

wIntegrate

Object Model Reference

Version 6.0.0
December 2004

IBM Corporation
555 Bailey Avenue
San Jose, CA 95141

Licensed Materials – Property of IBM

© Copyright International Business Machines Corporation 2004. All rights reserved.

AIX, DB2, DB2 Universal Database, Distributed Relational Database Architecture, NUMA-Q, OS/2, OS/390, and OS/400, IBM Informix®, C-ISAM®, Foundation.2000™, IBM Informix® 4GL, IBM Informix® DataBlade® module, Client SDK™, Cloudscape™, Cloudsync™, IBM Informix® Connect, IBM Informix® Driver for JDBC, Dynamic Connect™, IBM Informix® Dynamic Scalable Architecture™ (DSA), IBM Informix® Dynamic Server™, IBM Informix® Enterprise Gateway Manager (Enterprise Gateway Manager), IBM Informix® Extended Parallel Server™, i.Financial Services™, J/Foundation™, MaxConnect™, Object Translator™, Red Brick® Decision Server™, IBM Informix® SE, IBM Informix® SQL, InformiXML™, RedBack®, SystemBuilder™, U2™, UniData®, UniVerse®, wIntegrate® are trademarks or registered trademarks of International Business Machines Corporation.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

Windows, Windows NT, and Excel are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names used in this publication may be trademarks or service marks of others.

Documentation Team: Alan Buckley, Tim Rasmussen, David Robertshaw

US GOVERNMENT USERS RESTRICTED RIGHTS

Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

1. Reference - Objects.....	9
Application	10
AutoBridgeRx	12
AutoBridgeTx	13
AutoData	14
AutoFileTransfer	15
AutoImportTarget	16
AutoWindow	17
Capture	20
DataWatch	22
DisplayArea	23
DockBar	25
HotSpot	27
HotSpots	29
Selection	31
Session	33
Trigger	38
Triggers	40
2. Reference - Properties.....	41
AlwaysOnTop	42
AppName	43
AppTitle	44
Attribute	45
Attributes	47
BackColor	49
BackColor	50
Bottom	51
Busy	52
Caption	53
CaretPosition	54
CaretShape	55
CommandBars	56
ContainerFlags	57

Count	58
Count	59
Count	60
DisplayHorizontalScrollBar	61
DisplayOn	62
DisplayStatusBar	63
DisplayUpdate.....	64
DisplayVerticalScrollBar.....	65
Enabled.....	66
FileName.....	67
ForeColor	68
ForeColor	69
FoundCount	70
FullName.....	71
Height.....	72
HotSpots	73
Id	74
Id	75
Index	76
Information	77
Item	78
Item	79
Left.....	80
Length.....	81
MouseIcon	82
MousePointer.....	83
MousePosition	84
Name	85
Pattern	86
Paused.....	87
RequiredHeight.....	88
Right.....	89
Running.....	90
ScreenArea	91
ScrollArea	92
ScrollArea	93
Selected	94
Selection	95
Session	96

Setting.....	97
SettingsDialog.....	98
StatusBarText.....	99
Style.....	100
Style.....	101
Text.....	102
Text.....	103
Text.....	104
Text.....	105
Text.....	106
TextFormat.....	108
Top.....	109
Triggers.....	110
Type.....	111
Type.....	112
Variable.....	113
Version.....	114
Visible.....	115
WHeight.....	116
WLeft.....	117
WState.....	118
WTop.....	119
WWidth.....	120
Width.....	121
Window.....	122
WindowState.....	123
Word.....	124
hWnd.....	126
3. Reference - Methods.....	127
AboutBox.....	128
Add.....	129
Add.....	131
AppendText.....	133
AttachDockBar.....	134
AttributeFill.....	136
Box.....	137
CanExit.....	138
CanRestore.....	139
Capture.....	140

CharacterFill.....	142
Clear	143
Clear	144
Continue.....	145
Copy.....	146
Copy.....	147
CopySpecial.....	148
DataWatch	149
DisplayArea.....	150
DisplayAttribute.....	151
DisplayCommand.....	152
DisplayText	153
Exit.....	154
Extension	155
GetSessionScript	156
Inflate	157
InvokeDialog	158
MoveCaret	159
Open	160
OpenScript.....	162
Paste.....	163
Pause.....	164
Quit	165
Remove.....	166
RemoveAll.....	167
Resize	168
RunScriptCommand.....	169
RunScriptFile	170
Save.....	171
ScrollAll	172
ScrollLine	173
ScrollPage.....	174
Select	175
SelectAll	177
SelectWindow	178
Send.....	179
SetEnd	180
SetStart	181
SetStatusBar.....	182

SetupOpen	183
SetupSave	184
ShowDialog	185
SnapToSize	186
Start	187
Stop	188
StoreSetting	189
UpdateSettings	190
4. Reference - Events	191
About	192
Another	193
CaptionChanged	194
Change	195
Click	196
DbClick	197
Exit	198
Found	199
Full	200
HeightChanged	201
Open	202
Opened	203
RequestAlwaysOnTop	204
RequestEnable	205
RequestLockSize	206
RequestResize	207
RequireDockBar	208
Save	209
SaveAs	210
ShowStatusBar	211
5. Reference - Enumerations	212
wAttributeTypeEnum	213
wBoxStyleEnum	214
wCaptureFormatEnum	216
wCaptureOptionsEnum	217
wCaptureTargetEnum	219
wCaptureTextFormatEnum	220
wCaretShapeEnum	221
wCursorShapeEnum	222
wDisplayUpdateEnum	223

wHotSpotStyleEnum	224
wMousePositionEnum	225
wPositionEnum	226
wScrollAreaEnum	227
wSelectionFormatEnum	228
wSelectionInfoEnum	229
wSelectionTextEnum	230
wSelectionTypeEnum	231
wSendOptionsEnum	232
wTextColorEnum	233
wTextEffectEnum	234
wTextFormatEnum	235
wTextLayoutEnum	236
wTriggerTypeEnum	237
wWindowStateEnum	238
wWordOptionsEnum	239
Appendix A. Using the Object Model	240
Index	246

Reference - Objects

This chapter is an alphabetical list of the objects in the object model.

Creating a new document object or getting a document object from a wIntegrate configuration file returns an Application object.

The object for a wIntegrate session control is the Session object. This can also be retrieved through the Session property of the Application object.

The following shows the object model hierarchy:

Application->	Session->	AutoWindow->	DisplayArea
		HotSpots->	HotSpot
		Selection	
		Triggers->	Trigger
		Capture	
		DataWatch	

Application

Description

This object represents a wIntegrate application.

Properties

The following table describes the properties for the Application object:

Property	Description
<i>AlwaysOnTop</i>	Represents the always-on-top state of a wIntegrate application window.
<i>Caption</i>	Represents the caption or title text of a wIntegrate object.
<i>DisplayStatusBar</i>	Represents the visible state of a wIntegrate application window status bar.
<i>Height</i>	Represents the height of a wIntegrate object.
<i>Left</i>	Represents the horizontal position of a wIntegrate object.
<i>Session</i>	Returns the Session object associated with a wIntegrate application.
<i>StatusBarText</i>	Represents the text associated with a wIntegrate application window status bar.
<i>Top</i>	Represents the vertical position of a wIntegrate object.
<i>Version</i>	Returns the wIntegrate version number.
<i>Visible</i>	Represents the visible state of a wIntegrate object.
<i>Width</i>	Represents the width of a wIntegrate object.

Property	Description
<i>WindowState</i>	Represents the window state of the application window

Methods

The following table describes the methods for the Application object:

Method	Description
<i>Quit</i>	Quits wIntegrate.
<i>Resize</i>	Resizes a wIntegrate application window.

AutoBridgeRx

Description

This object is for internal use only and is subject to change.

Please do not use.

AutoBridgeTx

Description

This object is for internal use only and is subject to change.

Please do not use.

AutoData

Description

This object is for internal use only and is subject to change.

Please do not use.

AutoFileTransfer

Description

This object is for internal use only and is subject to change.

Please do not use.

AutoImportTarget

Description

This object is for internal use only and is subject to change.

Please do not use.

AutoWindow

Description

This object represents a wIntegrate session window.

There can be only one window object per wIntegrate session.

Properties

The following table describes the properties for the AutoWindow object:

Property	Description
<i>Attribute</i>	Returns the text attribute at a specified location within a wIntegrate session window.
<i>Attributes</i>	Returns the text attributes at a specified location within a wIntegrate session window.
<i>Caption</i>	Represents the caption or title text of a wIntegrate object.
<i>CaretPosition</i>	Represents the position of the text cursor or caret.
<i>CaretShape</i>	Represents the shape of the text cursor or caret.
<i>DisplayHorizontalScrollBar</i>	Represents the visible state of a wIntegrate session window horizontal scroll bar.
<i>DisplayOn</i>	Represents the the display state of a Window object.
<i>DisplayUpdate</i>	Represents the Update state of the Window object.

Property	Description
<i>DisplayVerticalScrollBar</i>	Represents the visible state of a wIntegrate session window vertical scroll bar.
<i>Enabled</i>	Represents the active state of a wIntegrate object.
<i>MouseIcon</i>	Represents the shape of the custom mouse cursor associated with a wIntegrate session window.
<i>MousePointer</i>	Represents the shape of the mouse cursor associated with a wIntegrate session window.
<i>MousePosition</i>	Represents the position of the mouse cursor when over a wIntegrate session window.
<i>ScreenArea</i>	Returns a DisplayArea object that covers the entire display screen of the Window object.
<i>ScrollArea</i>	Returns a DisplayArea object that covers the area of the Window object in which text will scroll.
<i>Text</i>	Returns the text at a specified location within a wIntegrate session window.
<i>Word</i>	Returns the word, or highlighted character of the word, at a specified location within the wIntegrate session window.

Methods

The following table describes the methods for the AutoWindow object:

Method	Description
<i>DisplayArea</i>	Creates a new DisplayArea Object which represent a rectangular section of the Window object.

Method	Description
<i>DisplayAttribute</i>	Sets the terminal emulation text attribute at the current text cursor or caret position in a wIntegrate session window.
<i>DisplayCommand</i>	Runs the emulation command specified.
<i>DisplayText</i>	Displays text in a wIntegrate session window at the current text cursor or caret position.
<i>MoveCaret</i>	Move the text caret to the new column and row.

See Also

[Window property](#)

Capture

Description

This object represents a Capture object associated with a wIntegrate session.

There can be many capture objects per wIntegrate session a new one is created using the Session. Capture method.

When a capture object is created a buffer size is used. When the amount of data reaches the buffer size the Full Event is generated and then the buffer is cleared (unless the wCaptureOptionsShift option was specified).

Properties

The following table describes the properties for the Capture object:

Property	Description
<i>Count</i>	Describes the amount of data currently captured in a wIntegrate capture object.
<i>FileName</i>	Represents the name of the file that receives captured data for a wIntegrate capture object.
<i>Length</i>	Describes the amount of data to capture for a wIntegrate capture object.
<i>Paused</i>	Represents the paused state of a capture object.
<i>Running</i>	Represents the running state of a capture object.
<i>Text</i>	Represents the text in a wIntegrate capture object.
<i>TextFormat</i>	Represents the format that the text in a wIntegrate capture object should be returned.

Methods

The following table describes the methods for the Capture object:

Method	Description
<i>AppendText</i>	Appends text to the data recorded in a capture object.
<i>Clear</i>	Removes or clears all captured data from a wIntegrate capture object.
<i>Continue</i>	Continue a capture after is has been paused with the pause method of a wIntegrate capture object.
<i>Pause</i>	Pauses a capture.
<i>Start</i>	Starts a capture object recording.
<i>Stop</i>	Stops the recording for a capture object.

Events

The following table describes the events for the Capture object:

Event	Description
<i>Full</i>	Called when a wIntegrate capture object is full of data.

See Also

[Capture method](#)

DataWatch

Description

This object represents a DataWatch object associated with a wIntegrate session which will fire an event whenever the specified text is received from the host.

There can be many DataWatch objects per wIntegrate session a new one is created using the Session. DataWatch method.

Properties

The following table describes the properties for the DataWatch object:

Property	Description
<i>Enabled</i>	Represents the active state of a wIntegrate object.
<i>FoundCount</i>	Represents the number of times the text in a DataWatch object has been found.
<i>Text</i>	Represents the text to be found in a wIntegrate DataWatch object.

Events

The following table describes the events for the DataWatch object:

Event	Description
<i>Found</i>	Called when a wIntegrate DataWatch object has match its text against the incoming data from the host.

See Also

[DataWatch method](#)

DisplayArea

Description

This object represents a rectangular area of the wIntegrate Window.

Properties

The following table describes the properties for the DisplayArea object:

Property	Description
<i>Bottom</i>	Represents the bottom vertical position of a wIntegrate object.
<i>Height</i>	Represents the height of a wIntegrate object.
<i>Left</i>	Represents the horizontal position of a wIntegrate object.
<i>Right</i>	Represents the horizontal position of the right hand edge of a wIntegrate object.
<i>ScrollArea</i>	Represents the scroll area type of a display area object.
<i>Text</i>	Returns all the text within a display area object.
<i>Top</i>	Represents the vertical position of a wIntegrate object.
<i>Width</i>	Represents the width of a wIntegrate object.

Methods

The following table describes the methods for the DisplayArea object:

Method	Description
<i>AttributeFill</i>	Fills the area of the Window represented by a DisplayArea object with the specified attribute.
<i>Box</i>	Draws a box in the rectangle area defined by a DisplayArea object.
<i>CharacterFill</i>	Fills the area of a DisplayArea object with the specified character.
<i>Inflate</i>	Increases the size of a DisplayArea object by adding an amount to its sides.

See Also

[DisplayArea method](#), [ScrollArea property \(Window\)](#)

DockBar

Description

The Docking bar control is used in conjunction with the session control in a third party container. e.g. a web page.

By linking the docking bar control to the session with the session controls AttachDockBar method, the docking bar will be populated with the menus and toolbars defined by the session or with script commands.

The DockBar object is the interface for the wIntegrate DockBar control.

Properties

The following table describes the properties for the DockBar object:

Property	Description
<i>Id</i>	Identifier for the dockbar control
<i>RequiredHeight</i>	Height required for the docking bar

Events

The following table describes the events for the DockBar object:

Event	Description
<i>HeightChanged</i>	Fired when a toolbar is added/removed

Version

5.2.0 Supported

See Also

[AttachDockBar method \(Session\)](#)

HotSpot

Description

This object represents a HotSpot object in a wIntegrate session.

There can be many HotSpot objects per wIntegrate session and they are maintained by a HotSpots collection object.

Properties

The following table describes the properties for the HotSpot object:

Property	Description
<i>BackColor</i>	Represents the background color of a wIntegrate HotSpot object.
<i>ForeColor</i>	Represents the foreground color of a wIntegrate HotSpot.
<i>Index</i>	Represents the index of a wIntegrate object within a collection.
<i>Style</i>	Represents the style of a wIntegrate HotSpot object.
<i>Left</i>	Represents the horizontal position of a wIntegrate object.
<i>Top</i>	Represents the vertical position of a wIntegrate object.
<i>Width</i>	Represents the width of a wIntegrate object.
<i>Height</i>	Represents the height of a wIntegrate object.

Events

The following table describes the events for the HotSpot object:

Event	Description
<i>Click</i>	Fired when the object is clicked upon
<i>DbClick</i>	Object has been double clicked on

See Also

[HotSpots](#) object, [Add method \(HotSpots\)](#), [Item property \(HotSpots\)](#)

HotSpots

Description

This object represents a collection of HotSpot objects in a wIntegrate session window.

There can be only one HotSpots object per wIntegrate session and there will always be one associated with a session, ie. there is no need to create the object.

Properties

The following table describes the properties for the HotSpots object:

Property	Description
<i>BackColor</i>	Represents the default background color of a HotSpots collection object.
<i>Count</i>	Represents the number of items in a collection of HotSpot objects.
<i>Enabled</i>	Represents the active state of a wIntegrate object.
<i>ForeColor</i>	Represents the foreground color of a HotSpots collection object.
<i>Item</i>	Represents a HotSpot item in a HotSpots collection.
<i>Style</i>	Represents the style of a HotSpots collection object.

Methods

The following table describes the methods for the HotSpots object:

Method	Description
<i>Add</i>	Adds a new hotspot to a wIntegrate session.

Method	Description
<i>Remove</i>	Removes a specific object from a collection of wIntegrate objects.
<i>RemoveAll</i>	Removes all objects from a collection of wIntegrate objects.

See Also

[HotSpots](#) property, [HotSpot](#) object

Selection

Description

This object represents a selected or highlighted area of a wIntegrate session window.

There can be only one selection object per wIntegrate session and there will always be one associated with a session, ie. there is no need to create the object.

Properties

The following table describes the properties for the Selection object:

Property	Description
<i>Information</i>	Returns information about a Selection.
<i>Selected</i>	Specifies whether an area has been highlighted by a selection or not.
<i>Text</i>	Represents the text highlighted by a selection.
<i>Type</i>	Describes the type of a selection.

Methods

The following table describes the methods for the Selection object:

Method	Description
<i>Clear</i>	Clears or removes the highlighted area of a selection.
<i>Copy</i>	Copies the text associated with a selection to the clipboard.
<i>Select</i>	Selects an area of a wIntegrate session window.
<i>SelectAll</i>	Selects all the data in a wIntegrate session.

Method	Description
<i>SelectWindow</i>	Selects all the visible data in a wIntegrate session window.
<i>SetStart</i>	Sets the start position
<i>SetEnd</i>	Sets the end position

Session

Description

This object represents a wIntegrate session.

There can be only one session object per wIntegrate application and there will always be one associated with an application, ie. there is no need to create the object.

The events from the session object are for use when used as a session control inside a container application. They are mainly requests from the session control for services that need to be controlled/implemented by the container.

Properties

The following table describes the properties for the Session object:

Property	Description
<i>AppName</i>	Application executable name
<i>AppTitle</i>	The title of the application
<i>Busy</i>	This returns non-zero if the session is busy
<i>CommandBars</i>	Reserved
<i>ContainerFlags</i>	Set the options for the session control container
<i>FullName</i>	Full name of the file that opened this session
<i>HotSpots</i>	Returns the HotSpots object associated with a wIntegrate session.
<i>Id</i>	Unique Id for the session
<i>Name</i>	Represents the internal name of a wIntegrate session.
<i>Selection</i>	Returns the selection object associated with a wIntegrate session.

Property	Description
<i>Setting</i>	Set or get setting variables
<i>SettingsDialog</i>	Represents the name of the wIntegrate dialog to use with the setting property / StoreSetting method.
<i>Triggers</i>	Returns the triggers object associated with a wIntegrate session.
<i>Variable</i>	Represents the a global script variable.
<i>WHeight</i>	Sessions internal height.
<i>WLeft</i>	Sessions internal left position
<i>WState</i>	Sessions internal show state.
<i>WTop</i>	Sessions internal top position.
<i>WWidth</i>	Sessions internal width.
<i>Window</i>	Returns the Window object associated with a wIntegrate session.
<i>Version</i>	Returns the wIntegrate version number.
<i>hWnd</i>	The window handle of the session

Methods

The following table describes the methods for the Session object:

Method	Description
<i>AboutBox</i>	Show the session about box
<i>AttachDockBar</i>	Attach a docking bar
<i>CanExit</i>	Internal method to check if the session can exit
<i>CanRestore</i>	Check if its OK to restored the session from minimized
<i>Capture</i>	Creates a new Capture object to record data from the wIntegrate session.

Method	Description
<i>Copy</i>	Copy current selection to the clipboard
<i>CopySpecial</i>	Copy selection to pre-set place using a pre-set format.
<i>DataWatch</i>	Creates a new DataWatch object which fires an event when the specified Text is received from the host.
<i>Exit</i>	Internal method ask session to exit
<i>Extension</i>	Internal method to return an extension
<i>GetSessionScript</i>	Return the session configuration script
<i>InvokeDialog</i>	Invokes the functionality behind a Session dialog box.
<i>Open</i>	Open a script file
<i>OpenScript</i>	Opens script text
<i>Paste</i>	Paste data from clipboard to the session window
<i>RunScriptCommand</i>	Runs a wIntegrate script command.
<i>RunScriptFile</i>	Runs a wIntegrate script file.
<i>Save</i>	Save session parameters as a script
<i>ScrollAll</i>	Scroll to the top or bottom of the window
<i>ScrollLine</i>	Scroll the display back or forward 1 line
<i>ScrollPage</i>	Scroll the display up or down one page
<i>Send</i>	Sends data to a remote host as if the user had typed the data in a wIntegrate session.
<i>SetStatusBar</i>	Internal method to setup the status bar
<i>SetupOpen</i>	Shows open dialog (obsolete)
<i>SetupSave</i>	Obsolete method to show save as dialog
<i>ShowDialog</i>	Displays a Session dialog box.
<i>SnapToSize</i>	Internal method to implement snap to size

Method	Description
<i>StoreSetting</i>	Allows the value session setting variable to be stored for update by a following UpdateSettings command.
<i>UpdateSettings</i>	Allows the value session setting variable to be stored for update and then updated all at once by this method.

Events

The following table describes the events for the Session object:

Event	Description
<i>About</i>	Request to show about box
<i>Another</i>	Request for the File Another dialog
<i>CaptionChanged</i>	The session title has changed
<i>Exit</i>	Request container to Exit
<i>Open</i>	Request to Open a file
<i>Opened</i>	Obsolete. No longer generated
<i>RequestAlwaysOnTop</i>	Request container is made always on top
<i>RequestEnable</i>	Request container is enabled or disabled
<i>RequestLockSize</i>	Request that the container does not change it size
<i>RequestResize</i>	Request to container to resize the session control
<i>RequireDockBar</i>	Internal event requesting a docking bar
<i>Save</i>	Request to Save the current file
<i>SaveAs</i>	Request to show save as dialog box
<i>ShowStatusBar</i>	Request status bar is shown

Version

5.2.0 ContainerFlags property added

See Also

[Session property](#)

Trigger

Description

This object represents a Trigger object in a wIntegrate session. A trigger causes a notification event when received data changes.

There can be many Trigger objects per wIntegrate session and they are maintained by a Triggers collection object.

Properties

The following table describes the properties for the Trigger object:

Property	Description
<i>Height</i>	Represents the height of a wIntegrate object.
<i>Left</i>	Represents the horizontal position of a wIntegrate object.
<i>Pattern</i>	The text monitored by the trigger.
<i>Top</i>	Represents the vertical position of a wIntegrate object.
<i>Type</i>	Describes the type of a wIntegrate trigger object.
<i>Width</i>	Represents the width of a wIntegrate object.

Events

The following table describes the events for the Trigger object:

Event	Description
<i>Change</i>	Called when the data associated with a wIntegrate trigger has changed.

See Also

[Triggers object](#), [Add method \(Triggers\)](#), [Item property \(Triggers\)](#)

Triggers

Description

This object represents a collection of Trigger objects in a wIntegrate session. A trigger causes a notification event when received data changes.

There can be only one Triggers object per wIntegrate session.

Properties

The following table describes the properties for the Triggers object:

Property	Description
<i>Count</i>	Represents the number of items in a collection of Trigger objects.
<i>Item</i>	Represents a Trigger item in a Triggers collection.

Methods

The following table describes the methods for the Triggers object:

Method	Description
<i>Add</i>	Adds a new trigger object to a wIntegrate session.

Reference - Properties

This chapter contains an alphabetical list of the properties in the object model.

AlwaysOnTop

Syntax

`AlwaysOnTop`

Description

Represents the always-on-top state of a wIntegrate application window. When the wIntegrate application window is always-on-top, all other windows will appear to be behind it.

Parameters

None

Property Value

Read/Write Boolean. The session window always-on-top state

Applies to

[Application](#)

AppName (Session)

Syntax

AppName

Description

The name of the executable containing the session control.

This should only be set by application's that contain the session control. It is not required that this property is set.

Parameters

None

Property Value

Read/Write String. Name of the executable for the application.

Applies to

[Session](#)

AppTitle (Session)

Syntax

AppTitle

Description

The title of the application.

This should only be set by application's that contain the session control and is used in the script language or in message boxes where the application name is displayed.

Parameters

None

Property Value

Read/Write String. The title of the application

Applies to

[Session](#)

Attribute

Syntax

Attribute *Type*, [*Left*], [*Top*], [*Page*]

Description

Returns the text attribute at a specified location within a wIntegrate session window.

If any of the Left, Top and Page arguments are omitted, their values are taken to be those of the current text cursor or caret position. If all of them are omitted, the current attribute is returned (ie. the attribute that will be applied to the next text character displayed).

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the Attribute method:

Parameters	Type	Description
<i>Type</i>	wAttributeTypeEnum	The attribute type
<i>Left</i>	Variant or Long	The column number of the attribute (default: 0)
<i>Top</i>	Variant or Long	The row number of the attribute (default: 0)
<i>Page</i>	Variant or Long	The page number of the attribute (default: the first page)

Property Value

Read-only Long. The required text attribute

Applies to

[AutoWindow](#)

Attributes

Syntax

Attributes *Type*, [*Left*], [*Top*], [*Right*], [*Bottom*], [*Layout*], [*StartPage*], [*EndPage*]

Description

Returns the text attributes at a specified location within a wIntegrate session window.

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the Attributes method:

Parameters	Type	Description
<i>Type</i>	wAttributeTypeEnum	The attribute type
<i>Left</i>	Variant or Long	The column number of the top left of the associated text (default: 0)
<i>Top</i>	Variant or Long	The row number of the top left of the associated text (default: 0)
<i>Right</i>	Variant or Long	The column number of the bottom right of the associated text (default: the right of the session window)
<i>Bottom</i>	Variant or Long	The row number of the bottom right of the associated text (default: the bottom of the session window)
<i>Layout</i>	Variant or wTextLayoutEnum	The layout of the associated text (default: wTextLayoutLines)

Parameters	Type	Description
<i>StartPage</i>	Variant or Long	The page number of the top left of the associated text (default: the first page)
<i>EndPage</i>	Variant or Long	The page number of the bottom right of the associated text (default: 0)

Property Value

Read-only Long Array. The required text attributes

Applies to

[AutoWindow](#)

BackColor (HotSpot)

Syntax

BackColor

Description

Represents the background color of a wIntegrate HotSpot object.

The color cannot be applied to an object if its style does not allow it - see HotSpot.Style.

Parameters

None

Property Value

Read/Write [wTextColorEnum](#). A value representing a color

Applies to

[HotSpot](#)

BackColor (HotSpots)

Syntax

BackColor

Description

Represents the default background color of a HotSpots collection object. It applies to all HotSpot objects in the HotSpots collection which do not set have an individual style.

The color is only applied to the hotspot when the style allows it - see HotSpots.Style.

Parameters

None

Property Value

Read/Write [wTextColorEnum](#). A value representing a color

Applies to

[HotSpots](#)

Bottom

Syntax

Bottom

Description

Represents the bottom vertical position of a wIntegrate object.

Parameters

None

Property Value

Read/Write Long. The vertical position of the object in pixels

Applies to

[DisplayArea](#)

Busy (Session)

Syntax

Busy

Description

This returns non-zero if the session is busy.

Currently it just checks if a file transfer is in progress

Parameters

None

Property Value

Read-only Boolean. True if the session object is busy.

Applies to

[Session](#)

Caption

Syntax

Caption

Description

Represents the caption or title text of a wIntegrate object.

When this property is set the special characters from the Setup Preference dialog box for the title can be used to represent the current port name, number etc. When this property is retrieved the caption will include the current value of these parameters.

Parameters

None

Property Value

Read/Write String. The caption text

Applies to

[Application](#), [AutoWindow](#)

CaretPosition

Syntax

CaretPosition *Plane*

Description

Represents the position of the text cursor or caret.

Parameters

The following table describes the parameters of the CaretPosition method:

Parameters	Type	Description
<i>Plane</i>	wPositionEnum	The plane of the caret position

Property Value

Read/Write Long. The caret position

Applies to

[AutoWindow](#)

CaretShape

Syntax

CaretShape

Description

Represents the shape of the text cursor or caret.

Parameters

None

Property Value

Read/Write [wCaretShapeEnum](#). The caret shape

Applies to

[AutoWindow](#)

CommandBars

Syntax

CommandBars

Description

This property is reserved for future use. It currently returns a NULL object.

Parameters

None

Applies to

[Session](#)

ContainerFlags (Session)

Syntax

ContainerFlags

Description

This property controls how the session control deals with the File and Exit handling when used in another application.

By default these menu options send an event to the container so that it can process these menu options itself. These flags allow the session control to process these options.

Parameters

None

Property Value

Read/Write Long. Sets the File menu options handled by the session control. The value is either 0 to let the container handle the menu options or a combination of one or more of the following; 1 - File Open, 2 - File Another, 4 - File Save As, 8 - File Save, 16 - File Exit/Exit All.

Applies to

[Session](#)

Version

5.2 Original version

Count (Capture)

Syntax

Count

Description

Describes the amount of data currently captured in a wIntegrate capture object.

Parameters

None

Property Value

Read-only Long. The amount of data captured

Applies to

[Capture](#)

Count (HotSpots)

Syntax

Count

Description

Represents the number of items in a collection of HotSpot objects.

Parameters

None

Property Value

Read-only Short. The number of items in the collection

Applies to

[HotSpots](#)

Count (Triggers)

Syntax

Count

Description

Represents the number of items in a collection of Trigger objects.

Parameters

None

Property Value

Read-only Long. The number of items in the collection

Applies to

[Triggers](#)

DisplayHorizontalScrollBar

Syntax

DisplayHorizontalScrollBar

Description

Represents the visible state of a wIntegrate session window horizontal scroll bar.

Parameters

None

Property Value

Read/Write Boolean. The session window horizontal scroll bar display state

Applies to

[AutoWindow](#)

DisplayOn

Syntax

DisplayOn

Description

Represents the the display state of a Window object. When the display is turned off the contents of the Window are hidden.

Parameters

None

Property Value

Read/Write Boolean. The display state

Applies to

[AutoWindow](#)

DisplayStatusBar

Syntax

DisplayStatusBar

Description

Represents the visible state of a wIntegrate application window status bar.

Parameters

None

Property Value

Read/Write Boolean. The session window status bar display state

Applies to

[Application](#)

DisplayUpdate

Syntax

DisplayUpdate

Description

Represents the Update state of the Window object. This allows the update of the display to be temporarily stopped to decrease screen flashing. The DisplayUpdate should only be turned off for short periods of time.

Parameters

None

Property Value

Read/Write [wDisplayUpdateEnum](#). The update state

Applies to

[AutoWindow](#)

DisplayVerticalScrollBar

Syntax

DisplayVerticalScrollBar

Description

Represents the visible state of a wIntegrate session window vertical scroll bar.

Parameters

None

Property Value

Read/Write Boolean. The session window vertical scroll bar display state

Applies to

[AutoWindow](#)

Enabled

Syntax

Enabled

Description

Represents the active state of a wIntegrate object. For a window object, the state is active when output appears in the window and inactive when no output appears.

Parameters

None

Property Value

Read/Write Boolean. The active state

Applies to

[AutoWindow](#), [DataWatch](#), [HotSpots](#)

FileName

Syntax

FileName

Description

Represents the name of the file that receives captured data for a wIntegrate capture object. The capture target must have been set to wCaptureTargetFile for capturing data to a file.

Parameters

None

Property Value

Read-only String. The capture data file name

Applies to

[Capture](#)

ForeColor (HotSpot)

Syntax

ForeColor

Description

Represents the foreground color of a wIntegrate HotSpot.

The color cannot be applied to an object if its style does not allow it - see HotSpot.Style.

Parameters

None

Property Value

Read/Write [wTextColorEnum](#). A value representing a color

Applies to

[HotSpot](#)

ForeColor (HotSpots)

Syntax

ForeColor

Description

Represents the foreground color of a HotSpots collection object. It applies to all HotSpot objects in the HotSpots collection which use the default style.

The color cannot be applied to an object if its style does not allow it - see HotSpots.Style.

Parameters

None

Property Value

Read/Write [wTextColorEnum](#). A value representing a color

Applies to

[HotSpots](#)

FoundCount

Syntax

FoundCount

Description

Represents the number of times the text in a DataWatch object has been found. The count is automatically reset to zero if a disabled DataWatch object is then enabled.

Parameters

None

Property Value

Read/Write Long. The number of times the text from the DataWatch object has been found.

Applies to

[DataWatch](#)

FullName (Session)

Syntax

FullName

Description

Full name of the file that opened this session.

Setting this property only changes the name internally.

Parameters

None

Property Value

Read/Write String. Full name of the file that opened this session

Applies to

[Session](#)

Height

Syntax

Height

Description

Represents the height of a wIntegrate object.

Parameters

None

Property Value

Read/Write Long. The height of the object in pixels

Applies to

[Application](#), [DisplayArea](#), [HotSpot](#), [Trigger](#)

HotSpots

Syntax

HotSpots

Description

Returns the HotSpots object associated with a wIntegrate session.

Parameters

None

Property Value

Read-only [HotSpots](#). The hotspots object

Applies to

[Session](#)

Id (DockBar)

Syntax

Id

Description

This property provides the identifier required for the AttachDockBar method on the session object.

Parameters

None

Applies to

[DockBar](#)

Version

5.2.0 First supported

Id (Session)

Syntax

Id

Description

The numeric Id for the session. Each session is assigned a unique Id.

Parameters

None

Property Value

Read-only Long. The Id for the session

Applies to

[Session](#)

Index

Syntax

Index

Description

Represents the index of a wIntegrate object within a collection.

The index can change during the lifetime of the object as other objects are added to or removed from the collection.

Parameters

None

Property Value

Read-only Long. The index of the object

Applies to

[Capture](#), [HotSpot](#)

Information

Syntax

Information *Type*

Description

Returns information about a Selection. The information depends on the specified argument.

Parameters

The following table describes the parameters of the Information method:

Parameters	Type	Description
<i>Type</i>	wSelectionInfoEnum	The type of information required

Property Value

Read-only Variant. The required information. The actual value type depends on theType argument

Applies to

[Selection](#)

Item (HotSpots)

Syntax

Item *Index*

Description

Represents a HotSpot item in a HotSpots collection.

This is the default property of the HotSpots collection.

Parameters

The following table describes the parameters of the Item method:

Parameters	Type	Description
<i>Index</i>	Variant or Long	The index of the item required

Property Value

Read-only [HotSpot](#). The required collection item

Applies to

[HotSpots](#)

Item (Triggers)

Syntax

Item *Index*

Description

Represents a Trigger item in a Triggers collection.

This is the default property for the triggers collection.

Parameters

The following table describes the parameters of the Item method:

Parameters	Type	Description
<i>Index</i>	Variant or Long	The index of the item required

Property Value

Read-only [Trigger](#). The required collection item

Applies to

[Triggers](#)

Left

Syntax

Left

Description

Represents the horizontal position of a wIntegrate object.

Parameters

None

Property Value

Read/Write Long. The horizontal position of the object in pixels

Applies to

[Application](#), [DisplayArea](#), [HotSpot](#), [Trigger](#)

Length

Syntax

Length

Description

Describes the amount of data to capture for a wIntegrate capture object. When the amount of data received reaches the length a Full event will be generated.

Parameters

None

Property Value

Read-only Long. The capture data length

Applies to

[Capture](#)

MouseIcon

Syntax

MouseIcon

Description

Represents the shape of the custom mouse cursor associated with a wIntegrate session window. Custom cursors maybe defined and stored in a file that should be called wCursor.dll and placed in the wIntegrate folder. All the custom cursors should be given names and it is these names that will be used to set the mouse cursor shape. To set the mouse cursor to a custom cursor shape, set the MouseIcon property to the required cursor then set the MousePointer property to the custom cursor value.

Parameters

None

Property Value

Read/Write String. The name of the custom mouse cursor shape

Applies to

[AutoWindow](#)

MousePointer

Syntax

MousePointer

Description

Represents the shape of the mouse cursor associated with a wIntegrate session window. It can be used to set the mouse cursor to one of the standard Windows shapes or, in conjunction with the MouseIcon property, a custom shape.

Parameters

None

Property Value

Read/Write [wCursorShapeEnum](#). The mouse cursor shape

Applies to

[AutoWindow](#)

MousePosition

Syntax

MousePosition *Parameter*

Description

Represents the position of the mouse cursor when over a wIntegrate session window.

Parameters

The following table describes the parameters of the MousePosition method:

Parameters	Type	Description
<i>Parameter</i>	wMousePositionEnum	The mouse position information required

Property Value

Read-only Long. The required information

Applies to

[AutoWindow](#)

Name (Session)

Syntax

Name

Description

Represents the internal name of a wIntegrate session. wIntegrate ensures that each session has a unique name so, although the property can be changed, it is inadvisable to do so.

Parameters

None

Property Value

Read/Write String. wIntegrate session internal name

Applies to

[Session](#)

Pattern

Syntax

Pattern

Description

Represents the pattern or text monitored by a wIntegrate trigger object.

For trigger types that do not include a pattern or text an empty string is returned.

Parameters

None

Property Value

Read-only String. The monitored text

Applies to

[Trigger](#)

Paused

Syntax

Paused

Description

Represents the paused state of a capture object.

Parameters

None

Property Value

Read-only Boolean. The paused state of the capture

Applies to

[Capture](#)

RequiredHeight (DockBar)

Syntax

RequiredHeight

Description

This property returns the minimum height required for the docking bar to show all the toolbars currently on it.

It is usually used on the receipt of a HeightChanged event from the docking bar to find out the best height to resize the docking bar to.

Parameters

None

Applies to

[DockBar](#)

Version

5.2.0 Added

Right

Syntax

Right

Description

Represents the horizontal position of the right hand edge of a wIntegrate object.

Parameters

None

Property Value

Read/Write Long. The horizontal position of the object in pixels

Applies to

[DisplayArea](#)

Running

Syntax

Running

Description

Represents the running state of a capture object.

Parameters

None

Property Value

Read-only Boolean. The running state of the capture

Applies to

[Capture](#)

ScreenArea

Syntax

ScreenArea

Description

Returns a DisplayArea object that covers the entire display screen of the Window object.

Parameters

None

Property Value

Read-only [DisplayArea](#). The DisplayArea object which covers the screenl area

Applies to

[AutoWindow](#)

Example

Put a border around the edge of the screen region

```
Session.Window.ScreenArea.Box wBoxStyleSingleBorder
```

ScrollArea (DisplayArea)

Syntax

ScrollArea

Description

Represents the scroll area type of a display area object. There can only be one scroll area so setting a new scroll area cancels an existing one. The scroll area is linked to the area of the screen so moving or resizing the DisplayArea object can change this property.

Parameters

None

Property Value

Read/Write [wScrollAreaEnum](#). The type of the scroll area

Applies to

[DisplayArea](#)

ScrollArea (Window)

Syntax

ScrollArea

Description

Returns a DisplayArea object that covers the area of the Window object in which text will scroll.

Parameters

None

Property Value

Read-only [DisplayArea](#). The DisplayArea object which covers the scroll area

Applies to

[AutoWindow](#)

Example

Clear the current scroll region

```
Session.Window.ScrollArea.ScrollArea = wScrollAreaNo
```

Selected

Syntax

Selected

Description

Specifies whether an area has been highlighted by a selection or not.

Parameters

None

Property Value

Read-only Boolean. The selected state

Applies to

[Selection](#)

Selection

Syntax

Selection

Description

Returns the selection object associated with a wIntegrate session.

Parameters

None

Property Value

Read-only [Selection](#). The selection object

Applies to

[Session](#)

Session

Syntax

Session

Description

Returns the Session object associated with a wIntegrate application.

Parameters

None

Property Value

Read-only [Session](#). The session object

Applies to

[Application](#)

Setting

Syntax

Setting *Name*

Description

Allows the setting of variables that are saved in a wIntegrate session or the variable associated with a dialog to be set or interrogated. The SettingsDialog property can be used to specify the dialog to use.

See the "Client Scripting Reference" manual, "Reference - Script Menu Options" chapter to see the available variables.

Parameters

The following table describes the parameters of the Setting method:

Parameters	Type	Description
<i>Name</i>	String	The name of the setting to use.

Property Value

Read/Write String. The value of the setting

Applies to

[Session](#)

Example

Set the emulation to vt100

```
wApp.Session.Setting("Emulation") = "vt100"
```

SettingsDialog

Syntax

SettingsDialog

Description

Represents the name of the wIntegrate dialog to use with the Setting property or StoreSetting method. For global session variables this value does not need to be set.

See the "Client Scripting Reference" manual, "Reference - Script Menu Options" chapter to see the available dialog (menu option) names.

Parameters

None

Property Value

Read/Write String. wIntegrate dialog name

Applies to

[Session](#)

StatusBarText

Syntax

StatusBarText

Description

Represents the text associated with a wIntegrate application window status bar.

Parameters

None

Property Value

Read/Write String. The session window status bar text

Applies to

[Application](#)

Style (HotSpot)

Syntax

Style

Description

Represents the style of a wIntegrate HotSpot object.

Parameters

None

Property Value

Read/Write [wHotSpotStyleEnum](#). A value representing a hotspot style

Applies to

[HotSpot](#)

Style (HotSpots)

Syntax

Style

Description

Represents the style of a HotSpots collection object. It applies to all HotSpot objects in the HotSpots collection.

Parameters

None

Property Value

Read/Write [wHotSpotStyleEnum](#). A value representing a hotspot style

Applies to

[HotSpots](#)

Text (Capture)

Syntax

Text

Description

Represents the text in a wIntegrate capture object. The format the text is returned in can be specified using the TextFormat property. The default format strips out character zero from the returned text.

Parameters

None

Property Value

Read-only String. The captured text

Applies to

[Capture](#)

Text (DataWatch)

Syntax

Text

Description

Represents the text to be found in a wIntegrate DataWatch object.

Parameters

None

Property Value

Read-only String. The text that is/will be watched for.

Applies to

[DataWatch](#)

Text (DisplayArea)

Syntax

Text [*Format*]

Description

Returns all the text within a display area object.

Parameters

The following table describes the parameters of the Text method:

Parameters	Type	Description
<i>Format</i>	wTextFormatEnum	The format of the required text (default: wTextFormatNone)

Property Value

Read-only String. The required text

Applies to

[DisplayArea](#)

Text (Selection)

Syntax

Text Format

Description

Represents the text highlighted by a selection.

Parameters

The following table describes the parameters of the Text method:

Parameters	Type	Description
<i>Format</i>	wSelectionTextEnum	The required format

Property Value

Read-only String. The selected text

Applies to

[Selection](#)

Text (Window)

Syntax

Text [*Left*], [*Top*], [*Right*], [*Bottom*], [*Layout*], [*Format*], [*StartPage*], [*EndPage*]

Description

Returns the text at a specified location within a wIntegrate session window.

This is the default property for the Window object.

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the Text method:

Parameters	Type	Description
<i>Left</i>	Variant or Long	The column number of the top left of the required text (default: 0)
<i>Top</i>	Variant or Long	The row number of the top left of the required text (default: 0)
<i>Right</i>	Variant or Long	The column number of the bottom right of the required text (default: the right of the session window)
<i>Bottom</i>	Variant or Long	The row number of the bottom right of the required text (default: the bottom of the session window)
<i>Layout</i>	Variant or wTextLayoutEnum	The layout of text required (default: wTextLayoutLines)

Parameters	Type	Description
<i>Format</i>	Variant or wTextFormatEnum	The format of the required text (default: wTextFormatNone)
<i>StartPage</i>	Variant or Long	The page number of the top left of the required text (default: the first page)
<i>EndPage</i>	Variant or Long	The page number of the bottom right of the required text (default: 0)

Property Value

Read-only Variant. The required text

Applies to

[AutoWindow](#)

TextFormat

Syntax

TextFormat

Description

Represents the format that the text in a wIntegrate capture object should be returned. The default value is wCaptureTextFormatStripZeroes.

Parameters

None

Property Value

Read/Write [wCaptureTextFormatEnum](#). The format to return the captured text

Applies to

[Capture](#)

Top

Syntax

Top

Description

Represents the vertical position of a wIntegrate object.

Parameters

None

Property Value

Read/Write Long. The vertical position of the object in pixels

Applies to

[Application](#), [DisplayArea](#), [HotSpot](#), [Trigger](#)

Triggers

Syntax

Triggers

Description

Returns the triggers object associated with a wIntegrate session.

Parameters

None

Property Value

Read-only [Triggers](#). The triggers object

Applies to

[Session](#)

Type (Selection)

Syntax

Type

Description

Describes the type of a selection.

Parameters

None

Property Value

Read/Write [wSelectionTypeEnum](#). The selection type

Applies to

[Selection](#)

Type (Trigger)

Syntax

Type

Description

Describes the type of a wIntegrate trigger object.

Parameters

None

Property Value

Read-only [wTriggerTypeEnum](#). The trigger type

Applies to

[Trigger](#)

Variable

Syntax

Variable Name

Description

Represents the a global script variable. This allows you to set and get variable which can be used by scripting.

Parameters

The following table describes the parameters of the Variable method:

Parameters	Type	Description
<i>Name</i>	String	The name of the global script variable

Property Value

Read/Write String. The value of the global script variable

Applies to

[Session](#)

Version

Syntax

Version

Description

Returns the wIntegrate version number.

Parameters

None

Property Value

Read-only String. The version number

Applies to

[Application](#), [Session](#)

Visible

Syntax

Visible

Description

Represents the visible state of a wIntegrate object.

Parameters

None

Property Value

Read/Write Boolean. The visible state

Applies to

[Application](#)

WHeight (Session)

Syntax

WHeight

Description

Sessions internal height.

This property is for internal use only and is subject to change.

Please do not use.

Parameters

None

Property Value

Read/Write Long. Sessions height

Applies to

[Session](#)

WLeft (Session)

Syntax

WLeft

Description

Sessions internal left position.

This property is for internal use only and is subject to change.

Please do not use.

Parameters

None

Property Value

Read/Write Long. Left position

Applies to

[Session](#)

WState (Session)

Syntax

WState

Description

Sessions internal show state.

This property is for internal use only and is subject to change.

Please do not use.

Parameters

None

Property Value

Read/Write Short. The internal state

Applies to

[Session](#)

WTop (Session)

Syntax

WTop

Description

Sessions internal top position.

This property is for internal use only and is subject to change.

Please do not use.

Parameters

None

Property Value

Read/Write Long. Top position

Applies to

[Session](#)

WWidth (Session)

Syntax

WWidth

Description

Sessions internal width.

This property is for internal use only and is subject to change.

Please do not use.

Parameters

None

Property Value

Read/Write Long. Internal width

Applies to

[Session](#)

Width

Syntax

Width

Description

Represents the width of a wIntegrate object.

Parameters

None

Property Value

Read/Write Long. The width of the object in pixels

Applies to

[Application](#), [DisplayArea](#), [HotSpot](#), [Trigger](#)

Window

Syntax

Window

Description

Returns the Window object associated with a wIntegrate session.

Parameters

None

Property Value

Read-only [AutoWindow](#). The window object

Applies to

[Session](#)

WindowState

Syntax

WindowState

Description

Represents the window state of a wIntegrate application window, ie. maximized, minimized, etc.

Parameters

None

Property Value

Read/Write [wWindowStateEnum](#). The wIntegrate session window state

Applies to

[Application](#)

Word

Syntax

Word [*Left*], [*Top*], [*Options*], [*Delimiter*], [*Page*]

Description

Returns the word, or highlighted character of the word, at a specified location within the wIntegrate session window.

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the Word method:

Parameters	Type	Description
<i>Left</i>	Variant or Long	The column number of the top left of the required word (default: current text cursor position)
<i>Top</i>	Variant or Long	The row number of the top left of the required word (default: current text cursor position)
<i>Options</i>	Variant or wWordOptionsEnum	Specifies what to retrieve (default: wDataWordOptionsWord)
<i>Delimiter</i>	Variant or Long	The delimiting character used to determine the location of the end of the word (default: " ")
<i>Page</i>	Variant or Long	The page number of the required word (default: current text cursor position)

Property Value

Read-only String. The required text

Applies to

[AutoWindow](#)

Example

Gets the word at the current text cursor position:

```
MyApp = CreateObject("wIntegrate.Document")  
MyWord = MyApp.Session.Window.Word
```

Gets the word from the start of the current page that is delimited by a comma:

```
MyApp = CreateObject("wIntegrate.Document")  
MyWord = MyApp.Session.Window.Word(0, 0, wDataWordOptionsDelimiter,  
",")
```

hWnd (Session)

Syntax

hWnd

Description

The Window's window handle of the session.

Parameters

None

Property Value

Read-only Long. The window handle.

Applies to

[Session](#)

Reference - Methods

This chapter contain an alphabetical list of the methods in the object model.

AboutBox (Session)

Syntax

AboutBox

Description

Show the session control about box

Parameters

None

Applies to

[Session](#)

Add (HotSpots)

Syntax

Add *Left, Top, Right, Bottom, [Style], [ForeColor], [BackColor]*

Description

Adds a new hotspot to a wIntegrate session. A hotspot must be removed with HotSpot Remove or HotSpot RemoveAll when it is no longer required.

Note that rows and columns start at 0 - the top left of a wIntegrate session being at row 0, column 0.

The ForeColor and BackColor arguments only apply if the wHotSpotStyleColored is used in the Style argument.

Parameters

The following table describes the parameters of the Add method:

Parameters	Type	Description
<i>Left</i>	Long	Specifies the column number of the top left of the hotspot
<i>Top</i>	Long	Specifies the row number of the top left of the hotspot
<i>Right</i>	Long	Specifies the column number of the bottom right of the hotspot
<i>Bottom</i>	Long	Specifies the row number of the bottom right of the hotspot
<i>Style</i>	Variant or wHotSpotStyleEnum	Specifies the style for this hotspot. Defaults to use the style from the HotSpots collection.

Parameters	Type	Description
<i>ForeColor</i>	Variant or wTextColorEnum	Specifies the foreground color.
<i>BackColor</i>	Variant or wTextColorEnum	Specifies the background color.

Applies to

[HotSpots](#)

Add (Triggers)

Syntax

Add Pattern, [Type], [Left], [Top], [Right], [Bottom]

Description

Adds a new trigger object to a wIntegrate session.

Note that rows and columns start at 0 - the top left of a wIntegrate session being at row 0, column 0.

Triggers can be applied to incoming data or to data that appears in the session window, depending on the value of the Type argument. The Left , Top , Right and Bottom must be specified if not triggering on the incoming data.

Parameters

The following table describes the parameters of the Add method:

Parameters	Type	Description
<i>Pattern</i>	String	Specifies the text pattern used to cause the trigger
<i>Type</i>	Variant or wTriggerTypeEnum	Specifies the type of trigger
<i>Left</i>	Variant or Long	Specifies the column number of the top left of the trigger
<i>Top</i>	Variant or Long	Specifies the row number of the top left of the trigger
<i>Right</i>	Variant or Long	Specifies the column number of the bottom right of the trigger
<i>Bottom</i>	Variant or Long	Specifies the row number of the bottom right of the trigger

Applies to

Triggers

Example

Add a new trigger to detect the select prompt at 10,22

```
Set PromptTrig = MyTriggers.Add("Select option:",wTriggerTypeText, 10, 22, 35, 10)
```

AppendText

Syntax

AppendText *Text*

Description

Appends text to the data recorded in a capture object.

Parameters

The following table describes the parameters of the AppendText method:

Parameters	Type	Description
<i>Text</i>	String	The text to append to the recording

Applies to

[Capture](#)

AttachDockBar (Session)

Syntax

AttachDockBar *DockBarId, Where, Attached*

Description

This method is used in an external container application (e.g. a web page) to attach a docking bar control to a session.

Once a docking bar control is attached, it will be updated with the appropriate script commands in the session.

Parameters

The following table describes the parameters of the AttachDockBar method:

Parameters	Type	Description
<i>DockBarId</i>	Long	Docking bar id
<i>Where</i>	Long	Location of the docking bar in respect to the session. 0 is above the session. 1 is below the session.
<i>Attached</i>	Boolean	True if the dock bar was attached correctly

Applies to

[Session](#)

Example

See "Example\HTML\WintJS.htm".

Version

5.2.0 First supported

AttributeFill

Syntax

AttributeFill *Type, Attribute*

Description

Fills the area of the Window represented by a DisplayArea object with the specified attribute.

Parameters

The following table describes the parameters of the AttributeFill method:

Parameters	Type	Description
<i>Type</i>	wAttributeTypeEnum	The attribute type
<i>Attribute</i>	wTextEffectEnum or wTextColorEnum	Specifies the value of the attribute, which depends on the attribute type

Applies to

[DisplayArea](#)

Box

Syntax

Box *Style*

Description

Draws a box in the rectangle area defined by a `DisplayArea` object. If the `DisplayArea` object has a `Width` or `Depth` of 1 a line is drawn.

Parameters

The following table describes the parameters of the `Box` method:

Parameters	Type	Description
<i>Style</i>	wBoxStyleEnum	The style to use for the box drawing

Applies to

[DisplayArea](#)

CanExit (Session)

Syntax

CanExit *Flags*

Description

Internal method to check if the session can exit.

This method is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the CanExit method:

Parameters	Type	Description
<i>Flags</i>	Long *	Input/Output flags

Return Value

Boolean. True if ok to exit

Applies to

[Session](#)

CanRestore (Session)

Syntax

CanRestore

Description

Check if its OK to restored the session from minimized.

This method checks if a restore script has been set for the session and runs it and returns FALSE if it has, otherwise it returns TRUE.

This is an internal method used to help implement the "Event OnRestore" script command.

Parameters

None

Return Value

Boolean. Returns true if it is OK to restore a minimized session.

Applies to

[Session](#)

Capture

Syntax

Capture [*Target*], [*Format*], [*Options*], [*Length*], [*FileName*], [*Widths*]

Description

Creates a new Capture object to record data from the wIntegrate session.

If capturing data to a file, both the FileName and Target arguments must be set appropriately.

Parameters

The following table describes the parameters of the Capture method:

Parameters	Type	Description
<i>Target</i>	Variant or wCaptureTargetEnum	Specifies the target or destination for the captured data
<i>Format</i>	Variant or wCaptureFormatEnum	Specifies the format of the captured data
<i>Options</i>	Variant or wCaptureOptionsEnum	Specifies the type of capture
<i>Length</i>	Variant or Long	Specifies the length of the buffer used to capture the data
<i>FileName</i>	Variant or String	Specifies the file name to capture the data to
<i>Widths</i>	Variant or String	Specifies a comma separated list of the field widths for the wCaptureFormatTable capture

Return Value

[Capture](#). The capture object representing this capture

Applies to

[Session](#)

CharacterFill

Syntax

CharacterFill *Character*

Description

Fills the area of a DisplayArea object with the specified character.

Parameters

The following table describes the parameters of the CharacterFill method:

Parameters	Type	Description
<i>Character</i>	String	The character to fill the area with

Applies to

[DisplayArea](#)

Clear (Capture)

Syntax

Clear

Description

Removes or clears all captured data from a wIntegrate capture object.

Parameters

None

Applies to

[Capture](#)

Clear (Selection)

Syntax

Clear

Description

Clears or removes the highlighted area of a selection.

Parameters

None

Applies to

[Selection](#)

Continue

Syntax

Continue

Description

Continue a capture after it has been paused with the pause method of a wIntegrate capture object.

Parameters

None

Applies to

[Capture](#)

Copy

Syntax

Copy Format

Description

Copies the text associated with a selection to the clipboard.

Parameters

The following table describes the parameters of the Copy method:

Parameters	Type	Description
<i>Format</i>	wSelectionFormatEnum	Specifies the clipboard format

Applies to

[Selection](#)

Copy (Session)

Syntax

Copy

Description

Copy current selection to the clipboard.

This method is obsolete and no longer supported, use the Selection.Copy method instead.

Parameters

None

Applies to

[Session](#)

CopySpecial (Session)

Syntax

CopySpecial

Description

Copy the current selection to the clipboard using the format defined in the Edit Copy Special to menu option.

This method is obsolete and no longer supported, use the Selection.Copy method or the Session.InvokeDialog("EditCopySpecial") method instead.

Parameters

None

Applies to

[Session](#)

DataWatch

Syntax

DataWatch *Text*, [*Type*]

Description

Creates a new DataWatch object which fires an event when the specified Text is received from the host.

Parameters

The following table describes the parameters of the DataWatch method:

Parameters	Type	Description
<i>Text</i>	String	The text to watch for
<i>Type</i>	Variant	Reserved. Set to 0. (Short)

Return Value

[DataWatch](#). The DataWatch object created.

Applies to

[Session](#)

Example

Set a data watch on the text "Login"

```
' Declare object to watch for the text "Login"
Dim WithEvents LoginWatch As DataWatch
.
.
.
' Set up object (probably in Form_Load subroutine)
Set LoginWatch = wApp.Session.DataWatch("Login:")
```

DisplayArea

Syntax

DisplayArea *Left, Top, Right, Bottom*

Description

Creates a new DisplayArea Object which represent a rectangular section of the Window object.

Parameters

The following table describes the parameters of the DisplayArea method:

Parameters	Type	Description
<i>Left</i>	Long	The left column of the area
<i>Top</i>	Long	The top row of the area
<i>Right</i>	Long	The right column of the area
<i>Bottom</i>	Long	The bottom row of the area

Return Value

[DisplayArea](#). A DisplayArea object covering the specified rectangle

Applies to

[AutoWindow](#)

Example

Fill an area of the Window with asterisks

```
Session.Window.DisplayArea(10,10,70,12).CharacterFill " * "
```

DisplayAttribute

Syntax

DisplayAttribute *Type, Attribute*

Description

Sets the terminal emulation text attribute at the current text cursor or caret position in a wIntegrate session window. The effect of the new attribute will not be visible until new text is displayed.

Parameters

The following table describes the parameters of the DisplayAttribute method:

Parameters	Type	Description
<i>Type</i>	wAttributeTypeEnum	Specifies the attribute type
<i>Attribute</i>	wTextEffectEnum or wTextColorEnum	Specifies the value of the attribute, which depends on the attribute type

Applies to

[AutoWindow](#)

DisplayCommand

Syntax

DisplayCommand *Name*, [*Param1*], [*Param2*]

Description

Runs the emulation command specified.

Parameters

The following table describes the parameters of the DisplayCommand method:

Parameters	Type	Description
<i>Name</i>	String	The name of the emulation command to run
<i>Param1</i>	String	The first parameter for command which require one or more parameters
<i>Param2</i>	String	The second parameter for commands which require two parameters

Applies to

[AutoWindow](#)

Example

Clear the screen

```
Window.DisplayCommand( "FormFeed" )
```

DisplayText

Syntax

DisplayText *Text*

Description

Displays text in a wIntegrate session window at the current text cursor or caret position.

Parameters

The following table describes the parameters of the DisplayText method:

Parameters	Type	Description
<i>Text</i>	String	Specifies the text to display

Applies to

[AutoWindow](#)

Exit (Session)

Syntax

Exit

Description

This method asks the session to exit. If an exit script has been defined the exit script is run and the method return FALSE. Otherwise the exit event is fired and the session return TRUE.

This method if for internal use only and subject to change please do not use.

Parameters

None

Return Value

Boolean. Returns TRUE if exit event was fired, otherwise FALSE

Applies to

[Session](#)

Extension (Session)

Syntax

Extension *Id*

Description

Internal method to return an extension

This method is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the Extension method:

Parameters	Type	Description
<i>Id</i>	Variant	Extension identifier

Return Value

Long. Extension object or NULL. For internal use only.

Applies to

[Session](#)

GetSessionScript (Session)

Syntax

GetSessionScript

Description

This method creates and returns a string that contains the session configuration script.

i.e. The text of a script that when run will set the session back to its current state.

Parameters

None

Return Value

String. The text of the configuration script

Applies to

[Session](#)

Inflate

Syntax

Inflate *X, Y*

Description

Increases the size of a DisplayArea object by adding an amount to its sides.

Parameters

The following table describes the parameters of the Inflate method:

Parameters	Type	Description
<i>X</i>	Long	The amount to remove from the left hand and add to the right hand sides. A negative value will reduce the DisplayArea
<i>Y</i>	Long	The amount to remove from the top and add to the bottom sides. A negative value will reduce the DisplayArea.

Applies to

[DisplayArea](#)

InvokeDialog

Syntax

InvokeDialog *Dialog*

Description

Invokes the functionality behind a Session dialog box.

Dialog names correspond closely to wIntegrate menu options - see the "Client Script Reference" manual, "Reference - Script Menu Options" chapter for possible values.

Parameters

The following table describes the parameters of the InvokeDialog method:

Parameters	Type	Description
<i>Dialog</i>	Variant	Specifies the name of the dialog to display (String)

Applies to

[Session](#)

MoveCaret

Syntax

MoveCaret *Column, Row*

Description

Move the text caret to the new column and row. This is a short cut to using the `CaretPosition` property twice.

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0.

Parameters

The following table describes the parameters of the `MoveCaret` method:

Parameters	Type	Description
<i>Column</i>	Long	The new column for the caret
<i>Row</i>	Long	The new row for the caret

Applies to

[AutoWindow](#)

Open (Session)

Syntax

Open *Filename*

Description

This method opens a script file. This causes the following things to happen:

The "PreSessionOpen" script (from SetupApplication) is run if it exists

The script with the filename passed is run

Any "LoadScripts" defined in the script are run (from Setup Preferences)

The "PostSessionOpen" script (from SetupApplication) is run if it exists.

Alternative approaches from the object method:

Use the "RunScriptFile" method as this just runs the specified script without any other side effects.

Use the "Setting" property (or StoreSetting/UpdateSettings) to set the session variables. The variable name are in the "Client Scripting Reference" manual chapter "Reference - Script Menu Options"

Parameters

The following table describes the parameters of the Open method:

Parameters	Type	Description
<i>Filename</i>	String	The name of the script file to open

Applies to

[Session](#)

OpenScript (Session)

Syntax

OpenScript *Script*

Description

This method is given the text of a configuration script and opens it in the same way as a script file is opened.

i.e. Pre and Post session scripts and LoadScripts are also run.

Parameters

The following table describes the parameters of the OpenScript method:

Parameters	Type	Description
<i>Script</i>	String	The text of the script

Return Value

Long. The error code. Returns 0 if open process executed OK. Note: The script error dialog box will appear if there is an error in the script itself.

Applies to

[Session](#)

Paste (Session)

Syntax

Paste

Description

Paste data from clipboard to the session window

Parameters

None

Applies to

[Session](#)

Pause

Syntax

Pause

Description

Pauses a capture. While the capture is paused no data will be recorded until the Continue method is called.

Parameters

None

Applies to

[Capture](#)

Quit

Syntax

Quit

Description

Quits wIntegrate.

Parameters

None

Applies to

[Application](#)

Remove

Syntax

Remove *Index*

Description

Removes a specific object from a collection of wIntegrate objects.

Parameters

The following table describes the parameters of the Remove method:

Parameters	Type	Description
<i>Index</i>	Short	The index of the item to be removed

Applies to

[HotSpots](#)

RemoveAll

Syntax

RemoveAll

Description

Removes all objects from a collection of wIntegrate objects.

Parameters

None

Applies to

[HotSpots](#)

Resize

Syntax

Resize *Width, Height*

Description

Resizes a wIntegrate application window.

Parameters

The following table describes the parameters of the Resize method:

Parameters	Type	Description
<i>Width</i>	Long	The required window width in pixels
<i>Height</i>	Long	The required window height in pixels

Applies to

[Application](#)

RunScriptCommand

Syntax

RunScriptCommand *Command*

Description

Runs a wIntegrate script command.

Parameters

The following table describes the parameters of the RunScriptCommand method:

Parameters	Type	Description
<i>Command</i>	String	Specifies the name of the script command to run

Applies to

[Session](#)

RunScriptFile

Syntax

RunScriptFile *Command*, [*Argument*]

Description

Runs a wIntegrate script file.

Parameters

The following table describes the parameters of the RunScriptFile method:

Parameters	Type	Description
<i>Command</i>	String	Specifies the name of the script file to run
<i>Argument</i>	String	Specifies optional arguments

Applies to

[Session](#)

Save (Session)

Syntax

Save *FileName*

Description

Save session parameters as a script.

Parameters

The following table describes the parameters of the Save method:

Parameters	Type	Description
<i>FileName</i>	String	The filename for the save. If this is "" the save as dialog box is shown.

Return Value

Long. Returns 0 if the save was successful, otherwise 1.

Applies to

[Session](#)

ScrollAll (Session)

Syntax

ScrollAll *Direction*

Description

Scroll the window display top or bottom of the window.

Alternatively use the InvokeDialog method with the "ScrollTopPage" or "ScrollEndPage" parameter.

Parameters

The following table describes the parameters of the ScrollAll method:

Parameters	Type	Description
<i>Direction</i>	Short	Direction to scroll. 0 down, 1 up.

Applies to

[Session](#)

ScrollLine (Session)

Syntax

ScrollLine *Direction*

Description

Scroll the display back or forward 1 line.

Alternatively use the InvokeDialog method with the "ScrollUpLine" or "ScrollDownLine" parameter.

Parameters

The following table describes the parameters of the ScrollLine method:

Parameters	Type	Description
<i>Direction</i>	Long	Direction to scroll: 0 down, 1 up

Applies to

[Session](#)

ScrollPage (Session)

Syntax

ScrollPage *Direction*

Description

Scroll the display up or down one page.

Alternatively use the InvokeDialog method with the "ScrollUpPage" or "ScrollDownPage" parameter.

Parameters

The following table describes the parameters of the ScrollPage method:

Parameters	Type	Description
<i>Direction</i>	Short	Direction to scroll: 0 - down, 1 - up.

Applies to

[Session](#)

Select

Syntax

Select *Left, Top, Right, Bottom, Type, StartPage, EndPage*

Description

Selects an area of a wIntegrate session window.

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the Select method:

Parameters	Type	Description
<i>Left</i>	Long	Specifies the column number of the top left of the selection
<i>Top</i>	Long	Specifies the row number of the top left of the selection
<i>Right</i>	Long	Specifies the column number of the bottom right of the selection
<i>Bottom</i>	Long	Specifies the row number of the bottom right of the selection
<i>Type</i>	wSelectionTypeEnum	Specifies the type of selection
<i>StartPage</i>	Long	Specifies the page number of the top left of the selection

Parameters	Type	Description
<i>EndPage</i>	Long	Specifies the page number of the bottom right of the selection

Applies to

[Selection](#)

SelectAll

Syntax

SelectAll

Description

Selects all the data in a wIntegrate session.

Parameters

None

Applies to

[Selection](#)

SelectWindow

Syntax

SelectWindow

Description

Selects all the visible data in a wIntegrate session window.

Parameters

None

Applies to

[Selection](#)

Send

Syntax

Send *Data*, [*Options*]

Description

Sends data to a remote host as if the user had typed the data in a wIntegrate session.

If the Options argument is omitted, the default wSendOptionsEnum value is used.

Parameters

The following table describes the parameters of the Send method:

Parameters	Type	Description
<i>Data</i>	String	The data to be sent
<i>Options</i>	Variant or wSendOptionsEnum	Specifies any send options

Applies to

[Session](#)

SetEnd

Syntax

SetEnd *Right, Bottom, Page*

Description

Sets the end position, ie. the bottom right corner, of a highlighted selection. The method does nothing if the selection is not highlighted.

Note that rows and columns start at 0 - the top left of a wIntegrate Session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the SetEnd method:

Parameters	Type	Description
<i>Right</i>	Long	Specifies the column number of the bottom right of the selection
<i>Bottom</i>	Long	Specifies the row number of the bottom right of the selection
<i>Page</i>	Long	Specifies the page number of the bottom right of the selection

Applies to

[Selection](#)

SetStart

Syntax

SetStart *Left, Top, Page*

Description

Sets the start position, ie. the top left corner, of a highlighted selection. The method does nothing if the selection is not highlighted.

Note that rows and columns start at 0 - the top left of a wIntegrate Session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Parameters

The following table describes the parameters of the SetStart method:

Parameters	Type	Description
<i>Left</i>	Long	Specifies the column number of the top left of the selection
<i>Top</i>	Long	Specifies the row number of the top left of the selection
<i>Page</i>	Long	Specifies the page number of the top left of the selection

Applies to

[Selection](#)

SetStatusBar (Session)

Syntax

SetStatusBar *Object*

Description

Internal method to setup the status bar.

This method is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the SetStatusBar method:

Parameters	Type	Description
<i>Object</i>	Long or 2 or	Status bar pointer

Return Value

Long. Returns 0 if successful

Applies to

[Session](#)

SetupOpen (Session)

Syntax

SetupOpen

Description

This method shows the open dialog box (for scripts only).

It is obsolete and no longer supported. Use ShowDialog("FileOpen") instead.

Parameters

None

Applies to

[Session](#)

SetupSave (Session)

Syntax

SetupSave

Description

This method shows the save as dialog box (to create a script of the session configuration only).

It is obsolete and no longer supported. Use ShowDialog("FileSaveAs") instead.

Parameters

None

Applies to

[Session](#)

ShowDialog

Syntax

ShowDialog *Dialog*

Description

Displays a Session dialog box.

Dialog names correspond closely to wIntegrate menu options - see "Client Script Reference" manual - "Reference - Script Menu Options" for possible values.

Parameters

The following table describes the parameters of the ShowDialog method:

Parameters	Type	Description
<i>Dialog</i>	Variant	Specifies the name of the dialog to display (String)

Applies to

[Session](#)

SnapToSize (Session)

Syntax

SnapToSize *Snap*

Description

Internal method to implement snap to size.

This method is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the SnapToSize method:

Parameters	Type	Description
<i>Snap</i>	Boolean or 1 or	True to do the snap to size, False to return the Snap to size flag only

Return Value

Boolean. Returns current setting of the snap to size flag

Applies to

[Session](#)

Start

Syntax

Start

Description

Starts a capture object recording. A capture object is automatically started when it is created, so this method only needs to be called after a recording has been stopped using the Stop method.

Parameters

None

Applies to

[Capture](#)

Stop

Syntax

Stop

Description

Stops the recording for a capture object. If the capture is to a printer the captured data will now be printed.

Parameters

None

Applies to

[Capture](#)

StoreSetting

Syntax

StoreSetting *Name, Value*

Description

Allows the value session setting variable to be stored for update by a following UpdateSettings command. Use it to reduce screen refresh/window resizing when updating a group of settings which do these updates once.

See the "Client Script Reference" manual, "Reference - Script Menu Options" section for a list of the settings available.

Parameters

The following table describes the parameters of the StoreSetting method:

Parameters	Type	Description
<i>Name</i>	String	The name of the setting to use.
<i>Value</i>	String	The value of the setting

Applies to

[Session](#)

Example

Set lines/columns and backpages for the screen

```
With wApp.Session
  .StoreSetting "Lines", 24
  .StoreSetting "Columns", 132
  .StoreSetting "BackPages", 5
  .UpdateSettings
End With
```

UpdateSettings

Syntax

UpdateSettings

Description

Updates the all the session settings that have been stored by the StoreSetting method at once.

Use this method with the StoreSetting method to reduce screen refresh/window resizing when updating a group of settings.

Parameters

None

Applies to

[Session](#)

Example

Set lines/columns and backpages for the screen

```
With wApp.Session
    .StoreSetting "Lines", 24
    .StoreSetting "Columns", 132
    .StoreSetting "BackPages", 5
    .UpdateSettings
End With
```

Reference - Events

This chapter contains an alphabetical list of the event in the object model.

About (Session)

Syntax

About

Description

This event is a request that the container of a session control shows its about dialog box. It occurs when the Help About menu option is selected or run from the script language.

Parameters

None

Applies to

[Session](#)

Another (Session)

Syntax

Another *FileName*, *ShowPrompt*

Description

This event is fired when the user selects the File Another menu option or runs it through a script.

Parameters

The following table describes the parameters of the Another method:

Parameters	Type	Description
<i>FileName</i>	String	The file name for the new session
<i>ShowPrompt</i>	Boolean	True to prompt for the new session

Applies to

[Session](#)

CaptionChanged (Session)

Syntax

CaptionChanged

Description

This event is fired whenever the session title changes. To get the new version of the Caption use the Window.Caption property.

Parameters

None

Applies to

[Session](#)

Change

Syntax

Change

Description

Called when the data associated with a wIntegrate trigger has changed.

Parameters

None

Applies to

[Trigger](#)

Click

Syntax

Click

Description

Called when a wIntegrate object is clicked on with the mouse.

Parameters

None

Applies to

[HotSpot](#)

DbClick

Syntax

DbClick

Description

Called when a wIntegrate object is double-clicked on with the mouse.

Parameters

None

Applies to

[HotSpot](#)

Exit (Session)

Syntax

Exit

Description

This event is sent when the Exit or Exit all menu options have been selected or run through the script language.

Parameters

None

Applies to

[Session](#)

Found

Syntax

Found

Description

Called when a wIntegrate DataWatch object has match its text against the incoming data from the host. The FoundCount property will have been updated before this event is generated.

Parameters

None

Applies to

[DataWatch](#)

Full

Syntax

Full

Description

Called when a wIntegrate capture object is full of data. Use the Text property to retrieve the captured data.

Parameters

None

Applies to

[Capture](#)

HeightChanged (DockBar)

Syntax

HeightChanged

Description

This event is fired when a toolbar is added or removed from a docking bar. It is used to inform the containing application (or web page) that the size of docking bar may need to be increased to show all the toolbars current on it or decreased to prevent a blank space at the bottom of the toolbar.

On receipt of this event the RequiredHeight property can be interrogated to get the appropriate height for the toolbar.

Parameters

None

Applies to

[DockBar](#)

Version

5.2.0 First supported

Open (Session)

Syntax

Open *FileName*, *ShowPrompt*

Description

This event is sent by the session control when the File Open menu option is selected or run through a script.

Parameters

The following table describes the parameters of the Open method:

Parameters	Type	Description
<i>FileName</i>	String	The file name to open.
<i>ShowPrompt</i>	Boolean	True if the session control would like the containing application to show a prompt for the file name.

Applies to

[Session](#)

Opened (Session)

Syntax

Opened

Description

This event is obsolete and no longer generated by the session control

Parameters

None

Applies to

[Session](#)

RequestAlwaysOnTop (Session)

Syntax

RequestAlwaysOnTop *OnTop, Changed*

Description

The event is fired when the session control is set to be always on top through the menus or script language.

This event is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the RequestAlwaysOnTop method:

Parameters	Type	Description
<i>OnTop</i>	Boolean	True to put container always on top. False to turn off the always on top status.
<i>Changed</i>	Boolean *	Set to true in the event if the always on top status was changed.

Applies to

[Session](#)

RequestEnable (Session)

Syntax

RequestEnable *Enable*

Description

This event is fired through the session control when a script command would like the entire container application to be enabled or disabled.

This event is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the RequestEnable method:

Parameters	Type	Description
<i>Enable</i>	Boolean	True if the container should be enabled, False if the container should be disabled.

Applies to

[Session](#)

RequestLockSize (Session)

Syntax

RequestLockSize *LockSize*

Description

This event is sent when the "Session LockSize" script command is used and requests that the container does not allow the user to change its size.

This event is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the RequestLockSize method:

Parameters	Type	Description
<i>LockSize</i>	Boolean	True to lock the size, False to release the lock on the size

Applies to

[Session](#)

RequestResize (Session)

Syntax

RequestResize *x, y, width, depth, state*

Description

This event is sent from the session control to request that it is moved, resized or has its state changed.

This event is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the RequestResize method:

Parameters	Type	Description
<i>x</i>	Long	New x co-ordinate
<i>y</i>	Long	New y co-ordinate
<i>width</i>	Long	New width
<i>depth</i>	Long	New depth
<i>state</i>	Long	New show state

Applies to

[Session](#)

RequireDockBar (Session)

Syntax

RequireDockBar *Where*

Description

Internal event requesting a docking bar.

This event is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the RequireDockBar method:

Parameters	Type	Description
<i>Where</i>	Long	Location for dock bar

Applies to

[Session](#)

Save (Session)

Syntax

Save

Description

This event is sent when the File Save menu option or script command is run to request that the container of the session control saves its configuration.

Parameters

None

Applies to

[Session](#)

SaveAs (Session)

Syntax

SaveAs *FileName*, *ShowPrompt*

Description

This event is run when the File Save As menu option is selected or run from a script to request the container of the session control to save the configuration with the specified file name.

Parameters

The following table describes the parameters of the SaveAs method:

Parameters	Type	Description
<i>FileName</i>	String	The file name to save the session with
<i>ShowPrompt</i>	Boolean	TRUE if the container should show a prompt to let the user select the file name.

Applies to

[Session](#)

ShowStatusBar (Session)

Syntax

ShowStatusBar *Show*

Description

Request status bar is shown.

This event is for internal use and subject to change. Please do not use.

Parameters

The following table describes the parameters of the ShowStatusBar method:

Parameters	Type	Description
<i>Show</i>	Boolean *	True to show the status bar, False to hide the status bar.

Applies to

[Session](#)

Reference - Enumerations

This chapter contains an alphabetical list of the Enumerations used in the object model.

wAttributeTypeEnum

Description

These are constants that refer to different displayed text attributes

The values should not be added together for combined attributes.

Values

The following table values of the wAttributeTypeEnum enumerator:

Value	Description
<i>wAttributeTypeFont</i>	Font
<i>wAttributeTypeBackColor</i>	Background color
<i>wAttributeTypeForeColor</i>	Foreground color
<i>wAttributeTypeColor</i>	Foreground and background colors
<i>wAttributeTypeUser</i>	User attributes
<i>wAttributeTypeAll</i>	All the above attributes
<i>wAttributeTypeEffect</i>	Effect

Applies to

[Attribute](#) property, [Attributes](#) property, [DisplayAttribute](#) method, [AttributeFill](#) method

wBoxStyleEnum

Description

These are constants that refer to the styles to use when drawing a box on a DisplayArea object

Values

The following table values of the wBoxStyleEnum enumerator:

Value	Description
<i>wBoxStyleSingleBorder</i>	A single line border
<i>wBoxStyleDoubleBorder</i>	A double line border
<i>wBoxStyleBlockBorder</i>	A shaded block border
<i>wBoxStyleSolidBorder</i>	A solid block border
<i>wBorderStyleSingleSideBorder</i>	Add to the above to give a single line border to the left and right and the border specified above for the top and bottom.
<i>wBorderStyleDoubleSideBorder</i>	Add to the above to give a double line border to the left and right and the border specified above for the top and bottom.
<i>wBorderStyleBlockSideBorder</i>	Add to the above to give a shaded block border to the left and right and the border specified above for the top and bottom.
<i>wBorderStyleSolidSideBorder</i>	Add to the above to give a solid block border to the left and right and the border specified above for the top and bottom.
<i>wBorderStyleNoFill</i>	Add to the above if the inside of the border shouldn't be clear to spaces

Value	Description
<i>wBorderStyleMerge</i>	Add to the above if the lines for the border should be merged with any lines already on the screen
<i>wBoxStyleNoBorder</i>	Spaces are used for the border

Applies to

[Box method](#)

wCaptureFormatEnum

Description

These are constants that describe the data format of a wIntegrate capture object

Values

The following table values of the wCaptureFormatEnum enumerator:

Value	Description
<i>wCaptureFormatKeys</i>	Outgoing keyboard keys
<i>wCaptureFormatScreens</i>	Screen by screen
<i>wCaptureFormatCodes</i>	Incoming control codes
<i>wCaptureFormatAll</i>	All incoming and outgoing data, formatted for display
<i>wCaptureFormatTable</i>	Incoming data parsed into a tab separated table.
<i>wCaptureFormatRawIn</i>	All incoming data, unformatted (default)

Applies to

[Capture method](#)

wCaptureOptionsEnum

Description

These are the constants that allow various options to be applied to modify how the Capture will run

Note that the values maybe added together.

Values

The following table values of the wCaptureOptionsEnum enumerator:

Value	Description
<i>wCaptureOptionsShift</i>	When the capture buffer the data is shifted in the buffer so that the last half buffer characters are retained.
<i>wCaptureOptionsAutoOff</i>	The capture will automatically stop when the buffer is full
<i>wCaptureOptionsAppend</i>	The data for this capture is appended to the file (wCaptureTargetFile only).
<i>wCaptureOptionsDirectPrint</i>	The data is send directly to the printer with no formatting (wCaptureTargetPrinter only)
<i>wCaptureOptionsReadPrint</i>	The settings for the printer are read from the File->PrinterSetup dialog (wCaptureTargetPrinter only)
<i>wCaptureOptionsCharMapOff</i>	Switch off the emulations character mapping during the capture.
<i>wCaptureOptionsNone</i>	No special options apply (default)

Applies to

Capture method

wCaptureTargetEnum

Description

These are constants that describe the data target for a wIntegrate capture object

Use the Text property to retrieve data captured to memory. If capturing to a file, the FileName property must be set to a valid file name.

Values

The following table values of the wCaptureTargetEnum enumerator:

Value	Description
<i>wCaptureTargetFile</i>	Data is captured to file
<i>wCaptureTargetPrinter</i>	Data is captured to printer
<i>wCaptureTargetMemory</i>	Data is captured to memory (default)

Applies to

[Capture method](#)

wCaptureTextFormatEnum

Description

These are the constants that allow the format of the text returned from a capture object to be varied

Note that the wCaptureFormatStripZeroes maybe added to the other formats.

Values

The following table values of the wCaptureTextFormatEnum enumerator:

Value	Description
<i>wCaptureFormatBS</i>	Text returned in the wIntegrate Backslash format where certain characters are "escaped" out with a backslash character
<i>wCaptureFormatHex</i>	Each character is represented as tow hexadecimal digits
<i>wCaptureFormatStripZeroes</i>	Character zero is stripped from the returned text
<i>wCaptureFormatRaw</i>	Text returned as it was recorded

Applies to

[TextFormat](#) property

wCaretShapeEnum

Description

These are constants that describe a wIntegrate session text cursor or caret

Values

The following table values of the wCaretShapeEnum enumerator:

Value	Description
<i>wCaretShapeLine</i>	A line shape at the bottom of a text character cell
<i>wCaretShapeBlock</i>	A block shape occupying a whole text character cell

Applies to

[CaretShape](#) property

wCursorShapeEnum

Description

These are constants that refer to the different shapes of a mouse cursor

Values

The following table values of the wCursorShapeEnum enumerator:

Value	Description
<i>wCursorShapeArrow</i>	Normal arrow cursor
<i>wCursorShapeUpArrow</i>	Upward-pointing arrow cursor
<i>wCursorShapeCross</i>	Cross-hair cursor
<i>wCursorShapeIBeam</i>	Windows I-beam cursor
<i>wCursorShapeHourglass</i>	Hourglass cursor
<i>wCursorShapeCustom</i>	Custom cursor (seeMouseIcon)
<i>wCursorShapeDefault</i>	Standard wIntegrate scaling I-beam cursor

Applies to

[MousePointer property](#)

wDisplayUpdateEnum

Description

These are constants that refer to the display update mode for the Window object

Values

The following table values of the wDisplayUpdateEnum enumerator:

Value	Description
<i>wDisplayUpdateOff</i>	Do not update the display when the Window data changes
<i>wDisplayUpdateFormFeed</i>	Turn the display updating off until the next form feed
<i>wDisplayUpdateOn</i>	Update the display immediately the Window data changes

Applies to

[DisplayUpdate property](#)

wHotSpotStyleEnum

Description

These are constants that refer to different styles of a hotspot object

The wHotSpotStyleColored style must be applied to a hotspot before its color can be set (see HotSpot. BackColor , HotSpot. ForeColor , HotSpots. BackColor and HotSpots. ForeColor). Note that wHotSpotStyleColored maybe combined with any one of the other styles.

Values

The following table values of the wHotSpotStyleEnum enumerator:

Value	Description
<i>wHotSpotStyleRaised</i>	The hotspot appears raised
<i>wHotSpotStyleInset</i>	The hotspot appears inset or sunken
<i>wHotSpotStyleTextBox</i>	The hotspot appears with a text box style border
<i>wHotSpotStyleColored</i>	The hotspot's color can be specified
<i>wHotSpotStyleDefault</i>	Use with an individual hot spot to use the default style and colors from the hotspots collection.
<i>wHotSpotStyleTransparent</i>	The hotspot appears invisible against the wIntegrate session window (default)

Applies to

[Style property \(HotSpot\)](#), [Style property \(HotSpots\)](#)

wMousePositionEnum

Description

These are constants that refer to different characteristics of the mouse cursor position

Note that rows and columns start at 0 - the top left of a wIntegrate Session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Values

The following table values of the wMousePositionEnum enumerator:

Value	Description
<i>wMousePositionRow</i>	The character row number of the mouse cursor
<i>wMousePositionPage</i>	The page number of the mouse cursor
<i>wMousePositionColumn</i>	The character column number of the mouse cursor

Applies to

[MousePosition](#) property

wPositionEnum

Description

These are constants that refer to a location in a wIntegrate session window

Note that rows and columns start at 0 - the top left of a wIntegrate session window being at row 0, column 0.

Values

The following table values of the wPositionEnum enumerator:

Value	Description
<i>wPositionRow</i>	Row number
<i>wPositionColumn</i>	Column number

Applies to

[CaretPosition](#) property

wScrollAreaEnum

Description

These are constants that refer to window ScrollArea property of a DisplayArea object

Values

The following table values of the wScrollAreaEnum enumerator:

Value	Description
<i>wScrollAreaUnconfinedCaret</i>	The text cursor or caret maybe moved outside the scroll area
<i>wScrollAreaYes</i>	The area is a scroll area
<i>wScrollAreaNo</i>	Not a scroll area

Applies to

[ScrollArea property \(DisplayArea\)](#)

wSelectionFormatEnum

Description

These are constants that describe the required clipboard format of the contents of a Selection object

Values

The following table values of the wSelectionFormatEnum enumerator:

Value	Description
<i>wSelectionFormatTable</i>	The selection text formatted into tab-separated columns
<i>wSelectionFormatBitmap</i>	The selected area as a bitmap
<i>wSelectionFormatText</i>	The unformatted text described by the selection

Applies to

[Copy method](#)

wSelectionInfoEnum

Description

These are constants that refer to different characteristics of a selection

Note that rows and columns start at 0 - the top left of a wIntegrate Session window being at row 0, column 0. The current page is 0, the previous is 1, the one before that is 2, etc.

Values

The following table values of the wSelectionInfoEnum enumerator:

Value	Description
<i>wSelectionInfoTop</i>	The row number of the top left of the selection
<i>wSelectionInfoRight</i>	The column number of the bottom right of the selection
<i>wSelectionInfoBottom</i>	The row number of the bottom right of the selection
<i>wSelectionInfoType</i>	The type of selection
<i>wSelectionInfoStartPage</i>	The page number of the top left of the selection
<i>wSelectionInfoEndPage</i>	The page number of the bottom right of the selection
<i>wSelectionInfoLeft</i>	The column number of the top left of the selection

Applies to

[Information property](#)

wSelectionTextEnum

Description

These are constants that describe the required format of the text associated with a selection

Values

The following table values of the wSelectionTextEnum enumerator:

Value	Description
<i>wSelectionTextTable</i>	Text formatted into tab-separated columns
<i>wSelectionTextNormal</i>	Unformatted text

Applies to

[Text property \(Selection\)](#)

wSelectionTypeEnum

Description

These are constants that describe the selection type of a Selection object

Values

The following table values of the wSelectionTypeEnum enumerator:

Value	Description
<i>wSelectionTypeBlock</i>	The selection encloses a rectangular area of the session window contents
<i>wSelectionTypeLines</i>	The selection includes whole lines of the session window contents

Applies to

Type property ([Selection](#))

wSendOptionsEnum

Description

These are constants that describe the options for sending data to a remote host

Values

The following table values of the wSendOptionsEnum enumerator:

Value	Description
<i>wSendOptionsAppendCr</i>	Append Carriage Return to sent data (default)
<i>wSendOptionsNone</i>	No options

Applies to

[Send method](#)

wTextColorEnum

Description

These are constants that refer to different text colors

Values

The following table values of the wTextColorEnum enumerator:

Value	Description
<i>wTextColorBlue</i>	Blue
<i>wTextColorGreen</i>	Green
<i>wTextColorCyan</i>	Cyan
<i>wTextColorRed</i>	Red
<i>wTextColorMagenta</i>	Magenta
<i>wTextColorYellow</i>	Yellow
<i>wTextColorWhite</i>	White
<i>wTextColorBright</i>	Add to the above colors to give a brighter version of the color
<i>wTextColorBlack</i>	Black

Applies to

[BackColor](#) property (HotSpots), [ForeColor](#) property (HotSpots), [BackColor](#) property (HotSpot), [ForeColor](#) property (HotSpot)

wTextEffectEnum

Description

These are constants that refer to different text emulation effects

Note that the values may be added together for combined effects.

Values

The following table values of the wTextEffectEnum enumerator:

Value	Description
<i>wTextEffectDim</i>	Dim
<i>wTextEffectReverse</i>	Reverse
<i>wTextEffectUnderline</i>	Underline
<i>wTextEffectFlash</i>	Flash
<i>wTextEffectBold</i>	Bold
<i>wTextEffectSecret</i>	Hidden
<i>wTextEffectNormal</i>	Normal

Applies to

[DisplayAttribute](#) method, [AttributeFill](#) method

wTextFormatEnum

Description

These are constants that describe the required format of text retrieved from a wIntegrate session window

Values

The following table values of the wTextFormatEnum enumerator:

Value	Description
<i>wTextFormatTable</i>	Text formatted into tab-separated columns
<i>wTextFormatNone</i>	Unformatted text

Applies to

[Text property \(Window\)](#)

wTextLayoutEnum

Description

These are constants that describe the required layout of text retrieved from a wIntegrate session window

Values

The following table values of the wTextLayoutEnum enumerator:

Value	Description
<i>wTextLayoutBlock</i>	The text encloses a rectangular area of the session window
<i>wTextLayoutLines</i>	The text includes whole lines of the session window

Applies to

[Attributes property](#), [Text property \(Window\)](#)

wTriggerTypeEnum

Description

These are constants that refer to the different types of a wIntegrate trigger object

Values

The following table values of the wTriggerTypeEnum enumerator:

Value	Description
<i>wTriggerTypeText</i>	Notify when the display text equals the triggerPattern(default)
<i>wTriggerTypeMatch</i>	Notify when the display text matches the triggerPattern
<i>wTriggerTypeLineText</i>	Notify when a line of the display text equals the triggerPattern
<i>wTriggerTypeLineMatch</i>	Notify when a line of the display text matches the triggerPattern
<i>wTriggerTypeDataText</i>	Notify when the incoming data contains text equal to the triggerPattern
<i>wTriggerTypeAny</i>	Notify when the display text changes

Applies to

[Type property \(Trigger\)](#), [Add method \(Triggers\)](#)

wWindowStateEnum

Description

These are constants that refer to the different states of a window

Values

The following table values of the wWindowStateEnum enumerator:

Value	Description
<i>wWindowStateMaximized</i>	Maximized, ie. filling the whole screen
<i>wWindowStateMinimized</i>	Minimized, ie. iconized
<i>wWindowStateNormal</i>	Normal state, ie. neither maximized nor minimized

Applies to

[WindowState](#) property

wWordOptionsEnum

Description

These are constants that describe the options for retrieving a word from a wIntegrate session window

Values

The following table values of the wWordOptionsEnum enumerator:

Value	Description
<i>wWordOptionsLetter</i>	Get highlighted character in word
<i>wWordOptionsDelimiter</i>	Get delimited word
<i>wWordOptionsWord</i>	Get alphanumeric word

Applies to

[Word property](#)

Appendix A

Using the Object Model

Introduction

This chapter is aimed to give an overview of the facilities available using the wIntegrate Object model. It is not intended to be an exhaustive list of all the methods/properties available. Example code will usually be shown from Visual Basic, but applies equally to any other language that supports ActiveX controls or Automation.

Overview of the Object Model

The object model is used through (OLE) Automation to the wIntegrate Application or by the use of the wIntegrate Session Active X control from a third party application.

Accessing the object model through Automation

The Automation object for the wIntegrate application is "wIntegrate.document". This returns the application object. From the application object you can get the Session object.

Accessing the object model from an ActiveX control

Embedding the wIntegrate session control into an application gives direct access to the session object.

Overview of the main objects in the object model

The main objects used in the object model are:

Application object

This object allows you to control the way the wIntegrate application appears. It contains methods and properties to show/hide and reposition wIntegrate.

Session object

The session object is the root object for the emulation session. From here you can control all the functionality of the session itself (emulation, screen, scripting, sending/receiving data to/from the host, etc) through properties and methods or sub objects. When wIntegrate is embedded as an ActiveX control this is the dispatch interface returned.

Window object

The window object allows manipulation of the display of the session and contains functionality to display/read data directly to/from the screen. It also allows the shape to be set of the mouse and input cursors.

Selection object

The selection object allows the control of the selection of an area of the wIntegrate screen and this selection to be interrogated/manipulated.

Using the object model

This section provides a brief summary of some of the main features of the object model and how it can be used for certain tasks.

In the following examples wIntApp is an object variable holding the wIntegrate Application object. wIntSession is an object holding the wIntegrate session object, this could have been retrieved as the dispatch pointer from a session control or as wIntApp.Session.

Setting the show state of the application

The application object contains the WindowState property that allows you to if it appears minimized/maximized or restored. You can also hide or show the whole wIntegrate application with the Visible property.

Example

```
' Minimize wIntegrate
wIntApp.WindowState = wWindowStateMinimized
```

Setting up the Session

The parameters for the session are set up using the setting property. You can also setup a whole group of parameters by putting them all in a wIntegrate script and running it using the RunScriptFile method.

Example

```
' Set up the emulation and host
With wIntSession
.Setting("Emulation") = "vt100"
.Setting("Port") = "Windows Sockets"
.Setting("Host") = "myserver"
' open the port
.Setting("PortOpen") = "1"
EndWith
```

Retrieving text from the screen

Text can be retrieved from the screen using the Text property of the Window object or the text property of a Display Area object created from part of the Window. You can also use the Word property to return a word from the screen.

Example

```
' Get all the text including the back pages
text = wIntSession.Window.Text
' Get the first five characters of the current screen
text = wIntSession.Window.Text(0,0,4,0,0,0,0,0)
' Get a block of text formatted into a table from
' column 5 row 10 to column 20 row 12
text = wIntSession.Window.Text(5,10,20,12, wTextLayoutBlock,
wTextFormatTable, 0,0)
' Same text as above but using a display area
text = wIntSession.Window.DisplayArea(5, 10, 20,
12).Text(wTextFormatTable)
```

Displaying text on the screen

The Window object allows you to display text on the screen at any position and with any color or attribute. Use the MoveCaret method to position the text cursor. The DisplayAttribute method allows you to change the color or attribute and the DisplayText puts the text on the screen.

Example

```
' Put some red text on the top right of the screen
With wIntSession.Window
.MoveCaret 72,0
.DisplayAttribute wAttributeTypeColor, wTextColorRed
.DisplayText "ABCDEF"
EndWith
```

Finding out when part of the screen changes

Screen triggers can be set up using the triggers collection on the Session object to monitor an area of the screen until the text within it changes. These triggers will then send an event when the trigger condition is met.

Example

```
' Declare objects with events to hold the triggers
Dim WithEvents Trigger1 As Trigger
Dim WithEvents Trigger2 As Trigger
.
.
.
' Add the triggers, the first for any change in an area of the
screen
' the second to look for the word wIntegrate at a specific
location
Set Trigger1 = wIntSession.Triggers.Add("", wTriggerTypeAny, 5, 5,
20, 7)
Set Trigger2 = wIntSession.Triggers.Add("wIntegrate",
wTriggerTypeText, 5, 8, 14, 8)
.
```

```
.  
.br/>' Process the event from trigger1  
Private Sub Trigger1_Change()  
EventsReceived.AddItem ("Trigger 1 has been activated")  
EventsReceived.ListIndex = EventsReceived.ListCount - 1  
End Sub  
.br/>.br/.'  
' Finally release the triggers when no longer required  
Set Trigger1 = Nothing  
Set Trigger2 = Nothing
```

Sending data to the host

Data can be sent to the host using the Send method of the object model. This method has an optional parameter that specifies if a carriage return (ASCII 13) character is appended to the text.

Example

```
' Send a ECL command to list the ORDERS file to the session  
wIntSession.Send "LIST WIN.ORDER CUSTOMER", wSendOptionsAppendCr  
  
' Send my user name to the login prompt  
wIntSession.Send "Homer", wSendOptionsAppendCr
```

Retrieving/recording data sent from the host

Data can be recorded as it comes up from the host by using the Capture method of the session object. This returns a capture object, which can then be used to get the data recorded and to stop the recording. A capture object generates an event whenever its buffer becomes full. The captured data can be recorded to memory, a file or to the printer.

Example

```
' Create a capture of raw data from the host to memory  
' using a 1K buffer
```

```
Set Capture1 = wIntSession.Capture(wCaptureTargetMemory,  
wCaptureFormatRawIn, wCaptureOptionsNone, 1024)  
.   
.   
.   
' After a suitable interval or some specific event  
' turn off the capture, get the text and release it  
Capture1.Stop  
MsgBox Capture1.Text  
Set Capture1 = Nothing
```

Waiting for a specific piece of data to be received

The DataWatch method on the Session Object returns a DataWatch object, which will fire a Found event whenever the specified text is received from the host.

Example

```
' Set up an object to do the watching  
Dim WithEvents Watcher As DataWatch  
.   
.   
.   
' Set up a data watch for whenever the text "Buffy" is received  
' from the host  
Set Watcher = wIntSession.DataWatch("Buffy")  
.   
.   
.   
' Event triggered whenever the text is received  
Private Sub Watcher_Found()  
' Put the number of times the data has been received into a label  
lblOccurances.Caption = Str$(Watcher.FoundCount)  
End Sub  
.   
.   
.   
' Release the DataWatch when it is no longer required  
Set Watcher = Nothing
```

Index

A

About Session event, [192](#)
AboutBox Session method, [128](#)
Add HotSpots method, [129](#)
Add Triggers method, [131](#)
AlwaysOnTop property, [42](#)
Another Session event, [193](#)
AppendText method, [133](#)
Application object, [10](#)
AppName Session property, [43](#)
AppTitle Session property, [44](#)
AttachDockBar Session method, [134](#)
Attribute property, [45](#)
AttributeFill method, [136](#)
Attributes property, [47](#)
AutoBridgeRx object, [12](#)
AutoBridgeTx object, [13](#)
AutoData object, [14](#)
AutoFileTransfer object, [15](#)
AutoImportTarget object, [16](#)
AutoWindow object, [17](#)

B

BackColor HotSpot property, [49](#)
BackColor HotSpots property, [50](#)
Bottom property, [51](#)
Box method, [137](#)
Busy Session property, [52](#)

C

CanExit Session method, [138](#)
CanRestore Session method, [139](#)
Caption property, [53](#)
CaptionChanged Session event, [194](#)
Capture method, [140](#)
Capture object, [20](#)
CaretPosition property, [54](#)
CaretShape property, [55](#)
Change event, [195](#)
CharacterFill method, [142](#)
Clear Capture method, [143](#)
Clear Selection method, [144](#)
Click event, [196](#)
CommandBars property, [56](#)
ContainerFlags Session property, [57](#)
Continue method, [145](#)
Copy method, [146](#)
Copy Session method, [147](#)
CopySpecial Session method, [148](#)
Count Capture property, [58](#)
Count HotSpots property, [59](#)
Count Triggers property, [60](#)

D

DataWatch method, [149](#)
DataWatch object, [22](#)

DbClick event, [197](#)
DisplayArea method, [150](#)
DisplayArea object, [23](#)
DisplayAttribute method, [151](#)
DisplayCommand method, [152](#)
DisplayHorizontalScrollBar property, [61](#)
DisplayOn property, [62](#)
DisplayStatusBar property, [63](#)
DisplayText method, [153](#)
DisplayUpdate property, [64](#)
DisplayVerticalScrollBar property, [65](#)
DockBar object, [25](#)

E

Enabled property, [66](#)
Exit Session event, [198](#)
Exit Session method, [154](#)
Extension Session method, [155](#)

F

FileName property, [67](#)
ForeColor HotSpot property, [68](#)
ForeColor HotSpots property, [69](#)
Found event, [199](#)
FoundCount property, [70](#)
Full event, [200](#)
FullName Session property, [71](#)

G

GetSessionScript Session method, [156](#)

H

Height property, [72](#)
HeightChanged DockBar event, [201](#)
HotSpot object, [27](#)
HotSpots object, [29](#)
HotSpots property, [73](#)
hWnd Session property, [126](#)

I

Id DockBar property, [74](#)
Id Session property, [75](#)
Index property, [76](#)
Inflate method, [157](#)
Information property, [77](#)
InvokeDialog method, [158](#)
Item HotSpots property, [78](#)
Item Triggers property, [79](#)

L

Left property, [80](#)
Length property, [81](#)

M

MouseIcon property, [82](#)
MousePointer property, [83](#)
MousePosition property, [84](#)
MoveCaret method, [159](#)

N

Name Session property, [85](#)

O

Open Session event, [202](#)

Open Session method, [160](#)

Opened Session event, [203](#)

OpenScript Session method, [162](#)

P

Paste Session method, [163](#)

Pattern property, [86](#)

Pause method, [164](#)

Paused property, [87](#)

Q

Quit method, [165](#)

R

Remove method, [166](#)

RemoveAll method, [167](#)

RequestAlwaysOnTop Session event, [204](#)

RequestEnable Session event, [205](#)

RequestLockSize Session event, [206](#)

RequestResize Session event, [207](#)

RequiredHeight DockBar property, [88](#)

RequireDockBar Session event, [208](#)

Resize method, [168](#)

Right property, [89](#)

Running property, [90](#)

RunScriptCommand method, [169](#)

RunScriptFile method, [170](#)

S

Save Session event, [209](#)

Save Session method, [171](#)

SaveAs Session event, [210](#)

ScreenArea property, [91](#)

ScrollAll Session method, [172](#)

ScrollArea DisplayArea property, [92](#)

ScrollArea Window property, [93](#)

ScrollLine Session method, [173](#)

ScrollPage Session method, [174](#)

Select method, [175](#)

SelectAll method, [177](#)

Selected property, [94](#)

Selection object, [31](#)

Selection property, [95](#)

SelectWindow method, [178](#)

Send method, [179](#)

Session object, [33](#)

Session property, [96](#)

SetEnd method, [180](#)

SetStart method, [181](#)

SetStatusBar Session method, [182](#)

Setting property, [97](#)

SettingsDialog property, [98](#)

SetupOpen Session method, [183](#)

SetupSave Session method, [184](#)

ShowDialog method, [185](#)

ShowStatusBar Session event, [211](#)

SnapToSize Session method, [186](#)
Start method, [187](#)
StatusBarText property, [99](#)
Stop method, [188](#)
StoreSetting method, [189](#)
Style HotSpot property, [100](#)
Style HotSpots property, [101](#)

T

Text Capture property, [102](#)
Text DataWatch property, [103](#)
Text DisplayArea property, [104](#)
Text Selection property, [105](#)
Text Window property, [106](#)
TextFormat property, [108](#)
Top property, [109](#)
Trigger object, [38](#)
Triggers object, [40](#)
Triggers property, [110](#)
Type Selection property, [111](#)
Type Trigger property, [112](#)

U

UpdateSettings method, [190](#)

V

Variable property, [113](#)
Version property, [114](#)
Visible property, [115](#)

W

wAttributeTypeEnum enum, [213](#)
wBoxStyleEnum enum, [214](#)
wCaptureFormatEnum enum, [216](#)
wCaptureOptionsEnum enum, [217](#)
wCaptureTargetEnum enum, [219](#)
wCaptureTextFormatEnum enum, [220](#)
wCaretShapeEnum enum, [221](#)
wCursorShapeEnum enum, [222](#)
wDisplayUpdateEnum enum, [223](#)
WHeight Session property, [116](#)
wHotSpotStyleEnum enum, [224](#)
Width property, [121](#)
Window property, [122](#)
WindowState property, [123](#)
WLeft Session property, [117](#)
wMousePositionEnum enum, [225](#)
Word property, [124](#)
wPositionEnum enum, [226](#)
wScrollAreaEnum enum, [227](#)
wSelectionFormatEnum enum, [228](#)
wSelectionInfoEnum enum, [229](#)
wSelectionTextEnum enum, [230](#)
wSelectionTypeEnum enum, [231](#)
wSendOptionsEnum enum, [232](#)
WState Session property, [118](#)
wTextColorEnum enum, [233](#)
wTextEffectEnum enum, [234](#)
wTextFormatEnum enum, [235](#)
wTextLayoutEnum enum, [236](#)
WTop Session property, [119](#)
wTriggerTypeEnum enum, [237](#)
WWidth Session property, [120](#)
wWindowStateEnum enum, [238](#)
wWordOptionsEnum enum, [239](#)