
[LN2](#): Number  
[LOG10E](#): Number  
[LOG2E](#): Number  
[PI](#): Number  
[SQRT1\\_2](#): Number  
[SQRT2](#): Number  
[abs\(\)](#) : Number  
[acos\(\)](#) : Number  
[asin\(\)](#) : Number  
[atan\(\)](#) : Number  
[atan2\(\)](#) : Number  
[ceil\(\)](#) : Number  
[cos\(\)](#) : Number  
[exp\(\)](#) : Number  
[floor\(\)](#) : Number  
[log\(\)](#) : Number  
[max\(\)](#) : Number  
[min\(\)](#) : Number  
[pow\(\)](#) : Number  
[random\(\)](#) : Number  
[round\(\)](#) : Number  
[sin\(\)](#) : Number  
[sqrt\(\)](#) : Number  
[tan\(\)](#) : Number

**Matrix**

P: flash.geom  
→ Object  
[a](#): Number  
[b](#): Number  
[c](#): Number  
[d](#): Number  
[tx](#): Number  
[ty](#): Number  
[clone\(\)](#) : [Matrix](#)  
[concat\(\)](#) : void  
[createBox\(\)](#) : void  
[createGradientBox\(\)](#) : void  
[deltaTransformPoint\(\)](#) : [Point](#)  
[identity\(\)](#) : void  
[invert\(\)](#) : void  
[Matrix\(\)](#)  
[rotate\(\)](#) : void  
[scale\(\)](#) : void  
[toString\(\)](#) : String  
[transformPoint\(\)](#) : [Point](#)  
[translate\(\)](#) : void

**MaxAggregator**

P: mx.olap.aggregators  
→ Object  
[IOLAPCustomAggregator](#)  
[computeBegin\(\)](#) : Object  
[computeEnd\(\)](#) : Number

[computeLoop\(\)](#) : void  
[computeObjectBegin\(\)](#) : Object  
[computeObjectEnd\(\)](#) : Number  
[computeObjectLoop\(\)](#) : void

**MemoryError**

P: flash.errors  
→ Error  
[MemoryError\(\)](#) : void

**Menu**

P: mx.controls  
→ List  
[IFocusManagerContainer](#)  
[dataDescriptor](#): [IMenuDataDescriptor](#)  
[hasRoot](#): Boolean  
[parentMenu](#): [Menu](#)  
[showRoot](#): Boolean  
[createMenu\(\)](#) : [Menu](#)  
[hide\(\)](#) : void  
[makeListData\(\)](#) : [BaseListData](#)  
[measure\(\)](#) : void  
[Menu\(\)](#)  
[popUpMenu\(\)](#) : void  
[setMenuItemToggled\(\)](#) : void  
[show\(\)](#) : void

**MenuAutomationImpl**

P: mx.automation.delegates.  
controls  
→ [ListAutomationImpl](#)  
[showHideFromKeys](#): Boolean  
[init\(\)](#) : void  
[MenuAutomationImpl\(\)](#) : void

**MenuBar**

P: mx.controls  
→ [UIComponent](#)  
[IFocusManagerComponent](#)  
[dataDescriptor](#): [IMenuDataDescriptor](#)  
[dataProvider](#): Object  
[hasRoot](#): Boolean  
[iconField](#): String  
[labelField](#): String  
[labelFunction](#): Function  
[menuBarItemRenderer](#): [IFactory](#)  
[menuBarItems](#): Array  
[menuItemStyleFilters](#): Object  
[menus](#): Array  
[selectedIndex](#): int  
[showRoot](#): Boolean  
[getMenuAt\(\)](#) : [Menu](#)  
[itemToIcon\(\)](#) : [Class](#)  
[itemToLabel\(\)](#) : String  
[measure\(\)](#) : void  
[MenuBar\(\)](#)  
[updateBackground\(\)](#) : void

**MenuBarAutomationImpl**

P: mx.automation.delegates.  
controls  
→ [UIComponentAutomationImpl](#)  
[init\(\)](#) : void  
[MenuBarAutomationImpl\(\)](#)

**MenuBarBackgroundSkin**

P: mx.skins.halo  
→ Border

**MenuBarItem**

P: mx.controls.menuClasses  
→ [UIComponent](#)  
[IMenuBarItemRenderer](#)  
[IFontContextComponent](#)

[data](#): Object  
[dataProvider](#): Object  
[icon](#): [IFlexDisplayObject](#)  
[label](#): [UITextField](#)  
[menuBar](#): [MenuBar](#)  
[menuItemIndex](#): int  
[menuItemState](#): String  
[MenuBarItem\(\)](#)

**MenuBarItemAutomationImpl**

P: mx.automation.delegates.  
controls  
→ [UIComponentAutomationImpl](#)  
[init\(\)](#) : void  
[MenuBarItemAutomationImpl\(\)](#)

**MenuEvent**

P: mx.events  
→ [ListEvent](#)  
[CHANGE](#): String  
[index](#): int  
[item](#): Object  
[ITEM\\_CLICK](#): String  
[ITEM\\_ROLL\\_OUT](#): String  
[ITEM\\_ROLL\\_OVER](#): String  
[label](#): String  
[menu](#): [Menu](#)  
[MENU\\_HIDE](#): String  
[MENU\\_SHOW](#): String  
[menuBar](#): [MenuBar](#)  
[MenuEvent\(\)](#)

**MenuItemRenderer**

P: mx.controls.menuClasses  
→ [UIComponent](#)  
[IDataRenderer](#)  
[IListItemRenderer](#)  
[IMenuItemRenderer](#)  
[IDropInListItemRenderer](#)  
[IFontContextComponent](#)  
[branchIcon](#): [IFlexDisplayObject](#)  
[data](#): Object  
[icon](#): [IFlexDisplayObject](#)  
[label](#): [UITextField](#)  
[listData](#): [BaseListData](#)  
[measuredBranchIconWidth](#): Number  
[measuredIconWidth](#): Number  
[measuredTypeIconWidth](#): Number  
[menu](#): [Menu](#)  
[separatorIcon](#): [IFlexDisplayObject](#)  
[typeIcon](#): [IFlexDisplayObject](#)  
[MenuItemRenderer\(\)](#)

**MenuItemRendererAutomationImpl**

P: mx.automation.delegates.  
controls  
→ [UIComponentAutomationImpl](#)  
[init\(\)](#) : void  
[ContainerAutomationImpl](#)  
[MenuItemRendererAutomationImpl\(\)](#)

**MenuItemListData**

P: mx.controls.menuClasses  
→ [ListData](#)  
[maxMeasuredBranchIconWidth](#): Number  
[maxMeasuredIconWidth](#): Number  
[maxMeasuredTypeIconWidth](#): Number  
[useTwoColumns](#): Boolean  
[MenuItemListData\(\)](#)

**MenuShowEvent**

P: mx.automation.events  
→ Event  
[MENU\\_SHOW](#) : String = "menuShow"  
[MenuShowEvent\(\)](#):void

**MessageAckEvent**

P: mx.messaging.events  
→ [MessageEvent](#)  
[ACKNOWLEDGE](#): String  
[acknowledgeMessage](#): [AcknowledgeMessage](#)  
[correlation](#): [IMessage](#)  
[clone\(\)](#) : Event  
[createEvent\(\)](#) : [MessageAckEvent](#)  
[MessageAckEvent\(\)](#)  
[toString\(\)](#) : String

**MessageAgent**

P: mx.messaging  
→ [EventDispatcher](#)  
[IMXMLObject](#)  
[\\_agentType](#): String  
[\\_credentials](#): String  
[\\_credentialsCharset](#): String  
[\\_disconnectBarrier](#): Boolean  
[\\_ignoreFault](#): Boolean  
[\\_log](#): [ILogger](#)  
[channelSet](#): [ChannelSet](#)  
[clientId](#): String  
[connected](#): Boolean  
[destination](#): String  
[requestTimeout](#): int  
[acknowledge\(\)](#) : void  
[assertCredentials\(\)](#) : void  
[channelConnectHandler\(\)](#) : void  
[channelDisconnectHandler\(\)](#) : void  
[channelFaultHandler\(\)](#) : void  
[disconnect\(\)](#) : void  
[fault\(\)](#) : void  
[initChannelSet\(\)](#) : void  
[initialized\(\)](#) : void  
[internalSend\(\)](#) : void  
[logout\(\)](#) : void  
[MessageAgent\(\)](#)  
[reAuthorize\(\)](#) : void  
[setCredentials\(\)](#) : void  
[setRemoteCredentials\(\)](#) : void

**MessageEvent**

P: mx.messaging.events  
→ Event  
[MESSAGE](#): String  
[message](#): [IMessage](#)  
[RESULT](#): String  
[clone\(\)](#) : Event  
[createEvent\(\)](#) : [MessageEvent](#)  
[MessageEvent\(\)](#)  
[toString\(\)](#) : String

**MessageFaultEvent**

P: mx.messaging.events  
→ Event  
[FAULT](#): String  
[faultCode](#): String  
[faultDetail](#): String  
[faultString](#): String  
[message](#): [ErrorMessage](#)  
[rootCause](#): Object  
[clone\(\)](#) : Event  
[createEvent\(\)](#) : [MessageFaultEvent](#)  
[MessageFaultEvent\(\)](#)  
[toString\(\)](#) : String

**MessagePerformanceUtils**

P: mx.messaging.messages  
→ Object  
[clientReceiveTime](#): Number  
[messageSize](#): int  
[networkRTT](#): Number  
[originatingMessageSentTime](#): Number  
[originatingMessageSize](#): Number  
[pushedMessageFlag](#): Boolean  
[pushOneWayTime](#): Number  
[responseMessageSize](#): int  
[serverAdapterExternalTime](#): Number  
[serverAdapterTime](#): Number  
[serverNonAdapterTime](#): Number  
[serverPollDelay](#): Number  
[serverPrePushTime](#): Number  
[serverProcessingTime](#): Number  
[serverSendTime](#): Number  
[totalPushTime](#): Number  
[totalTime](#): Number  
[MessagePerformanceUtils\(\)](#)  
[prettyPrint\(\)](#) : String

**MessageResponder**

P: mx.messaging  
→ [Responder](#)  
[agent](#): [MessageAgent](#)  
[channel](#): [Channel](#)  
[message](#): [IMessage](#)  
[createRequestTimeoutErrorMessage\(\)](#) : [ErrorMessage](#)  
[MessageResponder\(\)](#)  
[requestTimedOut\(\)](#) : void  
[result\(\)](#) : void  
[resultHandler\(\)](#) : void  
[status\(\)](#) : void  
[statusHandler\(\)](#) : void

**MessageSerializationError**

P: mx.messaging.errors  
→ [MessagingError](#)  
[fault](#): [ErrorMessage](#)  
[MessageSerializationError\(\)](#)

**MessagingError**

P: mx.messaging.errors  
→ Error  
[MessagingError\(\)](#)  
[toString\(\)](#) : String

**MetadataEvent**

P: mx.events  
→ Event  
[ACTION\\_SCRIPT](#): String  
[CUE\\_POINT](#): String  
[info](#): Object  
[METADATA\\_RECEIVED](#): String  
[MetadataEvent\(\)](#)

**Microphone**

P: flash.media  
→ [EventDispatcher](#)  
[activityLevel](#): Number  
[gain](#): Number  
[index](#): int  
[muted](#): Boolean  
[name](#): String  
[names](#): Array  
[rate](#): int  
[silenceLevel](#): Number  
[silenceTimeout](#): int  
[soundTransform](#): [SoundTransform](#)  
[useEchoSuppression](#): Boolean  
[getMicrophone\(\)](#) : [Microphone](#)

[setLoopBack\(\)](#) : void  
[setSilenceLevel\(\)](#) : void  
[setUseEchoSuppression\(\)](#) : void

**MinAggregator**

P: mx.olap.aggregators  
→ Object  
[IOLAPCustomAggregator](#)  
[computeBegin\(\)](#) : Object  
[computeEnd\(\)](#) : Number  
[computeLoop\(\)](#) : void  
[computeObjectBegin\(\)](#) : Object  
[computeObjectEnd\(\)](#) : Number  
[computeObjectLoop\(\)](#) : void

**MiniDebugTarget**

P: mx.logging.targets  
→ [LineFormattedTarget](#)

**Module**

P: mx.modules  
→ [LayoutContainer](#)

**ModuleBase**

P: mx.modules  
→ [EventDispatcher](#)

**ModuleEvent**

P: mx.events  
→ [ProgressEvent](#)  
[ERROR](#): String  
[errorText](#): String  
[module](#): [IModuleInfo](#)  
[PROGRESS](#): String  
[READY](#): String  
[SETUP](#): String  
[UNLOAD](#): String  
[ModuleEvent\(\)](#)

**ModuleLoader**

P: mx.modules  
→ [VBox](#)  
[IDeferredInstantiationUIComponent](#)  
[applicationDomain](#): [ApplicationDomain](#)  
[child](#): [DisplayObject](#)  
[url](#): String  
[loadModule\(\)](#) : void  
[ModuleLoader\(\)](#)  
[unloadModule\(\)](#) : void

**ModuleManager**

P: mx.modules  
→ Object  
[getAssociatedFactory\(\)](#) : [IFlexModuleFactory](#)  
[getModule\(\)](#) : [IModuleInfo](#)

**MorphShape**

P: flash.display  
→ [Display Object](#)

**Mouse**

P: flash.ui  
→ Object  
[hide\(\)](#) : void  
[show\(\)](#) : void

**MouseEvent**

P: flash.events  
→ Event  
[altKey](#): Boolean  
[buttonDown](#): Boolean  
[CLICK](#): String  
[clickCount](#): int

commandKey: Boolean  
 CONTEXT\_MENU: String  
 controlKey: Boolean  
 ctrlKey: Boolean  
 delta: int  
 DOUBLE\_CLICK: String  
 localX: Number  
 localY: Number  
 MIDDLE\_CLICK: String  
 MIDDLE\_MOUSE\_DOWN: String  
 MIDDLE\_MOUSE\_UP: String  
 MOUSE\_DOWN: String  
 MOUSE\_MOVE: String  
 MOUSE\_OUT: String  
 MOUSE\_OVER: String  
 MOUSE\_UP: String  
 MOUSE\_WHEEL: String  
 relatedObject: InteractiveObject  
 RIGHT\_CLICK: String  
 RIGHT\_MOUSE\_DOWN: String  
 RIGHT\_MOUSE\_UP: String  
 ROLL\_OUT: String  
 ROLL\_OVER: String  
 shiftKey: Boolean  
 stageX: Number  
 stageY: Number  
 clone(): Event  
 MouseEvent()  
 toString(): String  
 updateAfterEvent(): void

#### Move ■

P: mx.effects  
 → TweenEffect  
 xBy: Number  
 xFrom: Number  
 xTo: Number  
 yBy: Number  
 yFrom: Number  
 yTo: Number  
 Move(): void

#### MoveEvent ■

P: mx.events  
 → Event  
 MOVE: String  
 oldX: Number  
 oldY: Number  
 MoveEvent(): void

#### MoveInstance ■

P: mx.effects.effectClasses  
 → TweenEffectInstance  
 xBy: Number  
 xFrom: Number  
 xTo: Number  
 yBy: Number  
 yFrom: Number  
 yTo: Number  
 MoveInstance()

#### MovieClip

P: flash.display  
 → Sprite  
 currentFrame: int  
 currentLabel: String  
 currentLabels: Array  
 currentScene: Scene  
 enabled: Boolean  
 framesLoaded: int  
 scenes: Array  
 totalFrames: int  
 trackAsMenu: Boolean  
 gotoAndPlay(): void

gotoAndStop(): void  
 MovieClip()  
 nextFrame(): void  
 nextScene(): void  
 play(): void  
 prevFrame(): void  
 prevScene(): void  
 stop(): void

#### MovieClipAsset ■

P: mx.core  
 → FlexMovieClip  
 IFlexAsset  
 IFlexDisplayObject  
 IBorder  
 borderMetrics: EdgeMetrics  
 measuredHeight: Number  
 measuredWidth: Number  
 move(): void  
 MovieClipAsset()  
 setActualSize(): void

#### MovieClipLoaderAsset ■

P: mx.core  
 → MovieClipAsset  
 IFlexAsset  
 IFlexDisplayObject  
 initialHeight: Number  
 initialWidth: Number  
 movieClipData: ByteArray  
 MovieClipLoaderAsset()

#### MultiTopicConsumer ■

P: mx.messaging  
 → AbstractConsumer  
 subscriptions: ArrayCollection  
 MultiTopicConsumer():void  
 addSubscription():void  
 removeSubscription():void

#### MultiTopicProducer ■

P: mx.message  
 → AbstractProducer  
 subtopics: ArrayCollection  
 MultiTopicProducer()  
 addSubtopic():void  
 removeSubtopic():void

#### Namespace

P: Top Level  
 → Object  
 prefix: String  
 uri: String  
 Namespace(): void  
 Namespace(): void  
 toString(): String  
 valueOf(): String

#### NameUtil ■

P: mx.utils  
 → Object  
 createUniqueName(): String  
 displayObjectToString(): String

#### NativeApplication ■

P: flash.desktop  
 → EventDispatcher  
 activeWindow: NativeWindow  
 applicationDescriptor: XML  
 applicationID: String  
 autoExit: Boolean  
 icon: InteractiveIcon  
 idleThreshold: int  
 menu: NativeMenu

#### nativeApplication: NativeApplication

openedWindows: Array  
 publisherID: String  
 runtimePatchLevel: uint  
 runtimeVersion: String  
 startAtLogin: Boolean  
 supportsDockIcon: Boolean  
 supportsMenu: Boolean  
 supportsSystemTrayIcon: Boolean  
 timeSinceLastUserInput: int  
 activate(): void  
 addEventListener(): void  
 clear(): Boolean  
 copy(): Boolean  
 cut(): Boolean  
 dispatchEvent(): Boolean  
 exit(): void  
 getDefaultApplication(): String  
 move(): void  
 MovieClipAsset()  
 setActualSize(): void  
 undo(): Boolean

#### NativeDragActions ■

P: flash.desktop  
 → Object  
 COPY: String  
 LINK: String  
 MOVE: String  
 NONE: String

#### NativeDragEvent ■

P: flash.events  
 → MouseEvent  
 allowedActions: NativeDragOptions  
 clipboard: Clipboard  
 dropAction: String  
 NATIVE\_DRAG\_COMPLETE: String  
 NATIVE\_DRAG\_DROP: String  
 NATIVE\_DRAG\_ENTER: String  
 NATIVE\_DRAG\_EXIT: String  
 NATIVE\_DRAG\_OVER: String  
 NATIVE\_DRAG\_START: String  
 NATIVE\_DRAG\_UPDATE: String  
 clone(): Event  
 NativeDragEvent()  
 toString(): String

#### NativeDragManager ■

P: flash.desktop  
 → Object  
 dragInitiator: InteractiveObject  
 dropAction: String  
 isDragging: Boolean  
 acceptDragDrop(): void  
 doDrag(): void

#### NativeDragOptions ■

P: flash.desktop  
 → Object  
 allowCopy: Boolean  
 allowLink: Boolean  
 allowMove: Boolean  
 toString(): String

#### NativeMenu ■

P: flash.display  
 → EventDispatcher  
 items: Array  
 numItems: int

parent: NativeMenu  
 addItem(): NativeMenuItem  
 addItemAt(): NativeMenuItem  
 addSubMenu(): NativeMenuItem  
 addSubMenuAt(): NativeMenuItem  
 clone(): NativeMenu  
 containsItem(): Boolean  
 display(): void  
 getItemAt(): NativeMenuItem  
 getItemByName(): NativeMenuItem  
 getItemIndex(): int  
 removeItem(): NativeMenuItem  
 removeItemAt(): NativeMenuItem  
 setItemIndex(): void

#### NativeMenuItem ■

P: flash.display  
 → EventDispatcher  
 checked: Boolean  
 data: Object  
 enabled: Boolean  
 isSeparator: Boolean  
 keyEquivalent: String  
 keyEquivalentModifiers: Array  
 name: String  
 submenu: NativeMenu  
 clone(): NativeMenuItem  
 mnemonicIndex: int  
 name: String  
 submenu: NativeMenu  
 clone(): NativeMenuItem  
 NativeMenuItem()  
 toString(): String

#### NativeWindow ■

P: flash.display  
 → EventDispatcher  
 active: Boolean  
 alwaysInFront: Boolean  
 bounds: Rectangle  
 clipboard: Clipboard  
 displayState: String  
 height: Number  
 maximizable: Boolean  
 maxSize: Point  
 menu: NativeMenu  
 minimizable: Boolean  
 minSize: Point  
 resizable: Boolean  
 stage: Stage  
 supportsMenu: Boolean  
 supportsNotification: Boolean  
 systemChrome: String  
 systemMaxSize: Point  
 systemMinSize: Point  
 resizable: Boolean  
 clone(): Event  
 NativeWindowEvent()  
 toString(): String

startResize(): Boolean

#### NativeWindowBoundsEvent ■

P: flash.events  
 → Events  
 afterBounds: Rectangle  
 beforeBounds: Rectangle  
 MOVE: String  
 MOVING: String  
 RESIZE: String  
 RESIZING: String  
 clone(): Event  
 NativeWindowBoundsEvent()  
 toString(): String

#### NativeWindowDisplayState ■

P: flash.display  
 → Object  
 MAXIMIZED: String  
 MINIMIZED: String  
 NORMAL: String

#### NativeWindowDisplayStateEvent ■

P: flash.events  
 → Event  
 afterDisplayState: String  
 beforeDisplayState: String  
 DISPLAY\_STATE\_CHANGE: String  
 DISPLAY\_STATE\_CHANGING: String  
 clone(): Event  
 NativeWindowDisplayStateEvent()  
 toString(): String

#### NativeWindowInitOptions ■

P: flash.display  
 → Object  
 hasMenu: Boolean  
 maximizable: Boolean  
 minimizable: Boolean  
 resizable: Boolean  
 systemChrome: String  
 transparent: Boolean  
 type: String  
 NativeWindowInitOptions()

#### NativeWindowResize ■

P: flash.display  
 → Object  
 BOTTOM: String  
 BOTTOM\_LEFT: String  
 BOTTOM\_RIGHT: String  
 LEFT: String  
 NONE: String  
 RIGHT: String  
 TOP: String  
 TOP\_LEFT: String  
 TOP\_RIGHT: String

#### NativeWindowSystemChrome ■

P: flash.display  
 → Object  
 NONE: String  
 STANDARD: String  
 UTILITY: String

#### NativeWindowType ■

P: flash.display  
 → Object  
 LIGHTWEIGHT: String  
 NORMAL: String  
 UTILITY: String

#### NavBar ■

P: mx.controls  
 → Box  
 dataProvider: Object  
 iconField: String  
 labelField: String  
 labelFunction: Function  
 selectedIndex: int  
 toolTipField: String  
 clickHandler(): void  
 createNavItem(): IFlexDisplayObject  
 hiliteSelectedNavItem(): void  
 itemToLabel(): String  
 NavBar()  
 resetNavItems(): void  
 updateNavItemIcon(): void  
 updateNavItemLabel(): void

#### NavBarAutomationImpl ■

P: mx.automation.delegates.  
 controls  
 → ContainerAutomationImpl  
 init(): void  
 NavBarAutomationImpl()

#### NetConnection

P: flash.net  
 → EventDispatcher  
 client: Object  
 connected: Boolean  
 connectedProxyType: String  
 defaultObjectEncoding: uint  
 objectEncoding: uint  
 proxyType: String  
 uri: String  
 usingTLS: Boolean  
 addHeader(): void  
 call(): void  
 close(): void  
 connect(): void  
 NetConnection()

#### NetConnectionChannel ■

P: mx.messaging.channels  
 → PollingChannel  
 netConnection: NetConnection  
 NetConnectionChannel()

#### NetStatusEvent

P: flash.events  
 → Event  
 info: Object  
 NET\_STATUS: String  
 clone(): Event  
 NetStatusEvent()  
 toString(): String

#### NetStream

P: flash.net  
 → EventDispatcher  
 bufferLength: Number  
 bufferTime: Number  
 bytesLoaded: uint  
 bytesTotal: uint  
 checkPolicyFile: Boolean  
 client: Object  
 currentFPS: Number  
 liveDelay: Number  
 objectEncoding: uint  
 soundTransform: SoundTransform  
 time: Number  
 attachAudio(): void  
 attachCamera(): void  
 close(): void  
 NetStream()

pause(): void  
 play(): void  
 publish(): void  
 receiveAudio(): void  
 receiveVideo(): void  
 receiveVideoFPS(): void  
 resetDRMVouchers(): void  
 resume(): void  
 seek(): void  
 send(): void  
 setDRMAuthenticationCredentials(): void  
 togglePause(): void

#### NewObjectSample

P: flash.sampler  
 → Sample  
 id: Number  
 object: \*  
 type: Class

#### NoChannelAvailableError ■

P: mx.messaging.errors  
 → MessagingError

#### NotificationType ■

P: flash.desktop  
 → Object  
 CRITICAL: String  
 INFORMATIONAL: String

#### Number

P: Top Level  
 → Object  
 MAX\_VALUE: Number  
 MIN\_VALUE: Number  
 NaN: Number  
 NEGATIVE\_INFINITY: Number  
 POSITIVE\_INFINITY: Number  
 Number()  
 toExponential(): String  
 toFixed(): String  
 toPrecision(): String  
 toString(): String  
 valueOf(): Number

#### NumberBase ■

P: mx.flash  
 → Object  
 decimalSeparatorFrom: String  
 decimalSeparatorTo: String  
 isValid: Boolean  
 thousandsSeparatorFrom: String  
 thousandsSeparatorTo: String  
 formatDecimal(): String  
 formatNegative(): String  
 formatPrecision(): String  
 formatRounding(): String  
 formatRoundingWithPrecision(): String  
 formatThousands(): String  
 NumberBase()  
 parseNumberString(): String

#### NumberBaseRoundType ■

P: mx.flash  
 → Object  
 DOWN: String  
 NEAREST: String  
 NONE: String  
 UP: String

#### NumberFormatter ■

P: mx.flash  
 → Formatter  
 decimalSeparatorFrom: String  
 decimalSeparatorTo: String  
 precision: Object  
 rounding: String  
 thousandsSeparatorFrom: String  
 thousandsSeparatorTo: String  
 useNegativeSign: Object  
 useThousandsSeparator: Object  
 format(): String  
 NumberFormatter()

#### NumberValidator ■

P: mx.validators  
 → Validator  
 allowNegative: Object  
 decimalPointCountError: String  
 decimalSeparator: String  
 domain: String  
 exceedsMaxError: String  
 integerError: String  
 invalidCharError: String  
 invalidFormatCharsError: String  
 lowerThanMinError: String  
 maxValue: Object  
 minValue: Object  
 negativeError: String  
 precision: Object  
 precisionError: String  
 separationError: String  
 thousandsSeparator: String  
 doValidation(): Array  
 NumberValidator()  
 validateNumber(): Array

#### NumericAxis ■

P: mx.charts.chartClasses  
 → AxisBase  
 IAxis  
 assignedMaximum: Number  
 assignedMinimum: Number  
 autoAdjust: Boolean  
 baseAtZero: Boolean  
 baseline: Number  
 computedInterval: Number  
 computedMaximum: Number  
 computedMinimum: Number  
 dataDescriptions: Array  
 labelCache: Array  
 labelFunction: Function  
 labelMaximum: Number  
 labelMinimum: Number  
 minorTickCache: Array  
 minorTicks: Array  
 padding: Number  
 parseFunction: Function  
 requiredDescribedFields: uint  
 ticks: Array  
 adjustMinMax(): void  
 buildLabelCache(): Boolean  
 buildMinorTickCache(): Array  
 filterCache(): void  
 formatForScreen(): String  
 getLabelEstimate(): AxisLabelSet  
 getLabels(): AxisLabelSet  
 guardMinMax(): Array  
 invalidateCache(): void  
 invertTransform(): Object  
 mapCache(): void  
 NumericAxis()  
 preferDropLabels(): Boolean  
 reduceLabels(): AxisLabelSet  
 transformCache(): void  
 update(): void

**NumericStepper** ■

P: mx.controls  
→ UIComponent  
*IDataRenderer*  
*IDropInListItemRenderer*  
*IFocusManagerComponent*  
*IMESupport*  
*IListItemRenderer*  
data: Object  
downArrowStyleFilters: Object  
imeMode: String  
inputFieldStyleFilters: Object  
listData: BaseListData  
maxChars: int  
maximum: Number  
minimum: Number  
nextValue: Number  
previousValue: Number  
stepSize: Number  
upArrowStyleFilters: Object  
value: Number  
NumericStepper()

**NumericStepperAutomationImpl** ■

P: mx.automation.delegates.  
controls  
→ UIComponentAutomationImpl  
componentInitialized(): void  
init(): void  
nsChangeHandler(): void  
NumericStepperAutomationImpl()

**NumericStepperDownSkin** ■

P: mx.skins.halo  
→ Border

**NumericStepperEvent** ■

P: mx.events  
→ Event  
CHANGE: String  
triggerEvent: Event  
value: Number  
NumericStepperEvent()

**NumericStepperUpSkin** ■

P: mx.skins.halo  
→ Border

**Object**

P: Top Level  
constructor: Object  
prototype: Object  
hasOwnProperty(): Boolean  
isPrototypeOf(): Boolean  
Object()  
propertyIsEnumerable(): Boolean  
setPropertyIsEnumerable(): void  
toString(): String  
valueOf(): Object

**ObjectEncoding**

P: flash.net  
→ Object  
AMF0: uint  
AMF3: uint  
DEFAULT: uint  
dynamicPropertyWriter: IDynamicPropertyWriter

**ObjectProxy** ■

P: mx.utils  
→ Proxy  
*IExternalizable*  
*IPropertyChangeNotifier*  
dispatcher: EventDispatcher  
notifiers: Object  
object: Object  
propertyList: Array  
proxyClass: Class  
type: QName  
uid: String  
addEventListener(): void  
callProperty(): \*  
deleteProperty(): Boolean  
dispatchEvent(): Boolean  
getComplexProperty(): \*  
getProperty(): \*  
hasEventListener(): Boolean  
hasProperty(): Boolean  
nextName(): String  
nextNameIndex(): int  
nextValue(): \*  
ObjectProxy()  
propertyChangeHandler(): void  
readExternal(): void  
removeEventListener(): void  
setProperty(): void  
setupPropertyList(): void  
willTrigger(): Boolean  
writeExternal(): void

**ObjectUtil** ■

P: mx.utils  
→ Object  
compare(): int  
copy(): Object  
dateCompare(): int  
getClassInfo(): Object  
hasMetadata(): Boolean  
isSimple(): Boolean  
numericCompare(): int  
stringCompare(): int  
toString(): String

**OLAPAttribute** ■

P: mx.olap  
→ OLAPHierarchy  
*OLAPAttribute*  
dataCompareFunction: Function  
dataField: String  
dataFunction: Function  
displayNameFunction: Function  
hasAll: Boolean  
members: IList  
OLAPAttribute()

**OLAPAxisPosition** ■

P: mx.olap  
→ Object  
*OLAPAxisPosition*  
members: IList  
OLAPAxisPosition()

**OLAPCell** ■

P: mx.olap  
→ Object  
*OLAPCell*  
formattedValue: String  
value: Number  
OLAPCell()

**OLAPCube** ■

P: mx.olap  
→ Proxy  
*OLAPCube*  
*IEventDispatcher*  
cubeBuildingTimeInterval: int  
dataProvider: ICollectionView  
dimensions: IList  
elements: Array  
measureDimensionName: String  
measures: IList  
name: String  
queryBuildingTimeInterval: int  
resultClass: Class  
addEventListener(): void  
cancelQuery(): void  
cancelRefresh(): void  
dispatchEvent(): Boolean  
execute(): Boolean  
findDimension(): IOLAPDimension  
hasEventListener(): Boolean  
OLAPCube()  
refresh(): void  
removeEventListener(): void  
toString(): String  
willTrigger(): Boolean

**OLAPDataGrid** ■

P: mx.controls  
→ AdvancedDataGrid  
COLUMN\_AXIS: int  
dataProvider: Object  
defaultCellString: String  
headerRendererProviders: Array  
itemRendererProviders: Array  
OLAP\_DIMENSION: int  
OLAP\_HIERARCHY: int  
OLAP\_LEVEL: int  
OLAP\_MEMBER: int  
ROW\_AXIS: int  
SLICER\_AXIS: int  
styleFunction: Function  
getCellRendererInfo(): OLAPDataGridItemRendererProvider  
getFormattedCellValue(): String  
getIndent(): int  
OLAPDataGrid()  
styleChanged(): void

**OLAPDataGridAutomationImpl** ■

P: mx.automation.delegates.  
advancedDataGrid  
→ AdvancedDataGridAutomationImpl  
automationTabularData: Object  
init(): void  
OLAPDataGridAutomationImpl()

**OLAPDataGridGroupRenderer** ■

P: mx.controls.olapDataGrid  
Classes  
→ UIComponent  
*IDataRenderer*  
*IDropInListItemRenderer*  
*IListItemRenderer*  
data: Object  
label: IUITextField  
listData: BaseListData  
OLAPDataGridGroupRenderer()

**OLAPDataGridGroupRendererAutomationImpl** ■

P: mx.automation.delegates.  
advancedDataGrid  
→ UIComponentAutomationImpl  
init(): void  
OLAPDataGridGroupRendererAutomationImpl()

**OLAPDataGridHeaderRendererProvider** ■

P: mx.controls.olapDataGrid  
Classes  
→ OLAPDataGridItemRendererProvider  
headerWordWrap:

**OLAPDataGridItemRendererProvider** ■

P: mx.controls.olapDataGrid  
Classes  
→ Object  
formatter: Formatter

**OLAPDataGridRendererProvider** ■

P: mx.controls.olapDataGrid  
Classes  
→ Object  
renderer: IFactory  
styleName: String  
type: int  
uniqueName: String

**OLAPDimension** ■

P: mx.olap  
→ OLAPElement  
*OLAPDimension*  
attributes: IList  
cube: IOLAPCube  
defaultMember: IOLAPMember  
elements: Array  
hierarchies: IList  
isMeasure: Boolean  
members: IList  
findAttribute(): IOLAPAttribute  
findHierarchy(): IOLAPHierarchy  
findMember(): IOLAPMember  
OLAPDimension()

**OLAPElement** ■

P: mx.olap  
→ Proxy  
*OLAPElement*  
dimension: IOLAPDimension  
displayName: String  
name: String  
uniqueName: String  
OLAPElement()  
toString(): String

**OLAPHierarchy** ■

P: mx.olap  
→ OLAPElement  
*OLAPHierarchy*  
allMemberName: String  
children: IList  
defaultMember: IOLAPMember  
elements: Array  
hasAll: Boolean  
levels: IList  
members: IList  
name: String  
findLevel(): IOLAPLevel  
findMember(): IOLAPMember  
OLAPHierarchy()

**OLAPLevel** ■

P: mx.olap  
→ OLAPElement  
*OLAPLevel*  
attribute: OLAPAttribute  
attributeName: String  
child: IOLAPLevel  
dataField: String  
depth: int  
hierarchy: IOLAPHierarchy  
members: IList  
name: String  
parent: IOLAPLevel  
uniqueName: String  
findMember(): IList  
OLAPLevel()

**OLAPMeasure** ■

P: mx.olap  
→ OLAPMember  
aggregator: Object  
OLAPMeasure()

**OLAPMember** ■

P: mx.olap  
→ OLAPElement  
*OLAPMeasure*  
children: IList  
dataField: String  
dimension: IOLAPDimension  
hierarchy: IOLAPHierarchy  
isAll: Boolean  
isMeasure: Boolean  
level: IOLAPLevel  
parent: IOLAPMember  
uniqueName: String  
findChildMember(): IOLAPMember  
OLAPMember()

**OLAPQuery** ■

P: mx.olap  
→ Object  
*OLAPQuery*  
axes: Array  
COLUMN\_AXIS: int  
ROW\_AXIS: int  
SLICER\_AXIS: int  
getAxis(): IOLAPQueryAxis  
setAxis(): void

**OLAPQueryAxis** ■

P: mx.olap  
→ Object  
*OLAPQueryAxis*  
axisOrdinal: int  
sets: Array  
tuples: Array  
addMember(): void  
addSet(): void  
addTuple(): void  
clear(): void  
OLAPQueryAxis()

**OLAPResult** ■

P: mx.olap  
→ Object  
*OLAPResult*  
axes: Array  
cellData: Array  
COLUMN\_AXIS: int  
query: IOLAPQuery  
ROW\_AXIS: int  
SLICER\_AXIS: int  
getAxis(): IOLAPResultAxis

getCell(): IOLAPCell  
hasRowData(): Boolean

**OLAPResultAxis** ■

P: mx.olap  
→ Object  
*OLAPResultAxis*  
positions: IList  
addPosition(): void  
removePosition(): Boolean

**OLAPSchema** ■

P: mx.olap  
→ Object  
*OLAPSchema*  
cubeArray: Array  
cubes: IList  
createCube(): IOLAPCube  
getCube(): IOLAPCube

**OLAPSet** ■

P: mx.olap  
→ Object  
*OLAPSet*  
tuples: Array  
addElement(): void  
addElements(): void  
addTuple(): void  
compareMembers(): int  
crossJoin(): IOLAPSet  
findCommonHierarchy(): IOLAPHierarchy  
hierarchy(): IOLAPSet  
OLAPSet(): void  
sortTuple(): int  
union(): IOLAPSet

**OLAPTrace** ■

P: mx.olap  
→ Object  
TRACE\_LEVEL\_1: int  
TRACE\_LEVEL\_2: int  
TRACE\_LEVEL\_3: int  
traceLevel: int  
traceOn: Boolean  
traceMsg(): void

**OLAPTuple** ■

P: mx.olap  
→ Object  
*OLAPTuple*  
explicitMembers: IList  
addMember(): void  
addMembers(): void  
OLAPTuple(): void

**Operation** ■

P: mx.rpc.remoting  
→ AbstractOperation  
*IMXMLSupport*  
argumentNames: Array  
makeObjectsBindable: Boolean  
Operation(): void  
send(): AsyncToken

**Operation** ■

P: mx.rpc.soap.mxml  
→ Operation  
*IMXMLSupport*  
concurrency: String  
showBusyCursor: Boolean  
cancel(): AsyncToken  
Operation(): void  
send(): AsyncToken

**Operation** ■

P: mx.rpc.remoting.mxml  
→ Operation  
*IMXMLASupport*  
concurrency: String  
showBusyCursor: Boolean  
cancel(): AsyncToken  
send(): AsyncToken

**Operation** ■

P: mx.rpc.soap  
→ AbstractOperation  
endpointURI: String  
headerFormat: String  
headers: Array  
httpHeaders: Object  
ignoreWhitespace: Boolean  
makeObjectsBindable: Boolean  
request: Object  
resultFormat: String  
resultHeaders: Array  
xmlSpecialCharsFilter: Function  
addHeader(): void  
addSimpleHeader(): void  
cancel(): AsyncToken  
clearHeaders(): void  
getHeader(): SOAPHeader  
Operation(): void  
removeHeader(): void

**OutputProgressEvent**

P: flash.events  
→ Event  
bytesPending: Number  
bytesTotal: Number  
OUTPUT\_PROGRESS: String  
clone(): Event  
OutputProgressEvent(): void  
toString(): String

**Panel** ■

P: mx.containers  
→ Container  
*IConstraintLayout*  
*IFontContextComponent*  
closeButtonStyleFilters: Object  
constraintColumns: Array  
constraintRows: Array  
controlBar: UIComponent  
fontContext: IFlexModuleFactory  
layout: String  
status: String  
statusTextField: IUITextField  
title: String  
titleBar: UIComponent  
titleLabel: Class  
titleLabel: IUITextField  
getHeaderHeight(): Number  
measure(): void  
Panel(): void  
startDragging(): void  
stopDragging(): void

**PanelAutomationImpl** ■

P: mx.automation.delegates.  
containers  
→ ContainerAutomationImpl  
init(): void  
PanelAutomationImpl()

**PanelSkin** ■

P: mx.skins.halo  
→ HaloBorder

**Parallel** ■

P: mx.effects  
→ CompositeEffect

**ParallelInstance** ■

P: mx.effects.effectClasses  
→ CompositeEffectInstance  
end(): void  
onEffectEnd(): void  
ParallelInstance()

**Pause** ■

P: mx.effects  
→ TweenEffect

**PauseInstance** ■

P: mx.effects.effectClasses  
→ TweenEffectInstance

**PhoneFormatter** ■

P: mx.flash  
→ Formatter  
areaCode: Object  
areaCodeFormat: String  
formatString: String  
validPatternChars: String  
format(): String  
PhoneFormatter()

**PhoneNumberValidator** ■

P: mx.validators  
→ Validator  
allowedFormatChars: String  
invalidCharError: String  
wrongLengthError: String  
doValidation(): Array  
PhoneNumberValidator()  
validatePhoneNumber(): Array

**PieChart** ■

P: mx.charts  
→ PolarChart

**PieSeries** ■

P: mx.charts.series  
→ Series  
angularAxis: IAxis  
explodeRadius: Number  
field: String  
fillFunction: Function  
items: Array  
itemType: Class  
labelField: String  
labelFunction: Function  
maxLabelRadius: Number  
nameField: String  
outerRadius: Number  
perWedgeExplodeRadius: Array  
renderDataType: Class  
reserveExplodeRadius: Number  
startAngle: Number  
beginInterpolation(): Object  
commitProperties(): void  
createChildren(): void  
dataToLocal(): Point  
getElementBounds(): void  
getMissingInterpolationValues(): void  
interpolate(): void  
localToData(): Array  
PieSeries()  
updateDisplayList(): void



**PieSeriesAutomationImpl** ■

P: mx.automation.delegates.charts  
 → SeriesAutomationImpl  
 init(): void  
 PieSeriesAutomationImpl()

**PieSeriesItem** ■

P: mx.charts.series.items  
 → ChartItem  
 angle: Number  
 fill: IFill  
 innerRadius: Number  
 labelAngle: Number  
 number: Number  
 origin: Point  
 outerRadius: Number  
 percentValue: Number  
 startAngle: Number  
 value: Object  
 PieSeriesItem()

**PieSeriesRenderData**

P: mx.charts.series.renderData  
 → RenderData  
 itemSum: Number  
 labelData: Object  
 labelScale: Number  
 PieSeriesRenderData()

**PixelSnapping**

P: flash.display  
 → Object  
[ALWAYS: String](#)  
[AUTO: String](#)  
[NEVER: String](#)

**PlotChart** ■

P: mx.charts  
 → CartesianChart

**PlotSeries** ■

P: mx.charts.series  
 → Series  
 fillFunction: Function  
 horizontalAxis: IAxis  
 items: Array  
 itemType: Class  
 renderData: Object  
 renderDataType: Class  
 verticalAxis: IAxis  
 xField: String  
 yField: String  
 applyItemRendererProperties(): void  
 commitProperties(): void  
 PlotSeries(): void

**PlotSeriesAutomationImpl** ■

P: mx.automation.delegates.charts  
 → SeriesAutomationImpl  
 init(): void  
 PlotSeriesAutomationImpl()

**PlotSeriesItem** ■

P: mx.charts.series.items  
 → ChartItem  
 fill: IFill  
 radius: Number  
 x: Number  
 xFilter: Number  
 xNumber: Number  
 xValue: Object  
 y: Number  
 yFilter: Number  
 yNumber: Number

yValue: Object  
 PlotSeriesItem()

**PlotSeriesRenderData** ■

P: mx.charts.series.renderData  
 → RenderData  
 radius: Number  
 PlotSeriesRenderData()

**PNGEncoder** ■

P: mx.graphics.codec  
 → Object  
 IImageEncoder  
 contentType: String  
 encode(): ByteArray  
 encodeByteArray(): ByteArray  
 PNGEncoder()

**Point**

P: flash.geom  
 → Object  
 length: Number  
 x: Number  
 y: Number  
 add(): Point  
 clone(): Point  
 distance(): Number  
 equals(): Boolean  
 interpolate(): Point  
 normalize(): void  
 offset(): void  
 Point()  
 polar(): Point  
 subtract(): Point  
 toString(): String

**PolarChart** ■

P: mx.charts.chartClasses  
 → ChartBase  
 angularAxis: IAxis  
 dataRegion: Rectangle  
 radialAxis: IAxis  
 commitProperties(): void  
 getFirstItem(): ChartItem  
 getLastItem(): ChartItem  
 getNextItem(): ChartItem  
 getPreviousItem(): ChartItem  
 PolarChart()  
 updateDisplayList(): void

**PolarDataCanvas** ■

P: mx.charts.chartClasses  
 → ChartElement  
 angularAxis: IAxis  
 dataTransform: DataTransform  
 includeInRanges: Boolean  
 radialAxis: IAxis  
 totalValue: Number  
 addChild(): DisplayObject  
 addChildAt(): DisplayObject  
 addDataChild(): void  
 beginBitmapFill(): void  
 beginFill(): void  
 clear(): void  
 commitProperties(): void  
 curveTo(): void  
 describeData(): Array  
 drawCircle(): void  
 drawEllipse(): void  
 drawRect(): void  
 drawRoundedRect(): void  
 endFill(): void  
 invalidateData(): void  
 invalidateDisplayList(): void

lineStyle(): void  
 lineTo(): void  
 mappingChanged(): void  
 moveTo(): void  
 PolarDataCanvas()  
 removeAllChildren(): void  
 removeChild(): DisplayObject  
 removeChildAt(): DisplayObject  
 stripNaNs(): void  
 updateDataChild(): void  
 updateDisplayList(): void  
 updateFilter(): void  
 validateData(): void  
 validateTransform(): Boolean

**PolarTransform** ■

P: mx.charts.chartClasses  
 → Data Transform  
[ANGULAR\\_AXIS: String](#)  
 origin: Point  
[RADIAL\\_AXIS: String](#)  
 radius: Number  
 PolarTransform()  
 setSize(): void  
 transformCache(): void

**PollingChannel** ■

P: mx.messaging.channels  
 → Channel  
 disablePolling(): void  
 enablePolling(): void  
 poll(): void  
 PollingChannel()

**PopUpButton** ■

P: mx.controls  
 → Button  
 openAlways: Boolean  
 popUp: IUIComponent  
 close(): void  
 open(): void  
 PopUpButton()

**PopUpButtonAutomationImpl** ■

P: mx.automation.delegates.controls  
 → ButtonAutomationImpl  
 init(): void  
 PopUpButtonAutomationImpl()

**PopUpButtonSkin** ■

P: mx.skins.halo  
 → UIComponent  
 IProgrammableSkin

**PopUpIcon** ■

P: mx.skins.halo  
 → ProgrammableSkin

**PopUpManager** ■

P: mx.managers  
 addPopUp(): void  
[bringToFront\(\): void](#)  
[centerPopUp\(\): void](#)  
[createPopUp\(\): IFlexDisplayObject](#)  
[removePopUp\(\): void](#)

**PopUpManagerChildList** ■

P: mx.managers  
 → Object  
[APPLICATION: String](#)  
[PARENT: String](#)  
[POPUP: String](#)

**PopUpMenuButton** ■

P: mx.controls  
 → PopUpButton  
 dataDescriptor: IMenuDataDescriptor  
 dataProvider: Object  
 labelField: String  
 labelFunction: Function  
 popUp: IUIComponent  
 showRoot: Boolean  
 PopUpMenuButton()

**PopUpMenuIcon** ■

P: mx.skins.halo  
 → PopUpIcon

**Preloader** ■

P: mx.preloaders  
 → Sprite  
 initialize(): void  
 Preloader()  
 registerApplication()

**PrintAdvancedDataGrid** ■

P: mx.printing  
 → AdvancedDataGrid  
 allowInteraction: Boolean  
 currentPageHeight: Number  
 displayIcons: Boolean  
 originalHeight: Number  
 sizeToPage: Boolean  
 source: AdvancedDataGrid  
 validNextPage: Boolean  
 validPreviousPage: Boolean  
 moveToFirstPage(): void  
 nextPage(): void  
 previousPage(): void  
 PrintAdvancedDataGrid(): void

**PrintDataGrid** ■

P: mx.printing  
 → DataGrid  
 currentPageHeight: Number  
 originalHeight: Number  
 sizeToPage: Boolean  
 validNextPage: Boolean  
 nextPage(): void  
 PrintDataGrid(): void

**PrintJob**

P: flash.printing  
 → EventDispatcher  
 orientation: String  
 pageHeight: int  
 pageWidth: int  
 paperHeight: int  
 paperWidth: int  
 addPage(): void  
 PrintJob()  
 send(): void  
 start(): Boolean

**PrintJobOptions**

P: flash.printing  
 → Object  
[printAsBitmap: Boolean](#)  
 PrintJobOptions()

**PrintJobOrientation**

P: flash.printing  
 → Object  
[LANDSCAPE: String](#)  
[PORTRAIT: String](#)

**PrintOLAPDataGrid** ■

P: mx.printing  
 → OLAPDataGrid  
 allowInteraction: Boolean  
 currentPageHeight: Number  
 originalHeight: Number  
 sizeToPage: Boolean  
 source: OLAPDataGrid  
 validNextPage: Boolean  
 validPreviousPage: Boolean  
 moveToFirstPage(): void  
 nextPage(): void  
 previousPage(): void  
 PrintOLAPDataGrid()

**ProgrammableSkin** ■

P: mx.skins  
 → FlexShape  
 IFlexDisplayObject  
 IInvalidating  
 ILayoutManagerClient  
 ISimpleStyleClient  
 IProgrammableSkin  
 initialized: Boolean  
 measuredHeight: Number  
 measuredWidth: Number  
 nestLevel: int  
 processedDescriptors: Boolean  
 styleName: Object  
 updateCompletePendingFlag: Boolean  
 drawRoundRect(): void  
 getStyle(): \*  
 horizontalGradientMatrix(): Matrix  
 invalidateDisplayList(): void  
 invalidateProperties(): void  
 invalidateSize(): void  
 move(): void  
 ProgrammableSkin()  
 rotatedGradientMatrix(): Matrix  
 setActualSize(): void  
 styleChanged(): void  
 updateDisplayList(): void  
 validateDisplayList(): void  
 validateNow(): void  
 validateProperties(): void  
 validateSize(): void  
 verticalGradientMatrix(): Matrix

**ProgressBar** ■

P: mx.controls  
 → UIComponent  
 IFontContextComponent  
 conversion: Number  
 direction: String  
 indeterminate: Boolean  
 label: String  
 labelPlacement: String  
 maximum: Number  
 minimum: Number  
 mode: String  
 percentComplete: Number  
 source: Object  
 value: Number  
 ProgressBar()  
 setProgress(): void

**ProgressBarAutomationImpl** ■

P: mx.automation.delegates.controls  
 → UIComponentAutomationImpl  
 init(): void  
 ProgressBarAutomationImpl()

**ProgressBarDirection** ■

P: mx.controls  
 → Object  
[LEFT: String](#)  
[RIGHT: String](#)

**ProgressBarLabelPlacement** ■

P: mx.controls  
 → Object  
[BOTTOM: String](#)  
[CENTER: String](#)  
[LEFT: String](#)  
[RIGHT: String](#)  
[TOP: String](#)

**ProgressBarMode** ■

P: mx.controls  
 → Object  
[EVENT: String](#)  
[MANUAL: String](#)  
[POLLED: String](#)

**ProgressBarSkin** ■

P: mx.skins.halo  
 → Border

**ProgressEvent**

P: flash.events  
 → Event  
 bytesLoaded: Number  
 bytesTotal: Number  
[PROGRESS: String](#)  
[SOCKET\\_DATA: String](#)  
 clone(): Event  
 ProgressEvent()  
 toString(): String

**ProgressIndeterminateSkin** ■

P: mx.skins.halo  
 → Border

**ProgressMaskSkin** ■

P: mx.skins.halo  
 → ProgrammableSkin

**ProgressTrackSkin** ■

P: mx.skins.halo  
 → Border

**PropertyChangeEvent** ■

P: mx.events  
 → Event  
 kind: String  
 newValue: Object  
 oldValue: Object  
 property: Object  
[PROPERTY\\_CHANGE: String](#)  
 source: Object  
[createUpdateEvent\(\): PropertyChangeEvent](#)  
 PropertyChangeEvent()

**PropertyChangeEventKind** ■

P: mx.events  
 → Object  
[DELETE: String](#)  
[UPDATE: String](#)

**PropertyChanges** ■

P: mx.effects.effectClasses  
 → Object  
 end: Object  
 start: Object  
 target: Object  
 PropertyChanges()

**Proxy**

P: flash.utils  
 → Object  
 callProperty(): \*  
 deleteProperty(): Boolean  
 getDescendants(): \*  
 getProperty(): \*  
 hasProperty(): Boolean  
 isAttribute(): Boolean  
 nextName(): String  
 nextNameIndex(): int  
 nextValue(): \*  
 setProperty(): void

**QName**

P: Top Level  
 → Object  
 localName: String  
 uri: String  
 QName()  
 toString(): String  
 valueOf(): QName

**Quadratic** ■

P: mx.effects.easing  
 → Object  
[easeIn\(\): Number](#)  
[easeInOut\(\): Number](#)  
[easeOut\(\): Number](#)

**QualifiedResourceManager** ■

P: mx.rpc.xml  
 → Object  
 resources: Array  
 resourcesMap: Object  
 addResource(): void  
 getResources(): Array  
 getResourcesForNamespace(): Array  
 getResourcesForURI(): Array  
 QualifiedResourceManager()

**Quartic** ■

P: mx.effects.easing  
 → Object  
[easeIn\(\): Number](#)  
[easeInOut\(\): Number](#)  
[easeOut\(\): Number](#)

**Quintic** ■

P: mx.effects.easing  
 → Object  
[easeIn\(\): Number](#)  
[easeInOut\(\): Number](#)  
[easeOut\(\): Number](#)

**RadialGradient** ■

P: mx.graphics  
 → GradientBase  
 IFill  
 angle: Number  
 focalPointRatio: Number  
 begin(): void  
 end(): void  
 RadialGradient()

**RadioButton** ■

P: mx.controls  
 → Button  
 IFocusManagerGroup  
 group: RadioButtonGroup  
 groupName: String  
 labelPlacement: String  
 value: Object  
 RadioButton()

**RadioButtonAutomationImpl** ■

P: mx.automation.delegates.controls  
 → ButtonAutomationImpl  
 init(): void  
 RadioButtonAutomationImpl()

**RadioButtonGroup** ■

P: mx.controls  
 → EventDispatcher  
 IMXMLObject  
 enabled: Boolean  
 labelPlacement: String  
 numRadioButtons: int  
 selectedValue: Object  
 selection: RadioButton  
 getRadioButtonAt(): RadioButton  
 initialized(): void  
 RadioButtonGroup()

**RadioButtonIcon** ■

P: mx.skins.halo  
 → Border

**RangeError**

P: Top Level  
 → Error  
 RangeError()

**Rectangle**

P: flash.geom  
 → Object  
 bottom: Number  
 bottomRight: Point  
 height: Number  
 left: Number  
 right: Number  
 size: Point  
 top: Number  
 topLeft: Point  
 width: Number  
 x: Number  
 y: Number  
 clone(): Rectangle  
 contains(): Boolean  
 containsPoint(): Boolean  
 containsRect(): Boolean  
 equals(): Boolean  
 inflate(): void  
 inflatePoint(): void  
 intersection(): Rectangle  
 intersects(): Boolean  
 isEmpty(): Boolean  
 offset(): void  
 offsetPoint(): void  
 Rectangle()  
 setEmpty(): void  
 toString(): String  
 union(): Rectangle

**RectangularBorder** ■

P: mx.skins  
 → Border  
 IRectangularBorder  
 backgroundImageBounds: Rectangle  
 hasBackgroundImage: Boolean  
 layoutBackgroundImage(): void  
 RectangularBorder()

**RectangularDropShadow** ■

P: mx.graphics  
 → Object  
 alpha: Number  
 angle: Number

**blRadius:** Number  
**brRadius:** Number  
**color:** int  
**distance:** Number  
**tlRadius:** Number  
**trRadius:** Number  
**drawShadow()** : void  
**RectangularDropShadow()**

### ReferenceError

**P:** Top Level  
 → Error  
**ReferenceError()**

### RegExp

**P:** Top Level  
 → Object  
**detail:** Boolean  
**extended:** Boolean  
**global:** Boolean  
**ignoreCase:** Boolean  
**lastIndex:** Number  
**multiline:** Boolean  
**source:** String  
**exec()** : Object  
**RegExp()**  
**test()** : Boolean

### RegExpValidationResult

**P:** mx.validators  
 → Validator  
**matchedIndex:** int  
**matchedString:** String  
**matchedSubstrings:** Array  
**RegExpValidationResult()**

### RegExpValidator

**P:** mx.validators  
 → ValidationResult  
**expression:** String  
**flags:** String  
**noExpressionError:** String  
**noMatchError:** String  
**doValidation()** : Array  
**RegExpValidator()** : void

### RemoteObject

**P:** mx.rpc.remoting  
 → AbstractService  
**makeObjectsBindable:** Boolean  
**source:** String  
**getOperation()** : AbstractOperation  
**RemoteObject()**  
**setRemoteCredentials()** : void  
**toString()** : String

### RemoteObject

**P:** mx.rpc.remoting.mxml  
 → RemoteObject  
**IMXMLSupport**  
**IMXMLObject**  
**concurrency:** String  
**endpoint:** String  
**showBusyCursor:** Boolean  
**getOperation()** : AbstractOperation  
**initialized()** : void  
**RemoteObject()** : void

### RemotingMessage

**P:** mx.messaging.messages  
 → AbstractMessage  
**operation:** String  
**source:** String  
**RemotingMessage()**

### RemoveChild

**P:** mx.states  
 → Object  
**IOOverride**  
**target:** DisplayObject  
**apply()** : void  
**initialize()** : void  
**remove()** : void  
**RemoveChild()**

### RemoveChildAction

**P:** mx.effects  
 → Effect

### RemoveChildActionInstance

**P:** mx.effects.effectClasses  
 → ActionEffectInstance

### RemoveItemAction

**P:** mx.effects  
 → Effect

### RemoveItemActionInstance

**P:** mx.effects.effectClasses  
 → ActionEffectInstance

### RenderData

**P:** mx.charts.chartClasses  
 → Effect  
**bounds:** Rectangle  
**cache:** Array  
**elementBounds:** Array  
**filteredCache:** Array  
**length:** uint  
**visibleRegion:** Rectangle  
**clone()** : RenderData  
**RenderData()**

### Repeater

**P:** mx.core  
 → UIComponent  
**IRepeater**  
**childDescriptors:** Array  
**container:** IContainer  
**count:** int  
**currentIndex:** int  
**currentItem:** Object  
**dataProvider:** Object  
**recycleChildren:** Boolean  
**startIndex:** int  
**executeChildBindings()** : void  
**initializeRepeater()** : void  
**Repeater()**

### RepeaterAutomationImpl

**P:** mx.automation.delegates.core  
 → UIComponentAutomationImpl  
**automationTabularData:** Object  
**init()** : void  
**RepeaterAutomationImpl()**  
**p:** UIComponentAutomationImpl

### Resize

**P:** mx.effects  
 → TweenEffect  
**heightBy:** Number  
**heightFrom:** Number  
**heightTo:** Number  
**hideChildrenTargets:** Array  
**widthBy:** Number  
**widthFrom:** Number  
**widthTo:** Number  
**Resize()**

### ResizeEvent

**P:** mx.events  
 → Event  
**oldHeight:** Number  
**oldWidth:** Number  
**RESIZE:** String  
**ResizeEvent()**

### ResizeInstance

**P:** mx.effects.effectClasses  
 → TweenEffectInstance  
**heightBy:** Number  
**heightFrom:** Number  
**heightTo:** Number  
**hideChildrenTargets:** Array  
**widthBy:** Number  
**widthFrom:** Number  
**widthTo:** Number  
**ResizeInstance()**

### ResourceBundle

**P:** mx.resources  
 → Object  
**IResourceBundle**  
**bundleName:** String  
**content:** Object  
**locale:** String  
**getContent()** : Object  
**ResourceBundle()**

### ResourceEvent

**P:** mx.events  
 → ProgressEvent  
**COMPLETE:** String  
**ERROR:** String  
**errorText:** String  
**PROGRESS:** String  
**ResourceEvent()**

### ResourceManager

**P:** mx.resources  
 → EventDispatcher  
**localeChain:** Array  
**addResourceBundle()** : void  
**findResourceBundleWithResource()** : IResourceBundle  
**getBoolean()** : Boolean  
**getBundleNamesForLocale()** : Array  
**getClass()** : Class  
**getInstance()** : IResourceManager  
**getInt()** : int  
**getLocales()** : Array  
**getNumber()** : Number  
**getObject()** : \*  
**getResourceBundle()** : IResourceBundle  
**getString()** : String  
**getStringArray()** : Array  
**getUint()** : uint  
**loadResourceModule()** : IEventDispatcher  
**removeResourceBundle()** : void  
**removeResourceBundlesForLocale()** : void  
**ResourceManager()**  
**unloadResourceModule()** : void  
**update()** : void

### ResourceManagerImpl

**P:** mx.resources  
 → EventDispatcher  
**IResourceManager**  
**localeChain:** Array  
**ResourceManagerImpl()**  
**addResourceBundle()** : void

**findResourceBundleWithResource()** : void  
**getBoolean()** : Boolean  
**getBundleNameForLocale()** : Array  
**getClass()** : Class  
**getInstance()** : IResourceManager  
**getInt()** : int  
**getLocales()** : Array  
**getNumber()** : Number  
**getObject()** : \*  
**getResourceBundle()** : IResourceBundle  
**getString()** : String  
**getStringArray()** : Array  
**getUint()** : uint  
**loadResourceModule()** : IEventDispatcher  
**removeResourceBundle()** : void  
**removeResourceBundlesForLocale()** : void  
**unloadResourceModule()** : void  
**update()** : void

### Responder

**P:** flash.net  
 → Object

### Responder

**P:** mx.rpc  
 → Object  
**IResponder**  
**fault()** : void  
**Responder()** : void  
**result()** : void

### ResultEvent

**P:** mx.rpc.events  
 → AbstractEvent  
**headers:** Object  
**RESULT:** String  
**result:** Object  
**ResultEvent()**  
**toString()** : String

### RevocationCheckSettings

**P:** flash.security  
 → Object  
**ALWAYS\_REQUIRED:** \*  
**BEST\_EFFORT:** \*  
**NEVER:** \*  
**REQUIRED\_IF\_AVAILABLE:** \*

### RichTextEditor

**P:** mx.controls  
 → Panel  
**defaultLinkProtocol:** String  
**htmlText:** String  
**selection:** TextRange  
**showControlBar:** Boolean  
**showToolTips:** Boolean  
**text:** String

### Rotate

**P:** mx.effects  
 → TweenEffect  
**angleFrom:** Number  
**angleTo:** Number  
**hideFocusRing:** Boolean  
**originX:** Number  
**originY:** Number  
**Rotate()**

### RotateInstance

**P:** mx.effects.effectClasses  
 → TweenEffectInstance  
**angleFrom:** Number  
**angleTo:** Number  
**originX:** Number  
**originY:** Number  
**RotateInstance()** : void

### RoundedRectangle

**P:** mx.graphics  
 → Rectangle  
**cornerRadius:** Number  
**RoundedRectangle()**

### RSLEvent

**P:** mx.events  
 → ProgressEvent  
**errorText:** String  
**RSL\_COMPLETE:** String  
**RSL\_ERROR:** String  
**RSL\_PROGRESS:** String  
**rsIndex:** int  
**rsTotal:** int  
**url:** URLRequest  
**RSLEvent()**

### Sample

**P:** flash.sampler  
 → Object  
**stack:** Array  
**time:** Number

### Scene

**P:** flash.display  
 → Object  
**labels:** Array  
**name:** String  
**numFrames:** int

### SchemaTypeRegistry

**P:** mx.rpc.xml  
 → Object  
**getClass()** : Class  
**getCollectionClass()** : Class  
**getInstance()** : SchemaTypeRegistry  
**registerClass()** : void  
**registerCollectionClass()** : void  
**unregisterClass()** : void  
**unregisterCollectionClass()** : void

### Screen

**P:** flash.display  
 → EventDispatcher  
**bounds:** Rectangle  
**colorDepth:** int  
**mainScreen:** Screen  
**screens:** Array  
**visibleBounds:** Rectangle  
**getScreensForRectangle()** : Array

### ScreenMouseEvent

**P:** flash.events  
 → MouseEvent  
**screenX:** Number  
**screenY:** Number  
**clone()** : Event  
**ScreenMouseEvent()**  
**toString()** : String

### ScriptTimeoutError

**P:** flash.errors  
 → Error  
**ScriptTimeoutError()**

### ScrollArrowSkin

**P:** mx.skins.halo  
 → Border

### ScrollBar

**P:** mx.controls.scrollClasses  
 → UIComponent  
**direction:** String  
**downArrowStyleFilters:** Object  
**lineScrollSize:** Number  
**maxScrollPosition:** Number  
**minScrollPosition:** Number  
**pageScrollSize:** Number  
**pageSize:** Number  
**scrollPosition:** Number  
**THICKNESS:** Number  
**thumbStyleFilters:** Object  
**upArrowStyleFilters:** Object  
**ScrollBar()**  
**setScrollProperties()** : void

### ScrollBarAutomationImpl

**P:** mx.automation.delegates.controls  
 → UIComponentAutomationImpl  
**init()** : void  
**ScrollBarAutomationImpl()**

### ScrollBarDirection

**P:** mx.controls.scrollClasses  
 → Object  
**HORIZONTAL:** String  
**VERTICAL:** String

### ScrollControlBase

**P:** mx.core  
 → UIComponent  
**border:** IFlexDisplayObject  
**borderMetrics:** EdgeMetrics  
**horizontalScrollBar:** ScrollBar  
**horizontalScrollPolicy:** String  
**horizontalScrollPosition:** Number  
**liveScrolling:** Boolean  
**maskShape:** Shape  
**maxHorizontalScrollPosition:** Number  
**maxVerticalScrollPosition:** Number  
**scrollAreaChanged:** Boolean  
**scrollTipFunction:** Function  
**showScrollTips:** Boolean  
**verticalScrollBar:** ScrollBar  
**verticalScrollPolicy:** String  
**verticalScrollPosition:** Number  
**viewMetrics:** EdgeMetrics  
**createBorder()** : void  
**createChildren()** : void  
**layoutChrome()** : void  
**mouseWheelHandler()** : void  
**roomForScrollBar()** : Boolean  
**ScrollControlBase()**  
**scrollHandler()** : void  
**setScrollBarProperties()** : void

### ScrollControlBaseAutomationImpl

**P:** mx.automation.delegates.core  
 → UIComponentAutomationImpl  
**init()** : void  
**replayAutomatableEvent()** : Boolean  
**ScrollControlBaseAutomationImpl()**

### ScrollEvent

**P:** mx.events  
 → Event  
**delta:** Number

**detail:** String  
**direction:** String  
**position:** Number  
**SCROLL:** String  
**ScrollEvent()**

### ScrollEventDetail

**P:** mx.events  
 → Object  
**AT\_BOTTOM:** String  
**AT\_LEFT:** String  
**AT\_RIGHT:** String  
**AT\_TOP:** String  
**LINE\_DOWN:** String  
**LINE\_LEFT:** String  
**LINE\_RIGHT:** String  
**LINE\_UP:** String  
**PAGE\_DOWN:** String  
**PAGE\_LEFT:** String  
**PAGE\_RIGHT:** String  
**PAGE\_UP:** String  
**THUMB\_POSITION:** String  
**THUMB\_TRACK:** String

### ScrollEventDirection

**P:** mx.events  
 → Object  
**HORIZONTAL:** String  
**VERTICAL:** String

### ScrollPolicy

**P:** mx.core  
 → Object  
**AUTO:** String  
**OFF:** String  
**ON:** String

### ScrollThumb

**P:** mx.controls.scrollClasses  
 → Button

### ScrollThumbSkin

**P:** mx.skins.halo  
 → Border

### ScrollTrackSkin

**P:** mx.skins.halo  
 → Border

### SecureAMFChannel

**P:** mx.messaging.channels  
 → PollingChannel  
**protocol:** String  
**SecureAMFChannel()**

### SecureHTTPChannel

**P:** mx.messaging.channels  
 → PollingChannel  
**protocol:** String  
**SecureHTTPChannel()**

### SecureStreamingAMFChannel

**P:** mx.messaging.channels  
 → StreamingAMFChannel  
**protocol:** String  
**SecureStreamingAMFChannel()**

### SecureStreamingHTTPChannel

**P:** mx.messaging.channels  
 → StreamingHTTPChannel  
**protocol:** String  
**SecureStreamingHTTPChannel()**

### Security

**P:** flash.system  
 → Object  
**APPLICATION:** String  
**exactSettings:** Boolean  
**LOCAL\_TRUSTED:** String  
**LOCAL\_WITH\_FILE:** String  
**LOCAL\_WITH\_NETWORK:** String  
**REMOTE:** String  
**sandboxType:** String  
**allowDomain()** : void  
**allowInsecureDomain()** : void  
**loadPolicyFile()** : void  
**showSettings()** : void

### SecurityDomain

**P:** flash.system  
 → Object  
**currentDomain:** SecurityDomain

### SecurityError

**P:** Top Level  
 → Error  
**SecurityError()** : void

### SecurityErrorEvent

**P:** flash.events  
 → ErrorEvent  
**SECURITY\_ERROR:** String  
**clone()** : Event  
**SecurityErrorEvent()**  
**toString()** : String

### SecurityPanel

**P:** flash.system  
 → Object  
**CAMERA:** String  
**DEFAULT:** String  
**DISPLAY:** String  
**LOCAL\_STORAGE:** String  
**MICROPHONE:** String  
**PRIVACY:** String  
**SETTINGS\_MANAGER:** String

### Sequence

**P:** mx.effects  
 → CompositeEffect

### SequenceInstance

**P:** mx.effects.effectClasses  
 → CompositeEffectInstance  
**end()** : void  
**onEffectEnd()** : void  
**SequenceInstance()**

### Series

**P:** mx.charts.chartClasses  
 → ChartElement  
**dataFunction:** Function  
**dataTipItems:** Array  
**dataTransform:** DataTransform  
**displayName:** String  
**filterData:** Boolean  
**interactive:** Boolean  
**items:** Array  
**legendData:** Array  
**renderData:** Object  
**selectable:** Boolean  
**selectedIndex:** int  
**selectedIndices:** Array  
**selectedItem:** ChartItem  
**selectedItems:** Array  
**transitionRenderData:** Object  
**beginInterpolation()** : Object

cacheDefaultValues() : Boolean  
 cacheIndexValues() : Boolean  
 cacheNamedValues() : Boolean  
 claimStyles() : uint  
 commitProperties() : void  
 endInterpolation() : void  
 extractMinInterval() : void  
 extractMinMax() : void  
 getAxis() : IAxis  
 getElementBounds() : void  
 getItemsInRegion() : Array  
 getMissingInterpolationValues() : void  
 getRenderDataForTransition() : Object  
 initializeInterpolationData() : Object  
 interpolate() : void  
 invalidateData() : void  
 invalidateFilter() : void  
 invalidateMapping() : void  
 invalidateTransform() : void  
 invalidateTransitions() : void  
 legendDataChanged() : void  
 Series()  
 setActualSize() : void  
 setAxis() : void  
 stripNaNs() : void  
 stylesInitialized() : void  
 updateData() : void  
 updateDisplayList() : void  
 updateFilter() : void  
 updateMapping() : void  
 updateTransform() : void  
 validateData() : void  
 validateTransform() : void

**SeriesAutomationImpl** ■  
 P: mx.automation.delegates.charts  
 → UIComponentAutomationImpl  
 init() : void  
 SeriesAutomationImpl()

**SeriesEffect** ■  
 P: mx.charts.effects  
 → TweenEffect  
 elementOffset: Number  
 minimumElementDuration: Number  
 offset: Number  
 type: String  
 SeriesEffect()

**SeriesEffectInstance** ■  
 P: mx.charts.effects.effectClasses  
 → TweenEffectInstance  
 elementOffset: Number  
 interpolationValues: Array  
 minimumElementDuration: Number  
 offset: Number  
 targetSeries: Series  
 type: String  
 beginTween() : void  
 SeriesEffectInstance()

**SeriesInterpolate** ■  
 P: mx.charts.effects  
 → SeriesEffect

**SeriesInterpolateInstance** ■  
 P: mx.charts.effects.effectClasses  
 → SeriesEffectInstance

**SeriesSlide** ■  
 P: mx.charts.effects  
 → SeriesEffect  
 direction: String  
 initInstance() : void  
 SeriesSlide()

**SeriesSlideInstance** ■  
 P: mx.charts.effects.effectClasses  
 → SeriesEffectInstance  
 direction: String  
 SeriesSlideInstance()

**SeriesZoom** ■  
 P: mx.charts.effects  
 → SeriesEffect  
 horizontalFocus: String  
 relativeTo: String  
 verticalFocus: String  
 SeriesZoom()

**SeriesZoomInstance** ■  
 P: mx.charts.effects.effectClasses  
 → SeriesEffectInstance  
 horizontalFocus: String  
 relativeTo: String  
 verticalFocus: String  
 SeriesZoomInstance()

**ServerConfig** ●  
 P: mx.messaging.config  
 → Object  
 xml: XML  
 checkChannelConsistency() : void  
 getChannel() : Channel  
 getChannelSet() : ChannelSet  
 getProperties() : XMLList

**ServiceMonitor** ●  
 P: air.net  
 → EventDispatcher  
 available: Boolean  
 lastUpdated: Date  
 pollInterval: Number  
 running: Boolean  
 augmentPrototype() : void  
 checkStatus() : void  
 ServiceMonitor()  
 start() : void  
 stop() : void  
 toString() : String

**SetEventHandler** ■  
 P: mx.states  
 → EventDispatcher  
 IOverride  
 handlerFunction: Function  
 name: String  
 target: EventDispatcher  
 apply() : void  
 initialize() : void  
 remove() : void  
 SetEventHandler()

**SetProperty** ■  
 P: mx.states  
 → Object  
 IOverride  
 name: String  
 target: Object  
 value:  
 apply() : void  
 initialize() : void  
 remove() : void  
 SetProperty()

**SetPropertyAction** ■  
 P: mx.effects  
 → Effect  
 name: String  
 value:

SetPropertyAction()  
 k Effect

**SetPropertyActionInstance** ■  
 P: mx.effects.effectClasses  
 → ActionEffectInstance  
 name: String  
 value: \*  
 SetPropertyActionInstance()

**SetStyle** ■  
 P: mx.states  
 → Object  
 IOverride  
 name: String  
 target: IStyleClient  
 value: Object  
 apply() : void  
 initialize() : void  
 remove() : void  
 SetStyle()

**SetStyleAction** ■  
 P: mx.effects  
 → Effect  
 name: String  
 relevantStyles: Array  
 value:  
 SetStyleAction()

**SetStyleActionInstance** ■  
 P: mx.effects.effectClasses  
 → ActionEffectInstance  
 name: String  
 value: \*  
 SetStyleActionInstance()

**SHA256** ■  
 P: mx.utils  
 → Object  
 TYPE\_ID : String = "SHA-256"  
 computerDigest():String

**ShadowBoxItemRenderer** ■  
 → ProgrammaticSkin  
 P: mx.charts.renderers  
 IDataRenderer  
 data: Object  
 ShadowBoxItemRenderer()

**ShadowLineRenderer** ■  
 P: mx.charts.renderers  
 → ProgrammaticSkin  
 IDataRenderer  
 data: Object  
 ShadowLineRenderer()

**Shape**  
 P: flash.display  
 → Display Object  
 graphics: Graphics  
 Shape()

**SharedObject**  
 P: flash.net  
 → EventDispatcher  
 allowThumbOverlap: Boolean  
 client: Object  
 data: Object  
 defaultObjectEncoding: uint  
 fps: Number  
 objectEncoding: uint  
 size: uint  
 clear() : void  
 close() : void

connect() : void  
 flush() : String  
 getLocal() : SharedObject  
 getRemote() : SharedObject  
 send() : void  
 setDirty() : void  
 setProperty() : void

**SharedObjectFlushStatus**  
 P: flash.net  
 → Object  
 FLUSHED: String  
 PENDING: String

**SignatureStatus** ●  
 P: flash.net  
 → Object  
 INVALID: String  
 UNKNOWN: String  
 VALID: String

**SignerTrustSettings** ●  
 P: flash.security  
 → Object  
 CODE\_SIGNING: String  
 PLAYLIST\_SIGNING: String  
 SIGNING: String

**SimpleButton**  
 P: flash.display  
 → InteractiveObject  
 downState: DisplayObject  
 enabled: Boolean  
 name: String  
 overState: DisplayObject  
 soundTransform: SoundTransform  
 trackAsMenu: Boolean  
 upState: DisplayObject  
 useHandCursor: Boolean  
 SimpleButton()

**SimpleXMLDecoder** ■  
 P: mx.rpc.xml  
 → Object  
 decodeXML() : Object  
 getLocalName() : String  
 simpleType() : Object

**SimpleXMLEncoder** ■  
 P: mx.rpc.xml  
 → Object  
 encodeDate() : String  
 encodeValue() : XMLNode  
 SimpleXMLEncoder()

**Sine** ■  
 P: mx.effects.easing  
 → Object  
 easeIn() : Number  
 easeInOut() : Number  
 easeOut() : Number

**Slider** ■  
 P: mx.controls.sliderClasses  
 → UIComponent  
 allowThumbOverlap: Boolean  
 client: Object  
 data: Object  
 dataTipFormatFunction: Function  
 direction: String  
 labels: Array  
 liveDragging: Boolean  
 maximum: Number  
 minimum: Number  
 showDataTip: Boolean  
 sliderDataTipClass: Class

sliderThumbClass: Class  
 snapInterval: Number  
 thumbCount: int  
 thumbStyleFilters: Object  
 tickInterval: Number  
 tickValues: Array  
 value: Number  
 values: Array  
 getThumbAt() : SliderThumb  
 measure() : void  
 setThumbValueAt() : void  
 Slider()  
 updateDisplayList() : void

**SliderAutomationImpl** ■  
 P: mx.automation.delegates.  
 controls  
 → UIComponentAutomationImpl  
 init() : void  
 SliderAutomationImpl()

**SliderDataTip** ■  
 P: mx.controls.sliderClasses  
 → ToolTip

**SliderDirection** ■  
 P: mx.controls.sliderClasses  
 → Object  
 HORIZONTAL: String  
 VERTICAL: String

**SliderEvent** ■  
 P: mx.events  
 → Event  
 CHANGE: String  
 clickTarget: String  
 keyCode: int  
 THUMB\_DRAG: String  
 THUMB\_PRESS: String  
 THUMB\_RELEASE: String  
 thumbIndex: int  
 triggerEvent: Event  
 value: Number  
 SliderEvent()

**SliderEventClickTarget** ■  
 P: mx.events  
 → Object  
 THUMB: String  
 TRACK: String

**SliderHighlightSkin** ■  
 P: mx.skins.halo  
 → Border

**SliderLabel** ■  
 P: mx.controls.sliderClasses  
 → Label

**SliderThumb** ■  
 P: mx.controls.sliderClasses  
 → Button  
 xPosition: Number  
 SliderThumb()

**SliderThumbSkin** ■  
 P: mx.skins.halo  
 → Border  
 drawThumbState() : void  
 SliderThumbSkin()

**SliderTrackSkin** ■  
 P: mx.skins.halo  
 → Border

**SOAPFault** ■  
 P: mx.rpc.soap  
 → Fault  
 detail: String  
 element: XML  
 faultactor: String  
 faultcode: QName  
 faultstring: String  
 SOAPFault()  
 toString() : String

**SOAPHeader** ■  
 P: mx.rpc.soap  
 → Object  
 content: Object  
 mustUnderstand: Boolean  
 qname: QName  
 role: String  
 SOAPHeader()

**SOAPMessage** ■  
 P: mx.messaging.messages  
 → HTTPRequestMessage  
 SOAP\_ACTION\_HEADER: String  
 getSOAPAction() : String  
 SOAPMessage()

**SocialSecurityValidator** ■  
 P: mx.validators  
 → Validator  
 allowedFormatChars: String  
 invalidCharError: String  
 wrongFormatError: String  
 zeroStartError: String  
 doValidation() : Array  
 SocialSecurityValidator()  
 validateSocialSecurity() : Array

**Socket**  
 P: flash.net  
 → EventDispatcher  
 IDataInput  
 IDataOutput  
 bytesAvailable: uint  
 connected: Boolean  
 endian: String  
 objectEncoding: uint  
 close() : void  
 connect() : void  
 flush() : void  
 readBoolean() : Boolean  
 readByte() : int  
 readBytes() : void  
 readDouble() : Number  
 readFloat() : Number  
 readInt() : int  
 readMultiByte() : String  
 readObject() : \*  
 readShort() : int  
 readUnsignedByte() : uint  
 readUnsignedInt() : uint  
 readUnsignedShort() : uint  
 readUTF() : String  
 readUTFBytes() : String  
 Socket()  
 writeBoolean() : void  
 writeByte() : void  
 writeBytes() : void  
 writeDouble() : void  
 writeFloat() : void  
 writeInt() : void  
 writeMultiByte() : void  
 writeObject() : void

writeShort() : void  
 writeUnsignedInt() : void  
 writeUTF() : void  
 writeUTFBytes() : void

**SocketMonitor** ●  
 P: air.net  
 → ServiceMonitor  
 host: String  
 port: int  
 checkStatus() : void  
 SocketMonitor()  
 toString() : String

**SolidColor** ■  
 P: mx.graphics  
 → EventDispatcher  
 IFill  
 alpha: Number  
 color: uint  
 begin() : void  
 end() : void  
 SolidColor()

**Sort** ●  
 P: mx.collections  
 → EventDispatcher  
 ANY\_INDEX\_MODE: String  
 compareFunction: Function  
 fields: Array  
 FIRST\_INDEX\_MODE: String  
 LAST\_INDEX\_MODE: String  
 unique: Boolean  
 findItem() : int  
 propertyAffectsSort() : Boolean  
 reverse() : void  
 Sort()  
 sort() : void

**SortError** ●  
 P: mx.collections.errors  
 → Error

**SortField** ●  
 P: mx.collections  
 → EventDispatcher  
 caseInsensitive: Boolean  
 compareFunction: Function  
 descending: Boolean  
 name: String  
 numeric: Object  
 reverse() : void  
 SortField()

**SortInfo** ■  
 P: mx.controls.advanced  
 → DataGridClasses  
 ACTUALSORT: String  
 descending: Boolean  
 PROPOSEDSORT: String  
 sequenceNumber: int  
 status: String  
 SortInfo()

**Sound**  
 P: flash.media  
 → EventDispatcher  
 bytesLoaded: uint  
 bytesTotal: int  
 id3: ID3Info  
 isBuffering: Boolean  
 length: Number  
 url: String

close() : void  
 load() : void  
 play() : SoundChannel  
 Sound()

**SoundAsset** ■  
 P: mx.core  
 → Sound  
 IFlexAsset

**SoundChannel**  
 P: flash.media  
 → EventDispatcher  
 leftPeak: Number  
 position: Number  
 rightPeak: Number  
 soundTransform: SoundTransform  
 stop() : void

**SoundEffect** ■  
 P: mx.effects  
 → Effect  
 autoLoad: Boolean  
 bufferTime: Number  
 isLoading: Boolean  
 loops: int  
 panEasingFunction: Function  
 panFrom: Number  
 panTo: Number  
 sound: Sound  
 source: Object  
 startTime: Number  
 useDuration: Boolean  
 volumeEasingFunction: Function  
 volumeFrom: Number  
 volumeTo: Number  
 load() : void  
 SoundEffect()

**SoundEffectInstance** ■  
 P: mx.effects.effectClasses  
 → EffectInstance  
 bufferTime: Number  
 isLoading: Boolean  
 loops: int  
 panEasingFunction: Function  
 panFrom: Number  
 panTo: Number  
 sound: Sound  
 soundChannel: SoundChannel  
 source: Object  
 startTime: Number  
 useDuration: Boolean  
 volumeEasingFunction: Function  
 volumeFrom: Number  
 volumeTo: Number  
 SoundEffectInstance()

**SoundLoaderContext**  
 P: flash.media  
 → Object  
 bufferTime: Number  
 checkPolicyFile: Boolean  
 SoundLoaderContext()

**SoundMixer**  
 P: flash.media  
 → Object  
 bufferTime: int  
 soundTransform: SoundTransform  
 areSoundsInaccessible() : Boolean  
 computeSpectrum() : void  
 stopAll() : void



**SoundTransform**

P: flash.media  
 → Object  
 leftToLeft: Number  
 leftToRight: Number  
 pan: Number  
 rightToLeft: Number  
 rightToRight: Number  
 volume: Number  
 SoundTransform()

**Spacer**

P: mx.controls  
 → UIComponent

**SpreadMethod**

P: flash.display  
 → Object  
 PAD: String  
 REFLECT: String  
 REPEAT: String

**Sprite**

P: flash.display  
 → DisplayObjectContainer  
 buttonMode: Boolean  
 dropTarget: DisplayObject  
 graphics: Graphics  
 hitArea: Sprite  
 soundTransform: SoundTransform  
 useHandCursor: Boolean  
 Sprite()  
 startDrag(): void  
 stopDrag(): void

**SpriteAsset**

P: mx.core  
 → FlexSprite  
 IFlexAsset  
 IFlexDisplayObject  
 IBorder  
 borderMetrics: EdgeMetrics  
 measuredHeight: Number  
 measuredWidth: Number  
 move(): void  
 setActualSize(): void  
 SpriteAsset()

**SQLCollationType**

P: flash.data  
 → Object  
 BINARY: String  
 NO\_CASE: String

**SQLColumnNameStyle**

P: flash.data  
 → Object  
 DEFAULT: String  
 LONG: String  
 SHORT: String

**SQLColumnSchema**

P: flash.data  
 → Object  
 allowNull: Boolean  
 autoIncrement: Boolean  
 dataType: String  
 defaultCollationType: String  
 name: String  
 primaryKey: Boolean  
 SQLColumnSchema()

**SQLConnection**

P: flash.data  
 → EventDispatcher  
 autoCompact: Boolean  
 cacheSize: uint  
 columnNameStyle: String  
 connected: Boolean  
 inTransaction: Boolean  
 lastInsertRowID: Number  
 pageSize: uint  
 totalChanges: Number  
 addEventListener(): void  
 analyze(): void  
 attach(): void  
 begin(): void  
 cancel(): void  
 close(): void  
 commit(): void  
 compact(): void  
 deanalyze(): void  
 detach(): void  
 getSchemaResult(): SQLSchemaResult  
 loadSchema(): void  
 open(): void  
 openAsync(): void  
 removeEventListener(): void  
 rollback(): void  
 SQLConnection()

**SQLException**

P: flash.errors  
 → Error  
 details: String  
 operation: String  
 SQLException(): void  
 toString(): String

**SQLExceptionEvent**

P: flash.events  
 → ErrorEvent  
 ERROR: String  
 error: SQLException  
 clone(): Event  
 SQLExceptionEvent()  
 toString(): String

**SQLExceptionOperation**

P: flash.errors  
 → Object  
 ANALYZE: String  
 ATTACH: String  
 BEGIN: String  
 CLOSE: String  
 COMMIT: String  
 COMPACT: String  
 DEANALYZE: String  
 DETACH: String  
 EXECUTE: String  
 OPEN: String  
 ROLLBACK: String  
 SCHEMA: String

**SQLExceptionEvent**

P: flash.events  
 → Event  
 ANALYZE: String  
 ATTACH: String  
 BEGIN: String  
 CANCEL: String  
 CLOSE: String  
 COMMIT: String  
 COMPACT: String  
 DEANALYZE: String  
 DETACH: String

**SQLIndexSchema**

P: flash.data  
 → SQLSchema  
 table: String  
 SQLIndexSchema()

**SQLMode**

P: flash.data  
 → Object  
 CREATE: String  
 READ: String  
 UPDATE: String

**SQLResult**

P: flash.data  
 → Object  
 complete: Boolean  
 data: Array  
 lastInsertRowID: Number  
 rowsAffected: Number  
 SQLResult()

**SQLSchema**

P: flash.data  
 → Object  
 database: String  
 name: String  
 sql: String  
 SQLSchema()

**SQLSchemaResult**

P: flash.data  
 → Object  
 indices: Array  
 tables: Array  
 triggers: Array  
 views: Array  
 SQLSchemaResult()

**SQLStatement**

P: flash.data  
 → EventDispatcher  
 executing: Boolean  
 itemClass: Class  
 parameters: Object  
 sqlConnection: SQLConnection  
 text: String  
 cancel(): void  
 clearParameters(): void  
 execute(): void  
 getResult(): SQLResult  
 next(): void  
 SQLStatement()

**SQLTableSchema**

P: flash.data  
 → SQLSchema  
 columns: Array  
 SQLTableSchema()

**SQLTransactionLockType**

P: flash.data  
 → Object  
 DEFERRED: String  
 EXCLUSIVE: String  
 IMMEDIATE: String

**SQLTriggerSchema**

P: flash.data  
 → SQLSchema  
 table: String  
 SQLTriggerSchema()

**SQLUpdateEvent**

P: flash.events  
 → Event  
 DELETE: String  
 INSERT: String  
 rowID: Number  
 table: String  
 UPDATE: String  
 clone(): Event  
 SQLUpdateEvent()

**SQLViewSchema**

P: flash.data  
 → SQLTableSchema

**StackedSeries**

P: mx.charts.chartClasses  
 → Series  
 allowNegativeForStacked: Boolean  
 horizontalAxis: IAxis  
 negTotalsByPrimaryAxis: Dictionary  
 posTotalsByPrimaryAxis: Dictionary  
 series: Array  
 stackedMaximum: Number  
 stackedMinimum: Number  
 type: String  
 verticalAxis: IAxis  
 buildSubSeries(): void  
 customizeSeries(): void  
 formatDataTip(): String  
 invalidateSeries(): void  
 invalidateStacking(): void  
 stack(): void  
 StackedSeries()  
 updateStacking(): void

**StackFrame**

P: flash.sampler  
 → Object  
 file: String  
 line: uint  
 name: String  
 toString(): String

**StackOverflowError**

P: flash.errors  
 → Error  
 StackOverflowError(): void

**Stage**

P: flash.display  
 → DisplayObjectContainer  
 align: String  
 cacheAsBitmap: Boolean  
 displayState: String  
 focus: InteractiveObject  
 frameRate: Number  
 fullScreenHeight: uint  
 fullScreenSourceRect: Rectangle  
 fullScreenWidth: uint  
 fullScreenWidth: uint  
 height: Number  
 mouseChildren: Boolean  
 nativeWindow: NativeWindow  
 numChildren: int  
 quality: String  
 scaleMode: String  
 showDefaultContextMenu: Boolean  
 stageFocusRect: Boolean

stageHeight: int  
 stageWidth: int  
 tabChildren: Boolean  
 tabEnabled: Boolean  
 textSnapshot: TextSnapshot  
 width: Number  
 addChild(): DisplayObject  
 addChildAt(): DisplayObject  
 addEventListener(): void  
 assignFocus(): void  
 dispatchEvent(): Boolean  
 hasEventListener(): Boolean  
 invalidate(): void  
 isFocusInaccessible(): Boolean  
 removeChildAt(): DisplayObject  
 setChildIndex(): void  
 swapChildrenAt(): void  
 willTrigger(): Boolean

**StageAlign**

P: flash.display  
 → Object  
 BOTTOM: String  
 BOTTOM\_LEFT: String  
 BOTTOM\_RIGHT: String  
 LEFT: String  
 RIGHT: String  
 TOP: String  
 TOP\_LEFT: String  
 TOP\_RIGHT: String

**StageDisplayState**

P: flash.display  
 → Object  
 FULL\_SCREEN: String  
 FULL\_SCREEN\_INTERACTIVE: String  
 NORMAL: String

**StageQuality**

P: flash.display  
 → Object  
 BEST: String  
 HIGH: String  
 LOW: String  
 MEDIUM: String

**StageScaleMode**

P: flash.display  
 → Object  
 EXACT\_FIT: String  
 NO\_BORDER: String  
 NO\_SCALE: String  
 SHOW\_ALL: String

**State**

P: mx.states  
 → EventDispatcher  
 basedOn: String  
 name: String  
 overrides: Array  
 State(): void

**StateChangeEvent**

P: mx.events  
 → Event  
 CURRENT\_STATE\_CHANGE: String  
 CURRENT\_STATE\_CHANGING: String  
 newState: String  
 oldState: String  
 StateChangeEvent()

**StaticText**

P: flash.text  
 → DisplayObject  
 text: String

**StatusBar**

P: mx.core.windowClasses  
 → UIComponent  
 status: String  
 statusTextField: UITextField  
 StatusBar()

**StatusBarBackgroundSkin**

P: mx.skins.halo  
 → ProgrammaticSkin  
 StatusBarBackgroundSkin()  
 updateDisplayList(): void

**StatusEvent**

P: flash.events  
 → Event  
 code: String  
 level: String  
 STATUS: String  
 clone(): Event  
 StatusEvent()  
 toString(): String

**StreamingAMFChannel**

P: mx.messaging.channels  
 → AMFChannel  
 StreamingAMFChannel()  
 poll(): void

**StreamingConnectionHandler**

P: mx.messaging.channels  
 → EventDispatcher  
 channel: Channel  
 chunkBuffer: ByteArray  
 dataBytesToRead: int  
 dataOffset: int  
 DISCONNECT\_CODE: String = “disconnect”  
 StreamingConnectionHandler()  
 closeStreamingConnection(): void  
 openStreamingConnection(): void

**StreamingHTTPChannel**

P: mx.messaging.channels  
 → HTTPChannel  
 StreamingHTTPChannel()  
 poll(): void

**String**

P: Top Level  
 → Object  
 length: int  
 charAt(): String  
 charCodeAt(): Number  
 concat(): String  
 fromCharCode(): String  
 indexOf(): int  
 lastIndexOf(): int  
 localeCompare(): int  
 match(): Array  
 replace(): String  
 search(): int  
 slice(): String  
 split(): Array  
 String()  
 substr(): String  
 substring(): String  
 toLocaleLowerCase(): String  
 toLocaleUpperCase(): String  
 toLowerCase(): String  
 toUpperCase(): String  
 valueOf(): String

**StringUtil**

P: mx.utils  
 → Object  
 isWhitespace(): Boolean  
 substitute(): String  
 trim(): String  
 trimArrayElements(): String

**StringValidator**

P: mx.validators  
 → Validator  
 maxLength: Object  
 minLength: Object  
 tooLongError: String  
 tooShortError: String  
 doValidation(): Array  
 StringValidator()  
 validateString(): Array

**Stroke**

P: mx.graphics  
 → EventDispatcher  
 IStroke  
 alpha: Number  
 caps: String  
 color: uint  
 joints: String  
 miterLimit: Number  
 pixelHinting: Boolean  
 scaleMode: String  
 weight: Number  
 apply(): void  
 Stroke()

**StyleEvent**

P: mx.events  
 → ProgressEvent  
 COMPLETE: String  
 ERROR: String  
 errorText: String  
 PROGRESS: String  
 StyleEvent()

**StyleManager**

P: mx.styles  
 → Object  
 NOT\_A\_COLOR: uint  
 selectors: Array  
 clearStyleDeclaration(): void  
 getColorName(): uint  
 getColorNames(): void  
 getStyleDeclaration(): CSSStyleDeclaration  
 isColorName(): Boolean  
 isInheritingStyle(): Boolean  
 isInheritingTextFormatStyle(): Boolean  
 isParentDisplayListInvalidatingStyle(): Boolean  
 isParentSizeInvalidatingStyle(): Boolean  
 isSizeInvalidatingStyle(): Boolean  
 isValidStyleValue(): Boolean  
 loadStyleDeclarations(): IEventDispatcher  
 registerColorName(): void  
 registerInheritingStyle(): void  
 registerParentDisplayListInvalidatingStyle(): void  
 registerParentSizeInvalidatingStyle(): void  
 registerSizeInvalidatingStyle(): void  
 setStyleDeclaration(): void  
 unloadStyleDeclarations(): void

**StyleProxy**

P: mx.styles  
 → Object  
 IStyleClient  
 className: String  
 filterMap: Object  
 inheritingStyles: Object  
 nonInheritingStyles: Object  
 source: IStyleClient  
 styleDeclaration: CSSStyleDeclaration  
 styleName: Object  
 clearStyle(): void  
 getClassStyleDeclarations(): Array  
 getStyle(): \*  
 notifyStyleChangeInChildren(): void  
 regenerateStyleCache(): void  
 registerEffects(): void  
 setStyle(): void  
 styleChanged(): void  
 StyleProxy()

**StyleSheet**

P: flash.text  
 → EventDispatcher  
 styleNames: Array  
 clear(): void  
 getStyle(): Object  
 parseCSS(): void  
 setStyle(): void  
 StyleSheet()  
 transform(): TextFormat

**SumAggregator**

P: mx.olap.aggregators  
 → Object  
 IOLAPCustomAggregator  
 computeBegin(): Object  
 computeEnd(): Number  
 computeLoop(): void  
 computeObjectBegin(): Object  
 computeObjectEnd(): Number  
 computeObjectLoop(): void

**SummaryField**

P: mx.collections  
 → Object  
 dataField: String  
 label: String  
 operation: String  
 summaryFunction: Function  
 SummaryField()

**SummaryObject**

P: mx.collections  
 → Object

**SummaryRow**

P: mx.collections  
 → Object  
 fields: Array  
 summaryObjectFunction: Function  
 summaryPlacement: String  
 SummaryRow()

**SubscriptionInfo**

P: mx.messaging  
 → Object  
 selector: String  
 subtopic: String  
 SubscriptionInfo()

**SwatchPanelSkin** ■

P: mx.skins.halo  
→ Border

**SwatchSkin** ■

P: mx.skins.halo  
→ UIComponent

**SWFLoader** ■

P: mx.controls  
→ UIComponent  
autoLoad: Boolean  
bytesLoaded: Number  
bytesTotal: Number  
content: DisplayObject  
contentHeight: Number  
contentWidth: Number  
loaderContext: LoaderContext  
maintainAspectRatio: Boolean  
percentLoaded: Number  
scaleContent: Boolean  
showBusyCursor: Boolean  
source: Object  
trustContent: Boolean  
load(): void  
SWFLoader()

**SWFLoaderAutomationImpl** ■

P: mx.automation.delegates.controls  
→ UIComponentAutomationImpl  
init(): void  
SWFLoaderAutomationImpl()

**SWFVersion**

P: flash.display  
→ Object  
FLASH1: uint  
FLASH2: uint  
FLASH3: uint  
FLASH4: uint  
FLASH5: uint  
FLASH6: uint  
FLASH7: uint  
FLASH8: uint  
FLASH9: uint

**SwitchSymbolFormatter** ■

P: mx.flash  
→ Object  
formatValue(): String  
SwitchSymbolFormatter()

**SyncEvent**

P: flash.events  
→ Event  
changeList: Array  
SYNC: String  
clone(): Event  
SyncEvent()  
toString(): String

**SyntaxError**

P: Top Level  
→ Error  
SyntaxError()

**System**

P: flash.system  
→ Object  
ime: IME  
totalMemory: uint  
useCodePage: Boolean

exit(): void  
gc(): void  
pause(): void  
resume(): void  
setClipboard(): void

**SystemManager** ■

P: mx.managers  
→ MovieClip  
IChildList  
IFlexDisplayObject  
IFlexModuleFactory  
ISystemManager  
application: UIComponent  
cursorChildren: IChildList  
document: Object  
embeddedFontList: Object  
explicitHeight: Number  
explicitWidth: Number  
focusPane: Sprite  
height: Number  
measuredHeight: Number  
measuredWidth: Number  
numChildren: int  
numModalWindows: int  
popUpChildren: IChildList  
preloaderBackgroundAlpha: Number  
preloaderBackgroundColor: uint  
preloaderBackgroundImage: Object  
preloaderBackgroundSize: String  
rawChildren: IChildList  
screen: Rectangle  
toolTipChildren: IChildList  
topLevelSystemManager: ISystemManager  
width: Number  
activate(): void  
addFocusManager(): void  
create(): Object  
deactivate(): void  
getDefinitionByName(): Object  
getExplicitOrMeasuredHeight(): Number  
getExplicitOrMeasuredWidth(): Number  
getSWFRoot(): DisplayObject  
isFontFaceEmbedded(): Boolean  
isTopLevel(): Boolean  
isTopLevelWindow(): Boolean  
move(): void  
removeFocusManager(): void  
setActualSize(): void  
SystemManager()

**SystemTrayIcon** ●

P: flash.desktop  
→ InteractiveIcon  
bitmaps: Array  
height: int  
MAX\_TIP\_LENGTH: Number  
menu: NativeMenu  
tooltip: String  
width: int

**TabBar** ■

P: mx.controls  
→ ToggleButtonBar

**TabNavigator** ■

P: mx.containers  
→ ViewStack  
IFocusManagerComponent  
tabBar: TabBar  
tabBarStyleFilters: Object  
getTabAt(): Button

measure(): void  
TabNavigator()  
updateDisplayList(): void

**TabNavigatorAutomationImpl** ■

P: mx.automation.delegates.containers  
→ ViewStackAutomationImpl  
componentInitialized(): void  
init(): void  
replayAutomatableEvent(): Boolean  
TabNavigatorAutomationImpl()

**TabSkin** ■

P: mx.skins.halo  
→ Border

**Text** ■

P: mx.controls  
→ Label  
commitProperties(): void  
Text()

**TextArea** ■

P: mx.controls  
→ ScrollControlBase  
IDataRenderer  
IDropInListItemRenderer  
IFocusManagerComponent, IIME-Support  
IListItemRenderer  
IFontContextComponent  
condenseWhite: Boolean  
data: Object  
displayAsPassword: Boolean  
editable: Boolean  
horizontalScrollPolicy: String  
horizontalScrollPosition: Number  
htmlText: String  
imeMode: String  
length: int  
listData: BaseListData  
maxChars: int  
restrict: String  
selectable: Boolean  
selectionBeginIndex: int  
selectionEndIndex: int  
styleSheet: StyleSheet  
text: String  
textField: UITextField  
textHeight: Number  
textWidth: Number  
verticalScrollPolicy: String  
verticalScrollPosition: Number  
wordWrap: Boolean  
getLineMetrics(): TextLineMetrics  
setSelection(): void  
TextArea()

**TextAreaAutomationImpl** ■

P: mx.automation.delegates.controls  
→ ScrollControlBaseAutomationImpl  
init(): void  
TextAreaAutomationImpl()

**TextColorType**

P: flash.text  
→ Object  
DARK\_COLOR: String  
LIGHT\_COLOR: String

**TextDisplayMode**

P: flash.text  
→ Object  
CRT: String  
DEFAULT: String  
LCD: String

**TextEvent**

P: flash.events  
→ Event  
LINK: String  
text: String  
TEXT\_INPUT: String  
clone(): Event  
TextEvent()  
toString(): String

**TextField**

→ InteractiveObject  
P: flash.text  
alwaysShowSelection: Boolean  
antiAliasType: String  
autoSize: String  
background: Boolean  
backgroundColor: uint  
border: Boolean  
borderColor: uint  
bottomScrollV: int  
caretIndex: int  
condenseWhite: Boolean  
contextMenu: NativeMenu  
defaultTextFormat: TextFormat  
displayAsPassword: Boolean  
embedFonts: Boolean  
gridFitType: String  
htmlText: String  
length: int  
maxChars: int  
maxScrollH: int  
maxScrollV: int  
mouseWheelEnabled: Boolean  
multiline: Boolean  
numLines: int  
restrict: String  
scrollH: int  
scrollV: int  
selectable: Boolean  
selectionBeginIndex: int  
selectionEndIndex: int  
sharpness: Number  
styleSheet: StyleSheet  
text: String  
textField: UITextField  
textHeight: Number  
textWidth: Number  
thickness: Number  
type: String  
useRichTextClipboard: Boolean  
wordWrap: Boolean  
appendText(): void  
getCharBoundaries(): Rectangle  
getCharIndexAtPoint(): int  
getFirstCharInParagraph(): int  
getImageReference(): DisplayObject  
getLineIndexAtPoint(): int  
getLineIndexOfChar(): int  
getLineLength(): int  
getLineMetrics(): TextLineMetrics  
getLineOffset(): int  
getLineText(): String  
getParagraphLength(): int  
getTextFormat(): TextFormat  
replaceSelectedText(): void  
replaceText(): void

setSelection(): void  
setTextFormat(): void  
TextField()

**TextFieldAsset** ■

P: mx.core  
→ FlexTextField  
IFlexAsset  
IFlexDisplayObject  
measuredHeight: Number  
measuredWidth: Number  
move(): void  
setActualSize(): void  
TextFieldAsset()

**TextFieldAutomationHelper** ■

P: mx.automation.delegates  
→ Object  
TextFieldAutomationHelper()  
recordAutomatableEvent(): void  
replayAutomatableEvent(): void

**TextFieldAutoSize**

P: flash.text  
→ Object  
CENTER: String  
LEFT: String  
NONE: String  
RIGHT: String

**TextFieldType**

P: flash.text  
→ Object  
DYNAMIC: String  
INPUT: String

**TextFormat**

P: flash.text  
→ Object  
align: String  
blockIndent: Object  
bold: Object  
bullet: Object  
color: Object  
font: String  
indent: Object  
italic: Object  
kerning: Object  
leading: Object  
leftMargin: Object  
letterSpacing: Object  
rightMargin: Object  
size: Object  
tabStops: Array  
target: String  
underline: Object  
url: String  
TextFormat()

**TextFormatAlign**

P: flash.text  
→ Object  
CENTER: String  
JUSTIFY: String  
LEFT: String  
RIGHT: String

**TextInput** ■

P: mx.controls  
→ UIComponent  
IDataRenderer  
IDropInListItemRenderer  
IFocusManagerComponent  
IIMESupport

IListItemRenderer  
IFontContextComponent  
condenseWhite: Boolean  
data: Object  
displayAsPassword: Boolean  
editable: Boolean  
fontContext: IFlexModuleFactory  
horizontalScrollPosition: Number  
htmlText: String  
imeMode: String  
length: int  
listData: BaseListData  
maxChars: int  
restrict: String  
selectionBeginIndex: int  
selectionEndIndex: int  
text: String  
textField: UITextField  
textHeight: Number  
textWidth: Number  
createBorder(): void  
getLineMetrics(): TextLineMetrics  
setSelection(): void  
TextInput()

**TextInputAutomationImpl** ■

P: mx.automation.delegates.controls  
→ UIComponentAutomationImpl  
componentInitialized(): void  
init(): void  
TextInputAutomationImpl()

**TextLineMetrics**

P: flash.text  
→ Object  
ascent: Number  
descent: Number  
height: Number  
leading: Number  
width: Number  
x: Number  
TextLineMetrics()

**TextRange** ■

P: mx.controls.textClasses  
→ Object  
beginIndex: int  
bullet: Boolean  
color: Object  
endIndex: int  
fontFamily: String  
fontSize: int  
fontStyle: String  
fontWeight: String  
htmlText: String  
kerning: Boolean  
letterSpacing: Number  
modifiesSelection: Boolean  
owner: UIComponent  
text: String  
textAlign: String  
textDecoration: String  
url: String  
TextRange()

**TextRenderer**

P: flash.text  
→ Object  
displayMode: String  
maxLevel: int  
setAdvancedAntiAliasingTable(): void

**TextSelectionEvent** ■

P: mx.automation.events  
→ Event  
beginIndex: int  
endIndex: int  
TEXT\_SELECTION\_CHANGE: String = "textSelectionChange"  
TextSelectionEvent()

**TextSnapshot**

P: flash.text  
→ Object  
charCount: int  
findText(): int  
getSelected(): Boolean  
getSelectedText(): String  
getText(): String  
getTextRunInfo(): Array  
hitTestTextNearPos(): Number  
setSelectColor(): void  
setSelected(): void

**Tile** ■

P: mx.containers  
→ Container  
direction: String  
tileHeight: Number  
tileWidth: Number  
measure(): void  
Tile()  
updateDisplayList(): void

**TileBase** ●

P: mx.controls.listClasses  
→ ListBase  
direction: String  
maxColumns: int  
maxRows: int  
measuringObjects: Dictionary  
adjustListContent(): void  
createItemRenderer(): IListItemRenderer  
drawTileBackground(): DisplayObject  
drawTileBackgrounds(): void  
makeListData(): BaseListData  
scrollHorizontally(): void  
TileBase()

**TileBaseAutomationImpl** ■

P: mx.automation.delegates.controls  
→ ListBaseAutomationImpl  
automationTabularData: Object  
init(): void  
TileBaseAutomationImpl()

**TileBaseDirection** ■

P: mx.controls.listClasses  
→ Object  
HORIZONTAL: String  
VERTICAL: String

**TileDirection** ■

P: mx.containers  
→ Object  
HORIZONTAL: String  
VERTICAL: String

**TileList** ■

P: mx.controls  
→ TileBase  
selectedIndex: int  
toggleOnClick: Boolean  
ToggleButtonBar(): void  
UIComponent

**TileListItemRenderer** ■

P: mx.controls.listClasses  
→ UIComponent  
data: Object  
icon: IFlexDisplayObject  
label: UITextField  
listData: BaseListData  
TileListItemRenderer()  
toolTipShowHandler(): void

**TileListItemRendererAutomationImpl** ■

P: mx.automation.delegates.controls  
→ UIComponentAutomationImpl  
IDataRenderer  
IDropInListItemRenderer  
IListItemRenderer  
IFontContextComponent  
init(): void  
TileListItemRendererAutomationImpl()

**Timer**

P: flash.utils  
→ EventDispatcher  
currentCount: int  
delay: Number  
repeatCount: int  
running: Boolean  
reset(): void  
start(): void  
stop(): void  
Timer()

**TimerEvent**

P: flash.events  
→ Event  
TIMER: String  
TIMER\_COMPLETE: String  
clone(): Event  
TimerEvent()  
toString(): String  
updateAfterEvent(): void

**TitleBackground** ■

P: mx.skins.halo  
→ ProgrammaticSkin

**TitleBar** ■●

P: mx.core.windowClasses  
→ UIComponent  
closeButton: Button  
maximizeButton: Button  
minimizeButton: Button  
title: String  
titleLabel: Class  
titleLabelField: UITextField  
doubleClickHandler(): void  
placeButtons(): void  
placeTitle(): void  
styleChanged(): void  
TitleBar()

**TitleWindow** ■

P: mx.containers  
→ Panel  
showCloseButton: Boolean  
TitleWindow()

**ToggleButtonBar** ■

P: mx.controls  
→ ButtonBar

**ToggleButtonBarAutomationImpl**

P: mx.automation.delegates.  
controls  
→ ButtonBarAutomationImpl  
[init\(\)](#) : void  
[ToggleButtonBarAutomationImpl\(\)](#)

**ToolTip**

→ UIComponent  
P: mx.controls  
*IToolTip*  
*IFontContextComponent*  
[maxWidth](#): Number  
[text](#): String  
[textField](#): IUITextField  
[ToolTip\(\)](#)

**ToolTipAutomationImpl**

P: mx.automation.delegates.  
controls  
→ UIComponentAutomationImpl  
[init\(\)](#) : void  
[ToolTipAutomationImpl\(\)](#)

**ToolTipBorder**

P: mx.skins.halo  
→ RectangularBorder

**ToolTipEvent**

P: mx.events  
→ Event  
[TOOL\\_TIP\\_CREATE](#): String  
[TOOL\\_TIP\\_END](#): String  
[TOOL\\_TIP\\_HIDE](#): String  
[TOOL\\_TIP\\_SHOW](#): String  
[TOOL\\_TIP\\_SHOWN](#): String  
[TOOL\\_TIP\\_START](#): String  
[tooltip](#): IToolTip  
[ToolTipEvent\(\)](#)

**ToolTipManager**

P: mx.managers  
→ EventDispatcher  
[currentTarget](#): DisplayObject  
[currentToolTip](#): IToolTip  
[enabled](#): Boolean  
[hideDelay](#): Number  
[hideEffect](#): IAbstractEffect  
[scrubDelay](#): Number  
[showDelay](#): Number  
[showEffect](#): IAbstractEffect  
[tooltipClass](#): Class  
[createToolTip\(\)](#) : IToolTip  
[destroyToolTip\(\)](#) : void

**TraceTarget**

P: mx.logging.targets  
→ LineFormattedTarget

**Transform**

P: flash.geom  
→ Object  
[colorTransform](#): ColorTransform  
[concatenatedColorTransform](#): ColorTransform  
[concatenatedMatrix](#): Matrix  
[matrix](#): Matrix  
[pixelBounds](#): Rectangle

**Transition**

P: mx.states  
→ Object  
[effect](#): IEffect  
[fromState](#): String  
[toState](#): String  
[Transition\(\)](#) : void

**Tree**

P: mx.controls  
→ List  
*IIMESupport*  
[dataDescriptor](#): ITreeDataDescriptor  
[dataProvider](#): Object  
[dragMoveEnabled](#): Boolean  
[firstVisibleItem](#): Object  
[hasRoot](#): Boolean  
[itemIcons](#): Object  
[maxHorizontalScrollPosition](#): Number  
[openItems](#): Object  
[showRoot](#): Boolean  
[dragCompleteHandler\(\)](#) : void  
[dragDropHandler\(\)](#) : void  
[expandChildrenOf\(\)](#) : void  
[expandItem\(\)](#) : void  
[getParentItem\(\)](#) : \*  
[initListData\(\)](#) : void  
[isItemOpen\(\)](#) : Boolean  
[makeListData\(\)](#) : BaseListData  
[setItemIcon\(\)](#) : void  
[Tree\(\)](#)

**TreeAutomationImpl**

P: mx.automation.delegates.  
controls  
→ ListAutomationImpl  
[automationTabularData](#): Object  
[init\(\)](#) : void  
[TreeAutomationImpl\(\)](#)

**TreeEvent**

P: mx.events  
→ Event  
[animate](#): Boolean  
[dispatchEvent](#): Boolean  
[item](#): Object  
[ITEM\\_CLOSE](#): String  
[ITEM\\_OPEN](#): String  
[ITEM\\_OPENING](#): String  
[itemRenderer](#): IListItemRenderer  
[opening](#): Boolean  
[triggerEvent](#): Event  
[TreeEvent\(\)](#)

**TreeItemRenderer**

P: mx.controls.treeClasses  
→ UIComponent  
[data](#): Object  
[disclosureIcon](#): IFlexDisplayObject  
[icon](#): IFlexDisplayObject  
[label](#): IUITextField  
[listData](#): BaseListData  
[TreeItemRenderer\(\)](#)

**TreeItemRendererAutomationImpl**

P: mx.automation.delegates.  
controls  
→ UIComponentAutomationImpl  
*IDataRenderer*  
*IDropInListItemRenderer*  
*IListItemRenderer*  
*IFontContextComponent*  
[init\(\)](#) : void  
[TreeItemRendererAutomationImpl\(\)](#)

**TreeListData**

P: mx.controls.treeClasses  
→ BaseListData  
[depth](#): int  
[disclosureIcon](#): Class  
[hasChildren](#): Boolean  
[icon](#): Class  
[indent](#): int  
[item](#): Object  
[open](#): Boolean  
[TreeListData\(\)](#)

**TriangleItemRenderer**

P: mx.charts.renderers  
→ ProgrammaticSkin  
*IDataRenderer*  
[data](#): Object  
[TriangleItemRenderer\(\)](#)

**Tween**

P: mx.effects  
→ EventDispatcher  
[duration](#): Number  
[easingFunction](#): Function  
[listener](#): Object  
[endTween\(\)](#) : void  
[pause\(\)](#) : void  
[resume\(\)](#) : void  
[reverse\(\)](#) : void  
[seek\(\)](#) : void  
[setTweenHandlers\(\)](#) : void  
[stop\(\)](#) : void  
[Tween\(\)](#)

**TweenEffect**

P: mx.effects  
→ Effect  
[easingFunction](#): Function  
[TweenEffect\(\)](#)  
[tweenEventHandler\(\)](#) : void

**TweenEffectInstance**

P: mx.effects.effectClasses  
→ EffectInstance  
[easingFunction](#): Function  
[playheadTime](#): Number  
[tween](#): Tween  
[createTween\(\)](#) : Tween  
[end\(\)](#) : void  
[onTweenEnd\(\)](#) : void  
[onTweenUpdate\(\)](#) : void  
[seek\(\)](#) : void  
[TweenEffectInstance\(\)](#)

**TweenEvent**

P: mx.events  
→ Event  
[TWEEN\\_END](#): String  
[TWEEN\\_START](#): String  
[TWEEN\\_UPDATE](#): String  
[TweenEvent\(\)](#)

**TypeError**

P: Top Level  
→ Error  
[TypeError\(\)](#)

**UIComponent**

P: mx.core  
→ FlexSprite  
*IAutomationObject*  
*IChildList*  
*IDeferredInstantiationUIComponent*  
*IFlexDisplayObject*  
*IFlexModule*

*Invalidating*

*ILayoutManagerClient*  
*IPropertyChangeNotifier*  
*IRepeaterClient*  
*ISimpleStyleClient*  
*IStyleClient*  
*IToolTipManagerClient*  
*UIComponent*  
*IValidatorListener*  
*IStateClient*  
*IConstraintClient*  
[activeEffects](#): Array  
[automationDelegate](#): Object  
[automationName](#): String  
[automationTabularData](#): Object  
[automationValue](#): Array  
[baselinePosition](#): Number  
[cacheHeuristic](#): Boolean  
[cachePolicy](#): String  
[className](#): String  
[contentMouseX](#): Number  
[contentMouseY](#): Number  
[currentState](#): String  
[cursorManager](#): \*  
[DEFAULT\\_MAX\\_HEIGHT](#): Number  
[DEFAULT\\_MAX\\_WIDTH](#): Number  
[DEFAULT\\_MEASURED\\_HEIGHT](#): Number  
[DEFAULT\\_MEASURED\\_MIN\\_HEIGHT](#): Number  
[DEFAULT\\_MEASURED\\_MIN\\_WIDTH](#): Number  
[DEFAULT\\_MEASURED\\_WIDTH](#): Number  
[descriptor](#): UIComponentDescriptor  
[document](#): Object  
[doubleClickEnabled](#): Boolean  
[enabled](#): Boolean  
[errorString](#): String  
[explicitHeight](#): Number  
[explicitMaxHeight](#): Number  
[explicitMaxWidth](#): Number  
[explicitMinHeight](#): Number  
[explicitMinWidth](#): Number  
[explicitWidth](#): Number  
[flexContextMenu](#): IFlexContextMenu  
[focusEnabled](#): Boolean  
[focusManager](#): IFocusManager  
[focusPane](#): Sprite  
[height](#): Number  
[id](#): String  
[includeInLayout](#): Boolean  
[inheritingStyles](#): Object  
[initialized](#): Boolean  
[instanceIndex](#): int  
[instanceIndices](#): Array  
[isDocument](#): Boolean  
[isPopUp](#): Boolean  
[maxHeight](#): Number  
[maxWidth](#): Number  
[measuredHeight](#): Number  
[measuredMinHeight](#): Number  
[measuredMinWidth](#): Number  
[measuredWidth](#): Number  
[minHeight](#): Number  
[minWidth](#): Number  
[moduleFactory](#): IFlexModuleFactory  
[mouseFocusEnabled](#): Boolean  
[nestLevel](#): int  
[nonInheritingStyles](#): Object  
[numAutomationChildren](#): int  
[owner](#): DisplayObjectContainer  
[parent](#): DisplayObjectContainer  
[parentApplication](#): Object  
[parentDocument](#): Object  
[percentHeight](#): Number  
[percentWidth](#): Number  
[processedDescriptors](#): Boolean  
[repeater](#): IRepeater  
[repeaterIndex](#): int  
[repeaterIndices](#): Array  
[repeaters](#): Array  
[resourceManager](#): IResourceManager  
[scaleX](#): Number  
[scaleY](#): Number  
[screen](#): Rectangle  
[showInAutomationHierarchy](#): Boolean  
[states](#): Array  
[styleDeclaration](#): CSSStyleDeclaration  
[styleName](#): Object  
[systemManager](#): ISystemManager  
[toolTip](#): String  
[transitions](#): Array  
[tweeningProperties](#): Array  
[uid](#): String  
[unscaledHeight](#): Number  
[unscaledWidth](#): Number  
[updateCompletePendingFlag](#): Boolean  
[validationSubField](#): String  
[visible](#): Boolean  
[width](#): Number  
[x](#): Number  
[y](#): Number  
[adjustFocusRect\(\)](#) : void  
[attachOverlay\(\)](#) : void  
[callLater\(\)](#) : void  
[childrenCreated\(\)](#) : void  
[clearStyle\(\)](#) : void  
[commitProperties\(\)](#) : void  
[contentToGlobal\(\)](#) : Point  
[contentToLocal\(\)](#) : Point  
[createAutomationIDPart\(\)](#) : Object  
[createChildren\(\)](#) : void  
[createInFontContext\(\)](#) : Object  
[createInModuleContext\(\)](#) : Object  
[createReferenceOnParentDocument\(\)](#) : void  
[deleteReferenceOnParentDocument\(\)](#) : void  
[determineTextFormatFromStyles\(\)](#) : UITextFormat  
[dispatchEvent\(\)](#) : Boolean  
[drawFocus\(\)](#) : void  
[drawRoundRect\(\)](#) : void  
[effectFinished\(\)](#) : void  
[effectStarted\(\)](#) : void  
[endEffectsStarted\(\)](#) : void  
[executeBindings\(\)](#) : void  
[finishPrint\(\)](#) : void  
[focusInHandler\(\)](#) : void  
[focusOutHandler\(\)](#) : void  
[getAutomationChildAt\(\)](#) : IAutomationObject  
[getClassStyleDeclarations\(\)](#) : Array  
[getConstraintValue\(\)](#) : \*  
[getExplicitOrMeasuredHeight\(\)](#) : Number  
[getExplicitOrMeasuredWidth\(\)](#) : Number  
[getFocus\(\)](#) : InteractiveObject  
[getRepeaterItem\(\)](#) : Object  
[getStyle\(\)](#) : \*  
[globalToContent\(\)](#) : Point  
[horizontalGradientMatrix\(\)](#) : Matrix  
[initializationComplete\(\)](#) : void  
[initialize\(\)](#) : void  
[initializeAccessibility\(\)](#) : void  
[initializeRepeaterArrays\(\)](#) : void

[invalidateDisplayList\(\)](#) : void  
[invalidateProperties\(\)](#) : void  
[invalidateSize\(\)](#) : void  
[isOurFocus\(\)](#) : Boolean  
[keyDownHandler\(\)](#) : void  
[keyUpHandler\(\)](#) : void  
[localToContent\(\)](#) : Point  
[measure\(\)](#) : void  
[measureHTMLText\(\)](#) : TextLineMetrics  
[measureText\(\)](#) : TextLineMetrics  
[move\(\)](#) : void  
[notifyStyleChangeInChildren\(\)](#) : void  
[owns\(\)](#) : Boolean  
[parentChanged\(\)](#) : void  
[prepareToPrint\(\)](#) : Object  
[regenerateStyleCache\(\)](#) : void  
[registerEffects\(\)](#) : void  
[replayAutomatableEvent\(\)](#) : Boolean  
[resolveAutomationIDPart\(\)](#) : Array  
[resourcesChanged\(\)](#) : void  
[resumeBackgroundProcessing\(\)](#) : void  
[setActualSize\(\)](#) : void  
[setConstraintValue\(\)](#) : void  
[setCurrentState\(\)](#) : void  
[setFocus\(\)](#) : void  
[setStyle\(\)](#) : void  
[setVisible\(\)](#) : void  
[styleChanged\(\)](#) : void  
[stylesInitialized\(\)](#) : void  
[suspendBackgroundProcessing\(\)](#) : void  
*UIComponent()*  
[updateDisplayList\(\)](#) : void  
[validateDisplayList\(\)](#) : void  
[validateNow\(\)](#) : void  
[validateProperties\(\)](#) : void  
[validateSize\(\)](#) : void  
[validationResultHandler\(\)](#) : void  
[verticalGradientMatrix\(\)](#) : Matrix

**UIComponentAutomationImpl**

P: mx.automation.delegates.core  
→ EventDispatcher  
*IAutomationObject*  
[automationName](#): String  
[automationValue](#): Array  
*uiAutomationObject*: IAutomationObject  
[uiComponent](#): DisplayObject  
[addLayoutCompleteSynchronization\(\)](#) : void  
[componentInitialized\(\)](#) : void  
[init\(\)](#) : void  
[UIComponentAutomationImpl\(\)](#)

**UIComponentCachePolicy**

P: mx.core  
→ Object  
[AUTO](#): String  
[OFF](#): String  
[ON](#): String

**UIComponentDescriptor**

P: mx.core  
→ ComponentDescriptor  
[effects](#): Array  
[stylesFactory](#): Function  
*UIComponentDescriptor()*

**UIDUtil**

P: mx.utils  
→ Object  
[createUID\(\)](#) : String  
[fromByteArray\(\)](#) : String  
[getUID\(\)](#) : String

[isUID\(\)](#) : Boolean  
[toByteArray\(\)](#) : ByteArray

**UIMovieClip**

P: mx.flash  
→ MovieClip  
*IDeferredInstantiationUIComponent*  
*IToolTipManagerClient*  
*IStateClient*  
*IFocusManagerComponent*  
*IConstraintClient*  
*IAutomationObject*  
[automationDelegate](#): Object  
[automationName](#): String  
[automationTabularData](#): Object  
[automationValue](#): Array  
[baseline](#): \*  
[baselinePosition](#): Number  
[bottom](#): \*  
[boundingBoxName](#): String  
[bounds](#): Rectangle  
[setActualSize\(\)](#) : void  
[cacheHeuristic](#): Boolean  
[cachePolicy](#): String  
[currentState](#): String  
[descriptor](#): UIComponentDescriptor  
[document](#): Object  
[explicitHeight](#): Number  
[explicitMaxHeight](#): Number  
[explicitMaxWidth](#): Number  
[explicitMinHeight](#): Number  
[explicitMinWidth](#): Number  
[explicitWidth](#): Number  
[focusEnabled](#): Boolean  
[focusPane](#): Sprite  
[height](#): Number  
[horizontalCenter](#): \*

[id](#): String  
[includeInLayout](#): Boolean  
[initialized](#): Boolean  
[isPopUp](#): Boolean  
[left](#): \*  
[maxHeight](#): Number  
[maxWidth](#): Number  
[measuredHeight](#): Number  
[measuredMinHeight](#): Number  
[measuredMinWidth](#): Number  
[measuredWidth](#): Number  
[minHeight](#): Number  
[minWidth](#): Number  
[mouseFocusEnabled](#): Boolean  
[numAutomationChildren](#): int  
[owner](#): DisplayObjectContainer  
[parentDocument](#): Object  
[percentHeight](#): Number  
[percentWidth](#): Number  
[right](#): \*  
[showInAutomationHierarchy](#): Boolean  
[systemManager](#): ISystemManager  
[toolTip](#): String  
[top](#): \*  
[tweeningProperties](#): Array  
[verticalCenter](#): \*  
[width](#): Number  
[createAutomationIDPart\(\)](#) : Object  
[createReferenceOnParentDocument\(\)](#) : void  
[deleteReferenceOnParentDocument\(\)](#) : void  
[drawFocus\(\)](#) : void  
[enterFrameHandler\(\)](#) : void  
[executeBindings\(\)](#) : void  
[findFocusCandidates\(\)](#) : void  
[focusInHandler\(\)](#) : void

[isDragEventPositionBased\(\)](#): Boolean  
  
**uint**  
P: Top Level  
→ Object  
[MAX\\_VALUE](#): uint  
[MIN\\_VALUE](#): uint  
toExponential(): String  
toFixed(): String  
toPrecision(): String  
toString(): String  
uint()  
valueOf(): uint

**UITextField**

P: mx.core  
→ FlexTextField  
*IAutomationObject*  
*IIMESupport*  
*IFlexModule*  
*IInvalidating*  
*ISimpleStyleClient*  
*IToolTipManagerClient*  
*IUITextField*  
[automationDelegate](#): Object  
[automationName](#): String  
[automationValue](#): Array  
[baselinePosition](#): Number  
[className](#): String  
[document](#): Object  
[enabled](#): Boolean  
[explicitHeight](#): Number  
[explicitMinHeight](#): Number  
[explicitMaxHeight](#): Number  
[explicitMaxWidth](#): Number  
[explicitMinWidth](#): Number  
[explicitWidth](#): Number  
[focusPane](#): Sprite  
[ignorePadding](#): Boolean  
[imeMode](#): String  
[includeInLayout](#): Boolean  
[inheritingStyles](#): Object



initialized: Boolean  
 isPopUp: Boolean  
 maxHeight: Number  
 maxWidth: Number  
 measuredHeight: Number  
 measuredMinHeight: Number  
 measuredMinWidth: Number  
 measuredWidth: Number  
 minHeight: Number  
 minWidth: Number  
 moduleFactory: IFlexModuleFactory  
 nestLevel: int  
 nonInheritingStyles: Object  
 nonZeroTextHeight: Number  
 owner: DisplayObjectContainer  
 parent: DisplayObjectContainer  
 percentHeight: Number  
 percentWidth: Number  
 processedDescriptors: Boolean  
 styleName: Object  
 systemManager: ISystemManager  
 toolTip: String  
 tweeningProperties: Array  
 updateCompletePendingFlag: Boolean  
 getExplicitOrMeasuredHeight(): Number  
 getExplicitOrMeasuredWidth(): Number  
 getStyle(): \*  
 getStyles(): TextFormat  
 getUITextFormat(): UITextFormat  
 initialize(): void  
 invalidateDisplayList(): void  
 invalidateProperties(): void  
 invalidateSize(): void  
 move(): void  
 owns(): Boolean  
 parentChanged(): void  
 replayAutomatableEvent(): Boolean  
 setActualSize(): void  
 setColor(): void  
 setFocus(): void  
 setStyle(): void  
 setVisible(): void  
 styleChanged(): void  
 truncateToFit(): Boolean  
 UITextField()  
 validateNow(): void

**UITextFieldAutomationImpl** ■

P: mx.automation.delegates.core  
 → Object  
*IAutomationObject*  
 automationValue: Array  
 init(): void  
 UITextFieldAutomationImpl()

**UITextFormat** ■

P: mx.core  
 → TextFormat  
 antiAliasType: String  
 gridFitType: String  
 moduleFactory: IFlexModuleFactory  
 sharpness: Number  
 thickness: Number  
 measureHTMLText(): TextLineMetrics  
 measureText(): TextLineMetrics  
 UITextFormat()

**UnconstrainItemAction** ■

P: mx.effects  
 → Effect

**UnconstrainItemActionInstance** ■

P: mx.effects.effectClasses  
 → ActionEffectInstance  
 effectHost: ListBase  
 UnconstrainItemActionInstance()

**Updater** ●

P: flash.desktop  
 → Object  
 update(): void  
 Updater()

**URLError**

P: Top Level  
 → Error  
 URLError()

**URLLoader**

P: flash.net  
 → EventDispatcher  
 bytesLoaded: uint  
 bytesTotal: uint  
 data:  
 dataFormat: String  
 close(): void  
 load(): void  
 URLLoader()

**URLLoaderDataFormat**

P: flash.net  
 → Object  
 BINARY: String  
 TEXT: String  
 VARIABLES: String

**URLMonitor** ●

P: air.net  
 → ServiceMonitor  
 acceptableStatuses: Array  
 urlRequest: URLRequest  
 checkStatus(): void  
 toString(): String  
 URLMonitor()

**URLRequest**

P: flash.net  
 → Object  
 authenticate: Boolean  
 cacheResponse: Boolean  
 contentType: String  
 data: Object  
 digest: String  
 followRedirects: Boolean  
 manageCookies: Boolean  
 method: String  
 requestHeaders: Array  
 url: String  
 useCache: Boolean  
 userAgent: String  
 URLRequest()

**URLRequestDefaults** ●

P: flash.net  
 → Object  
 authenticate: Boolean  
 cacheResponse: Boolean  
 followRedirects: Boolean  
 manageCookies: Boolean  
 useCache: Boolean  
 userAgent: String  
 setLoginCredentialsForHost(): \*

**URLRequestHeader**

P: flash.net  
 → Object  
 name: String  
 value: String  
 URLRequestHeader()

**URLRequestMethod**

P: flash.net  
 → Object  
 DELETE: String  
 GET: String  
 HEAD: String  
 OPTIONS: String  
 POST: String  
 PUT: String

**URLStream**

P: flash.net  
 → EventDispatcher  
*IDataInput*  
 bytesAvailable: uint  
 connected: Boolean  
 endian: String  
 objectEncoding: uint  
 close(): void  
 load(): void  
 readBoolean(): Boolean  
 readByte(): int  
 readBytes(): void  
 readDouble(): Number  
 readFloat(): Number  
 readInt(): int  
 readMultiByte(): String  
 readObject(): \*  
 readShort(): int  
 readUnsignedByte(): uint  
 readUnsignedInt(): uint  
 readUnsignedShort(): uint  
 readUTF(): String  
 readUTFBytes(): String

**URLUtil** ■

P: mx.utils  
 → Object  
 SERVER\_NAME\_TOKEN: String  
 SERVER\_PORT\_TOKEN: String  
 getFullURL(): String  
 getPort(): uint  
 getProtocol(): String  
 getServerName(): String  
 getServerNameWithPort(): String  
 hasUnresolvableTokens(): Boolean  
 isHTTPSURL(): Boolean  
 isHTTPURL(): Boolean  
 objectToString(): String  
 replacePort(): String  
 replaceProtocol(): String  
 replaceTokens(): String  
 stringToObject(): Object  
 urisEqual(): Boolean

**URLVariables**

P: flash.net  
 → Object  
 decode(): void  
 toString(): String  
 URLVariables(): void

**ValidationResult** ■

P: mx.validators  
 → Object  
 errorCode: String  
 errorMessage: String

isError: Boolean  
 subField: String  
 ValidationResult(): void

**ValidationResultEvent** ■

P: mx.events  
 → Event  
 field: String  
 INVALID: String  
 message: String  
 results: Array  
 VALID: String  
 ValidationResultEvent(): void

**Validator** ■

P: mx.validators  
 → EventDispatcher  
*IMXMLObject*  
 actualListeners: Array  
 actualTrigger: IEventDispatcher  
 DECIMAL\_DIGITS: String  
 enabled: Boolean  
 listener: Object  
 property: String  
 required: Boolean  
 requiredFieldError: String  
 resourceManager: IResourceManager  
 ROMAN\_LETTERS: String  
 source: Object  
 subFields: Array  
 trigger: IEventDispatcher  
 triggerEvent: String  
 addListenerHandler(): void  
 doValidation(): Array  
 getValueFromSource(): Object  
 handleResults(): ValidationResultEvent  
 initialized(): void  
 isRealValue(): Boolean  
 removeListenerHandler(): void  
 resourcesChanged(): void  
 validate(): ValidationResultEvent  
 validateAll(): Array  
 Validator(): void

**VBox** ■

P: mx.containers  
 → Box

**VDividedBox** ■

P: mx.containers  
 → DividedBox

**VerifyError**

P: Top Level  
 → Error  
 VerifyError(): void

**Video**

P: flash.media  
 → EventDispatcher  
 deblocking: int  
 smoothing: Boolean  
 videoHeight: int  
 videoWidth: int  
 attachCamera(): void  
 attachNetStream(): void  
 clear(): void  
 Video(): void

**VideoDisplay** ■

P: mx.controls  
 → UIComponent  
 autoBandWidthDetection: Boolean  
 autoPlay: Boolean

autoRewind: Boolean  
 border: IFlexDisplayObject  
 borderMetrics: EdgeMetrics  
 bufferTime: Number  
 bytesLoaded: int  
 bytesTotal: int  
 cuePointManager: Object  
 cuePointManagerClass: Class  
 cuePoints: Array  
 idleTimeout: int  
 live: Boolean  
 maintainAspectRatio: Boolean  
 metadata: Object  
 playheadTime: Number  
 playheadUpdateInterval: int  
 playing: Boolean  
 progressInterval: int  
 source: String  
 state: String  
 stateResponsive: Boolean  
 totalTime: Number  
 videoHeight: int  
 videoWidth: int  
 volume: Number  
 attachCamera(): void  
 close(): void  
 createBorder(): void  
 layoutChrome(): void  
 load(): void  
 pause(): void  
 play(): void  
 stop(): void  
 VideoDisplay()

**VideoDisplayAutomationImpl** ■

P: mx.automation.delegates.  
 controls  
 → UIComponentAutomationImpl  
 init(): void  
 VideoDisplayAutomationImpl()

**VideoError** ■

P: mx.controls.videoClasses  
 → Error  
 code: uint  
 DELETE\_DEFAULT\_PLAYER: uint  
 ILLEGAL\_CUE\_POINT: uint  
 INVALID\_CONTENT\_PATH: uint  
 INVALID\_SEEK: uint  
 INVALID\_XML: uint  
 NO\_BITRATE\_MATCH: uint  
 NO\_CONNECTION: uint  
 NO\_CUE\_POINT\_MATCH: uint  
 VideoError()

**VideoEvent** ■

P: mx.events  
 → Event  
 BUFFERING: String  
 CLOSE: String  
 COMPLETE: String  
 CONNECTION\_ERROR: String  
 DISCONNECTED: String  
 EXEC\_QUEUE\_CMD: String  
 LOADING: String  
 PAUSED: String  
 PLAYHEAD\_UPDATE: String  
 PLAYING: String  
 READY: String  
 RESIZING: String  
 REWIND: String  
 REWINDING: String  
 SEEKING: String

state: String  
 STATE\_CHANGE: String  
 stateResponsive: Boolean  
 STOPPED: String  
 VideoEvent()

**ViewStack** ■

P: mx.containers  
 → Container  
*IHistoryManagerClient*  
 contentHeight: Number  
 contentWidth: Number  
 contentX: Number  
 contentY: Number  
 historyManagementEnabled: Boolean  
 resizeToContent: Boolean  
 selectedChild: Container  
 selectedIndex: int  
 commitSelectedIndex(): void  
 loadState(): void  
 measure(): void  
 saveState(): void  
 updateDisplayList(): void  
 ViewStack()

**ViewStackAutomationImpl** ■

P: mx.automation.delegates.  
 containers  
 → ContainerAutomationImpl  
 init(): void  
 ViewStackAutomationImpl()

**VRule** ■

P: mx.controls  
 → UIComponent

**VScrollBar** ■

P: mx.controls  
 → ScrollBar

**VSlider** ■

P: mx.controls  
 → Slider

**WebService** ■

P: mx.rpc.soap  
 → AbstractWebService  
 DEFAULT\_DESTINATION\_HTTP: String  
 DEFAULT\_DESTINATION\_HTTPS: String  
 wsdl: String  
 canLoadWSDL(): Boolean  
 getOperation(): AbstractOperation  
 initializeOperation(): void  
 loadWSDL(): void  
 toString(): String  
 WebService()

**WebService** ■

P: mx.rpc.soap.mxml  
 → WebService  
*IMXMLSupport*  
*IMXMLObject*  
 concurrency: String  
 serviceName: String  
 showBusyCursor: Boolean  
 getOperation(): AbstractOperation  
 initialized(): void  
 WebService(): void

**WedgItemRenderer** ■

P: mx.charts.renderers  
 → ProgrammaticSkin  
*IDataRenderer*  
 data: Object  
 WedgItemRenderer()

**Window** ■●

P: mx.core  
 → LayoutContainer  
*IWindow*  
 alwaysInFront: Boolean  
 closed: Boolean  
 controlBar: UIComponent  
 cursorManager:  
 maxHeight: Number  
 maximizable: Boolean  
 maxWidth: Number  
 menu: FlexNativeMenu  
 minHeight: Number  
 minimizable: Boolean  
 minWidth: Number  
 nativeWindow: NativeWindow  
 resizable: Boolean  
 showGripper: Boolean  
 showStatusBar: Boolean  
 showTitleBar: Boolean  
 status: String  
 statusBar: UIComponent  
 statusBarFactory: IFactory  
 statusBarStyleFilters: Object  
 systemChrome: String  
 systemTrayIconMenu: FlexNativeMenu  
 title: String  
 titleBar: UIComponent  
 titleBarFactory: IFactory  
 titleBarStyleFilters: Object  
 titleIcon: Class  
 transparent: Boolean  
 type: String  
 activate(): void  
 close(): void  
 exit(): void  
 maximize(): void  
 minimize(): void  
 mouseDownHandler(): void  
 orderInBackOf(): Boolean  
 orderInFrontOf(): Boolean  
 orderToBack(): Boolean  
 orderToFront(): Boolean  
 restore(): void  
 WindowedApplication()

**WindowedSystemManager** ■●

P: mx.managers  
 → MovieClip  
*ISystemManager*

cursorChildren: IChildList  
 document: Object  
 embeddedFontList: Object  
 focusPane: Sprite  
 height: Number  
 numModalWindows: int  
 popUpChildren: IChildList  
 rawChildren: IChildList  
 screen: Rectangle  
 toolTipChildren: IChildList  
 topLevelSystemManager: ISystem-  
 Manager  
 width: Number  
 activate(): void  
 addFocusManager(): void  
 create(): Object  
 deactivate(): void  
 getDefinitionByName(): Object  
 isFontFaceEmbedded(): Boolean  
 isTopLevel(): Boolean  
 isTopLevelWindow(): Boolean  
 removeFocusManager(): void  
 WindowedSystemManager(): void

**WindowBackground** ■●

P: mx.skins.halo  
 → ProgrammaticSkin

**WindowCloseButtonSkin** ■●

P: mx.skins.halo  
 → UIComponent

**WindowedApplication** ■●

P: mx.core  
 → Application  
*IWindow*  
 alwaysInFront: Boolean  
 applicationID: String  
 autoExit: Boolean  
 closed: Boolean  
 dockIconMenu: FlexNativeMenu  
 maxHeight: Number  
 maximizable: Boolean  
 maxWidth: Number

menu: FlexNativeMenu  
 minHeight: Number  
 minimizable: Boolean  
 minWidth: Number  
 nativeApplication: NativeApplication  
 nativeWindow: NativeWindow  
 resizable: Boolean  
 showGripper: Boolean  
 showStatusBar: Boolean  
 showTitleBar: Boolean  
 status: String  
 statusBar: UIComponent  
 statusBarFactory: IFactory  
 statusBarStyleFilters: Object  
 systemChrome: String  
 systemTrayIconMenu: FlexNativeMenu  
 title: String  
 titleBar: UIComponent  
 titleBarFactory: IFactory  
 titleBarStyleFilters: Object  
 titleIcon: Class  
 transparent: Boolean  
 type: String  
 activate(): void  
 close(): void  
 exit(): void  
 maximize(): void  
 minimize(): void  
 mouseDownHandler(): void  
 orderInBackOf(): Boolean  
 orderInFrontOf(): Boolean  
 orderToBack(): Boolean  
 orderToFront(): Boolean  
 restore(): void  
 WindowedApplication()

**WindowedSystemManager** ■●

P: mx.managers  
 → MovieClip  
*ISystemManager*

cursorChildren: IChildList  
 document: Object  
 embeddedFontList: Object  
 focusPane: Sprite  
 height: Number  
 numModalWindows: int  
 popUpChildren: IChildList  
 rawChildren: IChildList  
 screen: Rectangle  
 toolTipChildren: IChildList  
 topLevelSystemManager: ISystem-  
 Manager  
 width: Number  
 activate(): void  
 addFocusManager(): void  
 create(): Object  
 deactivate(): void  
 getDefinitionByName(): Object  
 isFontFaceEmbedded(): Boolean  
 isTopLevel(): Boolean  
 isTopLevelWindow(): Boolean  
 removeFocusManager(): void  
 WindowedSystemManager(): void

**WindowMaximizeButtonSkin** ■●

P: mx.skins.halo  
 → UIComponent

**WindowMinimizeButtonSkin** ■●

P: mx.skins.halo  
 → UIComponent

**WindowRestoreButtonSkin** ■●  
P: mx.skins.halo  
→ UIComponent

**WipeDown** ■  
P: mx.effects  
→ MaskEffect

**WipeDownInstance** ■  
P: mx.effects.effectClasses  
→ MaskEffectInstance

**WipeLeft** ■  
P: mx.effects  
→ MaskEffect

**WipeLeftInstance** ■  
P: mx.effects.effectClasses  
→ MaskEffectInstance

**WipeRight** ■  
P: mx.effects  
→ MaskEffect

**WipeRightInstance** ■  
P: mx.effects.effectClasses  
→ MaskEffectInstance

**WipeUp** ■  
P: mx.effects  
→ MaskEffect

**WipeUpInstance** ■  
P: mx.effects.effectClasses  
→ MaskEffectInstance

**WSDLBinding** ■  
P: mx.rpc.wsdl  
→ Object  
name: String  
portType: String  
style: String  
transport: String  
WSDLBinding()

**XML**  
P: Top Level  
→ Object  
ignoreComments: Boolean  
ignoreProcessingInstructions: Boolean  
ignoreWhitespace: Boolean  
prettyIndent: int  
prettyPrinting: Boolean  
addNamespace(): XML  
appendChild(): XML  
attribute(): XMLList  
attributes(): XMLList  
child(): XMLList  
childIndex(): int  
children(): XMLList  
comments(): XMLList  
contains(): Boolean  
copy(): XML  
defaultSettings(): Object  
descendants(): XMLList  
elements(): XMLList  
hasComplexContent(): Boolean  
hasOwnProperty(): Boolean  
hasSimpleContent(): Boolean  
inScopeNamespaces(): Array  
insertChildAfter(): \*  
insertChildBefore(): \*  
length(): int  
localName(): Object

name(): Object  
namespace(): \*  
namespaceDeclarations(): Array  
nodeKind(): String  
normalize(): XML  
parent(): \*  
prependChild(): XML  
processingInstructions(): XMLList  
propertyIsEnumerable(): Boolean  
removeNamespace(): XML  
replace(): XML  
setChildren(): XML  
setLocalName(): void  
setName(): void  
setNamespace(): void  
setSettings(): void  
settings(): Object  
text(): XMLList  
toString(): String  
toXMLString(): String  
valueOf(): XML  
XML()

**XMLDocument**  
P: flash.xml  
→ Object  
docTypeDecl: Object  
idMap: Object  
ignoreWhite: Boolean  
xmlDecl: Object  
createElement(): XMLNode  
createTextNode(): XMLNode  
parseXML(): void  
toString(): String  
XMLDocument()

**XMLList**  
P: Top Level  
→ Object  
attribute(): XMLList  
attributes(): XMLList  
child(): XMLList  
children(): XMLList  
comments(): XMLList  
contains(): Boolean  
copy(): XMLList  
descendants(): XMLList  
elements(): XMLList  
hasComplexContent(): Boolean  
hasOwnProperty(): Boolean  
hasSimpleContent(): Boolean  
length(): int  
normalize(): XMLList  
parent(): Object  
processingInstructions(): XMLList  
propertyIsEnumerable(): Boolean  
text(): XMLList  
toString(): String  
toXMLString(): String  
valueOf(): XMLList  
XMLList()

**XMLListCollection** ■  
P: mx.collections  
→ ListCollectionView  
source: XMLList  
attribute(): XMLList  
attributes(): XMLList  
child(): XMLList  
children(): XMLList  
copy(): XMLList  
descendants(): XMLList  
elements(): XMLList  
text(): XMLList

toString(): String  
toXMLString(): String  
XMLListCollection()

**XMLNode**  
P: flash.xml  
→ Object  
attributes: Object  
childNodes: Array  
firstChild: XMLNode  
lastChild: XMLNode  
localName: String  
namespaceURI: String  
nextSibling: XMLNode  
nodeName: String  
nodeType: uint  
nodeValue: String  
parentNode: XMLNode  
prefix: String  
previousSibling: XMLNode  
appendChild(): void  
cloneNode(): XMLNode  
getNamespaceForPrefix(): String  
getPrefixForNamespace(): String  
hasChildNodes(): Boolean  
insertBefore(): void  
removeNode(): void  
toString(): String  
XMLNode()

**XMLNodeType**  
P: flash.xml  
→ Object  
ELEMENT\_NODE: uint  
TEXT\_NODE: uint

**XMLSignatureValidator** ●  
P: flash.security  
→ EventDispatcher  
digestStatus: String  
identityStatus: String  
referencesStatus: String  
revocationCheckSetting: String  
signerCN: String  
signerDN: String  
signerExtendedKeyUsages: Array  
signerTrustSettings: Array  
uriDereferencer: URIDereferencer  
useSystemTrustStore: Boolean  
validityStatus: String  
addCertificate(): \*  
verify(): void  
XMLSignatureValidator()

**XMLSocket**  
P: flash.net  
→ EventDispatcher  
connected: Boolean  
close(): void  
connect(): void  
send(): void  
XMLSocket()

**XMLUtil** ■  
P: mx.utils  
→ Object  
createXMLDocument(): XMLDocument  
getAttributeByQName(): XMLList  
qnamesEqual(): Boolean  
qnameToString(): String

**ZipCodeFormatter** ■  
P: mx.flash  
→ Formatter  
formatString: String  
format(): String  
ZipCodeFormatter()

**ZipCodeValidator** ■  
P: mx.validators  
→ Validator  
allowedFormatChars: String  
domain: String  
invalidCharError: String  
invalidDomainError: String  
wrongCAFormatError: String  
wrongLengthError: String  
wrongUSFormatError: String  
doValidation(): Array  
validateZipCode(): Array  
ZipCodeValidator()

**ZipCodeValidatorDomainType** ■  
P: mx.validators  
→ Object  
US\_ONLY: String  
US\_OR\_CANADA: String

**Zoom** ■  
P: mx.effects  
→ TweenEffect  
captureRollEvents: Boolean  
originX: Number  
originY: Number  
zoomHeightFrom: Number  
zoomHeightTo: Number  
zoomWidthFrom: Number  
zoomWidthTo: Number  
Zoom()

**ZoomInstance** ■  
P: mx.effects.effectClasses  
→ TweenEffectInstance  
captureRollEvents: Boolean  
originX: Number  
originY: Number  
zoomHeightFrom: Number  
zoomHeightTo: Number  
zoomWidthFrom: Number  
zoomWidthTo: Number  
ZoomInstance()

# Online Resources and Information

---

## Developer Resources

### API References

Adobe ActionScript 3 Language Reference  
[http://www.adobe.com/go/flex3\\_apiref](http://www.adobe.com/go/flex3_apiref)

Adobe AIR JavaScript Language Reference  
[http://www.adobe.com/go/learn\\_air\\_html\\_jslr](http://www.adobe.com/go/learn_air_html_jslr)

### Documentation

Flex 3 Documentation  
[http://www.adobe.com/go/flex\\_documentation](http://www.adobe.com/go/flex_documentation)

Adobe AIR Documentation  
<http://www.adobe.com/go/airdocs>

### Developer Centers

Adobe Developer Center  
<http://www.adobe.com/devnet>

Adobe AIR Developer Center  
<http://www.adobe.com/go/airdevcenter>

Flex Developer Center  
[http://www.adobe.com/go/flex\\_devcenter](http://www.adobe.com/go/flex_devcenter)

Flash Player Developer Center  
<http://www.adobe.com/go/flashplayerdevcenter>

### Miscellaneous Resources

Adobe Flex Public Bugbase  
<http://bugs.adobe.com/flex/>

Flex.org  
<http://www.flex.org>

Flex Quick Start Tutorials  
[http://www.adobe.com/go/flex3\\_quick\\_start\\_tutorials](http://www.adobe.com/go/flex3_quick_start_tutorials)

Adobe Labs  
<http://www.adobe.com/go/labs>

Open Source Flex  
<http://www.adobe.com/go/opensourceflex>

### Mailing Lists

Flex Coders  
<http://www.adobe.com/go/flexcoders>

Apollo Coders (Adobe AIR)  
<http://tech.groups.yahoo.com/group/apollocoders/>

### Forums

Flex  
[http://www.adobe.com/go/flex\\_forums](http://www.adobe.com/go/flex_forums)

Adobe AIR  
<http://www.adobe.com/go/airforums>

---

## Downloads

Flex SDK  
[http://www.adobe.com/go/flex3\\_sdk](http://www.adobe.com/go/flex3_sdk)

Flex Builder 3  
[http://www.adobe.com/go/flex\\_trial](http://www.adobe.com/go/flex_trial)

Adobe AIR  
<ttp://www.adobe.com/go/getair>

---

## Product Technology Pages

Flex  
<http://www.adobe.com/go/flex>

Flex Builder  
<http://www.adobe.com/go/flexbuilder>

Adobe AIR  
<http://www.adobe.com/go/air>

Flash Player  
<http://www.adobe.com/go/flashplayer>

Adobe Creative Suite 3  
<http://www.adobe.com/go/creativesuite>